
ROBOTS AND MANUFACTURING AUTOMATION

C. RAY ASFAHL

University of Arkansas, Fayetteville

JOHN WILEY & SONS

New York Chichester Brisbane Toronto Singapore

Р. Асфаль

РОБОТЫ и автоматизация производства

Перевод с английского
М.Ю. Евстегнеева, Б.И. Копылова,
канд. техн. наук А.С. Чубукова



Москва
•Машиностроение•
1989

ББК 32.816

A91

УДК [621.865.8.004.14:338.364]=03=20

Асфаль Р.

A91 Роботы и автоматизация производства/Пер. с англ.
М. Ю. Евстегнеева и др. — М.: Машиностроение, 1989. —
448 с.: ил.

ISBN 5-217-00620-X

В книге автора — сотрудника Центра робототехники и автоматизации при Арканзасском университете (США) анализируются основные направления развития робототехники, в том числе взаимосвязь промышленных роботов с другими элементами автоматизации. Особое внимание уделено не проблемам конструирования роботов, а вопросам их внедрения и эксплуатации на современных предприятиях и заводах будущего.

Для инженеров всех отраслей промышленности, занимающихся автоматизацией производственных процессов.

2402020000—028
A_{038 (01)}—89 28—89

ББК 32.816

ISBN 5-217-00620-X (СССР)
ISBN 0-471-80212-3 (США)

© 1985. John Wiley & Sons, Inc.
© Перевод на русский язык, из-
дательство «Машиностроение»,
1989

Промышленные роботы стали общепризнанным символом движения за автоматизацию производства, причем их место обсуждается уже десятилетиями. Однако, несмотря на всю свою сенсационность, сегодня не они являются самым значительным достижением в области автоматизации производства. За кулисами находятся программируемые контроллеры, микропроцессоры, ЭВМ, управляющие технологическими процессами, и логические системы управления, пользующиеся даже большим успехом и шире применяющиеся в этой области, чем роботы. В то же время все перечисленные устройства могут рассматриваться как члены одной семьи оборудования для гибкой автоматизации, меняющей сегодня систему промышленного производства. В книге описана взаимосвязь между всеми этими устройствами, входящими в автоматизированную систему производства.

О роботах написано много книг, однако в основном в них дано лишь схематичное представление о предмете. Назрела необходимость в учебнике для технических колледжей. Предлагаемая книга изобилует примерами, в конце каждой главы приведены вопросы и упражнения для домашних и классных занятий.

Основное внимание в книге уделено применению роботов и автоматизированных производственных систем, а не их проектированию. Автор попытался сделать книгу полезной для инженеров, технологов и управленческого персонала. Познания из областей высшей математики и программирования для ЭВМ необязательны для понимания материала, но некоторые упражнения составлены в расчете на студента, имеющего такую подготовку.

В прилагаемом к изданию руководстве даны объяснения к упражнениям и приведены рекомендации по отбору классных заданий в соответствии с предполагаемым составом обучающихся.

Этой книгой можно пользоваться для проведения курса обучения, включающего или не включающего лабораторные занятия. Даже при отсутствии какого-либо оборудования читатель в состоянии проработать все упражнения и понять все темы. Если лабораторные занятия предусмотрены в курсе обучения, руководство по эксплуатации имеющегося оборудования будет полезным дополнением к учебнику. В отличие от роботов промышленные программируемые контроллеры, микропроцессоры и

микроЭВМ для управления технологическими процессами не дороги, что делает доступным организацию интересных лабораторных работ. Имеются также дешевые настольные демонстрационные роботы. Не следует упускать из виду имеющиеся прекрасные аудиовизуальные пособия для сопровождения материала учебника.

Многие читатели, имеющие отношение к промышленности, хотели бы знать, как начать внедрение роботов и автоматизацию производства — в книге они найдут некоторые предпосылки. В гл. 1 описаны конструктивные особенности изделий и характеристики процессов управления, способствующие увеличению потенциала автоматизации. Иногда подготовка к автоматизации больше дает для повышения эффективности производства на фирме, чем сама автоматизация, о чем свидетельствуют приведенные в тексте примеры из практики. В гл. 2 доступным языком описаны составные части роботов и автоматизированных систем, о которых рассказано в последующих главах. Наличие механизированной подачи, перемещения и ориентации изделий (гл. 3) во многих случаях важно и при применении роботов, в то же время принципы, изучаемые в этой главе, могут быть применены и при работе вручную. Гл. 4 посвящена автоматическим линиям для обработки и сборки изделий, включая анализ надежности интегрированных автоматизированных систем. Разбираться в этом анализе необходимо для понимания последующих глав о робототехнике. В гл. 5 речь идет о станках с программным управлением, предшественниках промышленных роботов и прародителях гибкой автоматизации. Гл. 6—9 полностью посвящены промышленным роботам: их описанию, программированию, внедрению и использованию. В гл. 10 и 11 описаны и проанализированы промышленные логические системы управления — на сегодняшний день наиболее универсальный и широко используемый инструмент автоматизации производства. Эти системы используются и для управления самими роботами, хотя широкая публика видит только робот. В гл. 12 представлены ПК (программируемые контроллеры, а не персональные компьютеры). Использование ПК — наиболее популярный способ внедрения промышленных логических систем управления. Концепцию этих устройств легко понять, владея основами, изложенными в гл. 10 и 11. Проработав гл. 12, читатели сами будут в состоянии создать специализированный робот из стандартных компонентов в противовес предлагаемым готовым покупным промышленным роботам. ЭВМ, управляющие технологическими процессами (гл. 13), используются для управления группами роботизированных ячеек, производственными линиями и даже целыми заводами. В то же время существуют децентрализованные системы, управляемые малогабаритными недорогими микропроцессорами (гл. 14), что является технологическим открытием и движущей силой автоматизации и роботизации. В заключительной гл. 15 введено понятие объединения роботов и си-

стем автоматизации промышленности, рассказано о гибких производственных системах (ГПС) групповой технологии и производствах с комплексным управлением от ЭВМ. В гл. 15 исследованы и проанализированы вопросы этики робототехники и производственной автоматизации.

Книга является начальным курсом производственной автоматизации. Для инженерных специальностей лучше всего начинать изучение вопросов робототехники и автоматизации производства на ранней стадии обучения. Затем студенты, желающие пройти специализацию, могут прослушать следующие курсы более высокого уровня по автоматизированному проектированию и изготовлению с использованием ЭВМ, производством с комплексным управлением от ЭВМ, ГПС, вычислительной технике или специальные курсы по робототехнике в зависимости от оснащенности лабораторий. Книга может оказаться полезной для высшего управленческого аппарата как в производственной, так и в административной сфере. Предварительный черновой вариант курса хорошо зарекомендовал себя при преподавании на уровне выпускников при привлечении дополнительной литературы и результатов исследований.

Я признателен студентам, работникам и представителям промышленности, сотрудничающим в Центре робототехники и автоматизации Арканзасского университета за их идеи, проекты и опыт, использованные при создании книги. Особой благодарности заслуживает г-н Дэйв Бет, представитель фирмы «Зингер», являющийся членом университетского Центра по промышленности. Во многом помогли также директор Центра профессор В. Рейфер и один из его основателей профессор Р. Элирсон. Конструктивная критика и полезные предложения были получены от рецензентов Д. Бедворта, М. Кейса, Д. Кимблера и Х. Янга. Заслуживают особого упоминания Д. Батсон, Д. Ранни, В. Сариана, А. Янг, Д. Чемпион, Д. Дуглас и Р. Симс. Большую помощь оказал П. Росато из фирмы «Юнимейшен — Вестингауз», предоставив в наше распоряжение много иллюстраций и примеров. Неоценимы были многие таланты моей жены Вики, которая не только перепечатывала рукопись, но также была моим советником и техническим редактором с начала до конца работы.

Р. Асфаль

1. ПОДГОТОВКА К АВТОМАТИЗАЦИИ

1.1. ВЛИЯНИЕ КОНКУРЕНЦИИ

Дж. Бэйкер, исполнительный вице-президент и руководитель сектора технических систем фирмы «Дженерал электрик», сравнил жестокую конкуренцию в современных отраслях промышленности, выпускающих штучную продукцию, с удавом, сжимающим свою жертву. По словам Бэйкера, существует общее мнение, что эта огромная змея просто бросается с дерева на свою жертву и сжимает ее, пока та не умрет, но на самом деле удав действует более изощренно. Он сильно обхватывает грудную клетку жертвы своими кольцами и каждый раз, как жертва расслабляется и выдыхает воздух, удав выбирает образовавшуюся слабинку. После трех или четырех выдохов слабины уже не будет. Жертва быстро задыхается, и удав ее проглатывает [1].

Дж. Бэйкер рассматривает взлеты и спады экономических циклов как элементы циклов «дыхания» промышленности под воздействием конкуренции. Каждый раз, когда сбыт растет, товарно-материальные запасы сокращаются, а уровень прибыли восстанавливается, можно быть удовлетворенным эффективностью производства и с облегчением вздохнуть. Но именно в этот момент удав конкуренции еще плотнее сжимает свои кольца.

Было время, когда конкурентная борьба промышленников за выживание происходила в пределах одной страны. Воздвигнутые по всему миру транспортные, коммуникационные и торговые барьеры создавали условия для изоляции промышленности каждой из стран от зарубежных конкурентов. Еще более важным было то, что финансовые и технологические преимущества сосредоточились в руках немногих развитых стран, которые казались неуязвимыми для руководителей промышленности менее развитых государств. Однако роскошь, которую позволяли себе богатые страны за счет воздвигнутых на пути международной конкуренции барьеров, обернулась их слабостью, стала трещинами в броне, защищавшей их от конкурентов. Эта роскошь — высокий уровень заработной платы, неэффективное руководство промышленностью, устаревшее оборудование — позволила изголодавшимся конкурентам разрушить барьеры и захватить рынки за счет низкого уровня заработной платы, целеустремленного руководства и нового оборудования, вобравшего в себя самые последние достижения технологии, разработанной в тех самых странах, которые находились в осаде.

Любая ностальгическая мысль о былом могуществе промышленных гигантов лишь дает удаву повод для улыбки. Воспоминания о триумфальных днях не спасут промышленность, растерявшую преимущества перед конкурентами.

Для таких отраслей промышленности автоматизация создает возможности для требуемого подъема производительности труда, который позволил бы изменить тенденцию развития, снова завоевать рынки и вырваться из объятий удава. Автоматизация, несомненно, не является новостью, и в широком смысле этого слова, ее появление можно соотнести с временами промышленной революции, когда машины впервые значительно повысили производительность труда рабочих. Однако развитие автоматизации происходило скачкообразно, оно характеризуется рядом крупных достижений. Одним из них было внедрение взаимозаменяемости в производстве, другим — сборочные конвейеры Генри Форда. Эта книга еще об одном таком достижении, которое пока находится в стадии реализации и может со временем стать одним из величайших научно-технических достижений. Хотя слово «робот» вынесено в заглавие настоящей книги, промышленные роботы сами по себе не являются достижением, они — продукт или результат достижения. Роботы стали воплощением современного развития автоматизации производства и привлекают внимание любого инженера, связанного со штучным производством.

Конечно, автоматизация не является единственным способом освободиться из объятий удава. Когда покупательная способность денег падает, реакция трудящихся замечательна. Требование повышения заработной платы прежде следовало почти неизбежно. Лидеры американских профсоюзов 60-х годов не поверили бы своим глазам, если бы им в то время дали возможность ознакомиться с уровнем заработной платы и льготами, обеспечиваемыми профсоюзами в 80-х годах.

Другой концепцией, используемой как оружие в конкурентной борьбе, является участие рабочих в управлении производством. Первой областью, где это оружие было применено, стала борьба за качество, а наиболее популярным термином, характеризующим участие рабочих в управлении производством, стал термин «кружок качества». Японцы первыми применили метод организации кружков качества для преодоления дурной славы, закрепившейся за их некачественной продукцией, выпускавшейся в 50-х годах. Кружки качества призывают заводских рабочих любого уровня делиться идеями, направленными на улучшение качества выпускаемой ими продукции. Призыв нашел отклик, и сейчас Япония имеет репутацию страны, производящей высококачественную продукцию по конкурентоспособным ценам. Сейчас идей организации кружков качества пользуются и в других странах, особенно в США. Круг их задач расширен и включает, кроме обеспечения качества, вопросы снижения стоимости выпускаемой продукции и обеспечения безопасности работ.

В этой книге описаны виды оборудования и методы, с помощью которых можно обеспечить автоматизацию завода, что Дж. Бейкер назвал «спасательным кругом промышленности». Наряду с указанием путей автоматизации в книге рассмотрены неблагоприятные моменты и подводные камни, встречающиеся на этих путях. Одна из простейших ошибок, которую можно допустить, приступая к автоматизации, это начать ее, не подготовив предварительно продукцию и процесс изготовления. В этой главе указаны пути подготовки завода к автоматизации. Большая часть этой подготовки полезна для любого завода, вне зависимости от того, будет ли он в конце концов автоматизирован или нет.

1.2. КОНСТРУИРОВАНИЕ ДЛЯ АВТОМАТИЗАЦИИ

Конструкция изделия должна обеспечивать его технологичность в изготовлении и сборке. Появление промышленного робота повлекло за собой повышение внимания к конструкции изделий с точки зрения облегчения ориентации, позиционирования и сопрягаемости деталей при сборке. Приятной неожиданностью явилось то, что проработка, выполненная на этапе проектирования с целью обеспечения возможности роботизации транспортных и сборочных операций, приводит также к повышению производительности ручного производства. Существенная часть проектов роботизации выливается в разработку высокоэффективных полуавтоматических процессов производства, в которых роботы могут вообще выпасть из окончательного рассмотрения.

В гл. 3 показано, что большинство автоматов для перемещения и ориентации деталей действуют наощупь, т. е. они используют механические характеристики деталей для осуществления ориентации и позиционирования. Хотя некоторые роботы оснащаются

развитыми системами технического зрения для ориентации и перемещения деталей, в большинстве случаев их применение экономически нецелесообразно. Поэтому роботы или автоматизированные системы должны работать «вслепую», и инженеру по автоматизации также полезно закрыть глаза, когда

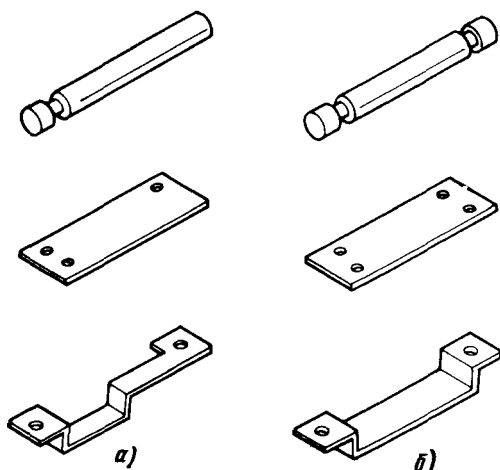


Рис. 1.1
Ориентация деталей за счет их симметричности:
а — затруднена; б — улучшена

Рис. 1.2

Ориентация деталей за счет их асимметрии:

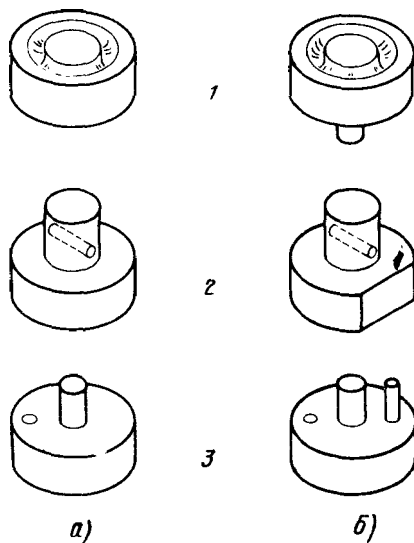
а — затруднена; *б* — улучшена; *1* — деталь с выступом; *2* — деталь с лыской; *3* — деталь со штырем

он пытается представить, как они обращаются с деталями.

Симметрия. В некоторых случаях симметрия деталей способствует автоматизации, а в других — делает ее невозможной. Рассмотрим примеры, приведенные на рис. 1.1. Все детали, расположенные справа, симметричны, что делает ориентирование ненужным. Для того чтобы добиться симметрии двух верхних деталей

левого столбца, необходима их дополнительная обработка, затраты на которую будут меньше по сравнению с затратами на ориентирование соответствующих асимметричных деталей. Рис. 1.2 иллюстрирует другую проблему. Если конструктивные особенности каждой детали трудно обнаружить механическим способом, то решение проблемы состоит в нарушении симметрии. Для этого на первой детали сделан дополнительный выступ в нижней части диска. На большем цилиндре второй детали выфрезерована лыска, которая служит для фиксирования угловой позиции отверстия, просверленного в меньшем цилиндре. Штифт на третьей детали также используется с целью фиксирования угловой позиции отверстия, ось которого параллельна оси детали. Эти примеры показывают, как в отдельных случаях нарушение симметрии деталей упрощает процесс автоматизации, причем детали типа цилиндров и дисков являются наиболее вероятными кандидатами на внесение черт асимметрии, потому что без ориентирующих признаков они могут принимать неопределенное число положений. Детали прямоугольной формы однако обычно выигрывают от симметрии, поскольку они могут иметь небольшое число положений и конструктор может сделать все эти положения работоспособными.

Взаимное сцепление деталей. Каждый, кто разбирал ящик с несортированными пружинами, мог прийти к выводу о необходимости специальных конструкций пружин, предотвращающих их сцепление или запутывание. Рис. 1.3 иллюстрирует решение проблемы сцепления пружин и другие примеры конструкций, уменьшающих возможность сцепления. Многие детали имеют отверстия и выступы, функционально не связанные друг с другом



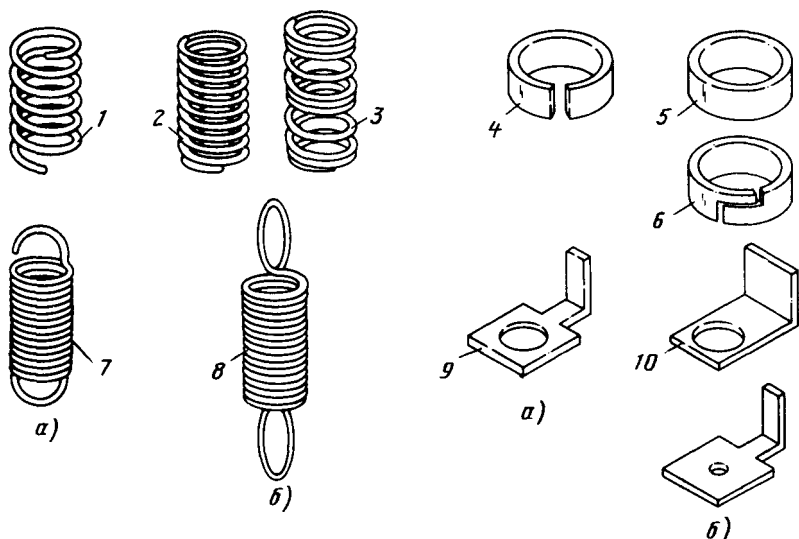


Рис. 1.3

Неудачные (а) и удачные (б) конструкции деталей, приводящие к взаимному сцеплению во время подачи:

1 — неподжатые крайние витки; 2 — крайние витки поджаты; 3 — промежутки между витками меньше, чем диаметр проволоки, что предотвращает запутывание; 4 — широкая прорезь; 5 — прорезь отсутствует; 6 — криволинейная узкая прорезь; 7 — незамкнутые зацепы; 8 — замкнутые зацепы; 9 — отверстие больше язычка; 10 — язычок больше отверстия

и не предназначенные для сопряжения. Соотношение размеров этих элементов деталей должно исключать возможность попадания выступа в отверстие и сцепления деталей.

Конструирование для обеспечения подачи деталей. В гл. 3 рассмотрены механизмы для последовательной подачи деталей друг за другом по транспортной дорожке. Для обеспечения продвижения деталей вперед в этих механизмах используется вибра-

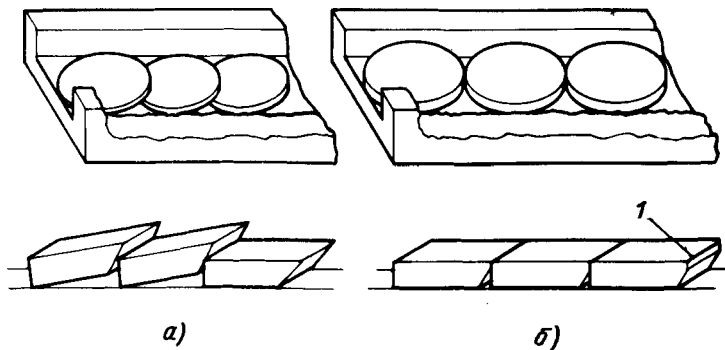


Рис. 1.4

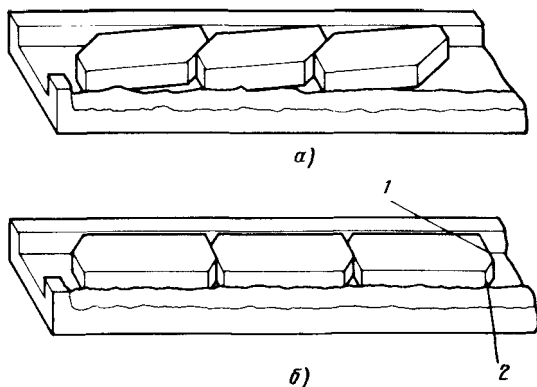
Конструкции, вызывающие наполнение деталей:

а — неудачные; б — удачные; 1 — плоский торец детали

Рис. 1.5

Конструкции, вызывающие заклинивание деталей:

a — неудачные; *б* — удачные; *1* — увеличенная плоскость торца, перпендикулярного направлению движения; *2* — уменьшенный угол



ция или сила тяжести, и усилие передается от детали к детали, когда они подталкиваются сзади. Этот метод особенно хорошо применим для плоских деталей со стабильно заданной ориентацией. Но если детали слишком тонкие или если их кромки имеют скосы, они будут наползать одна на другую (рис. 1.4). Аналогичной является проблема заклинивания деталей, когда соприкасающиеся кромки не перпендикулярны направлению перемещения (рис. 1.5).

Конструирование для сборки сопрягаемых деталей

Сборка сопрягаемых деталей с помощью робота или другого механизма является достаточно сложной технической задачей даже в тех случаях, когда детали правильно ориентированы и выдержан допуск на их изготовление. В гл. 6 раскрыты некоторые секреты, которые помогут преодолеть трудности, возникающие при сборке сопрягаемых деталей, и все же довольно часто можно обеспечить упрощение сборки еще на стадии конструирования. Рис. 1.6 показывает, как использование фасок на деталях облегчает сборку вала со втулкой. Из этого примера видно, что незначительное усовершенствование конструкции деталей является решающим для обеспечения успешного применения промышленных роботов.

Крепежные детали. В гл. 4 рассмотрена конструкция механизма для завинчивания винтов, но все же, надо стараться избегать применения винтов и другого крепежа, так как их использование усложняет процесс сборки. На рис. 1.7 представлена конструкция защелки, которая просто фиксирует деталь на месте. Рис. 1.8 является характерным примером того, как использование принципов конструирования узлов с учетом последующей автоматизации их сборки привело к созданию новой конструкции, в которой один винт заменяет сразу четыре.

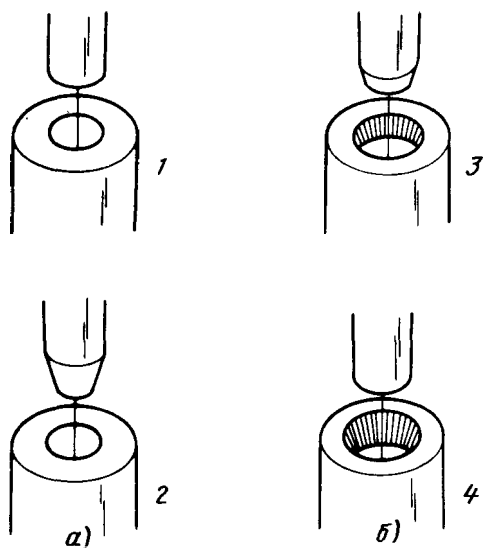


Рис. 1.6

Сборка с помощью фасок:

1 — фаски отсутствуют, сборка затруднена; 2 — фаска на верхней детали; 3 — фаска на обеих деталях; 4 — фаска на нижней детали

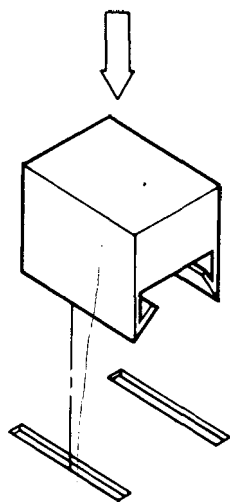


Рис. 1.7

Конструкция защелки, входящей в прорези

При применении крепежных деталей автоматизация сборки упростится, если провести их унификацию по типам и размерам. Использование более дорогих и мощных винтов большего, чем требуется, размера может показаться нарушением принципов конструирования и экономической целесообразности. Но если такие винты можно использовать во всей сборочной единице, имеет смысл это сделать в угоду автоматизации. Унификация может обеспечить крупную экономию за счет сокращения затрат

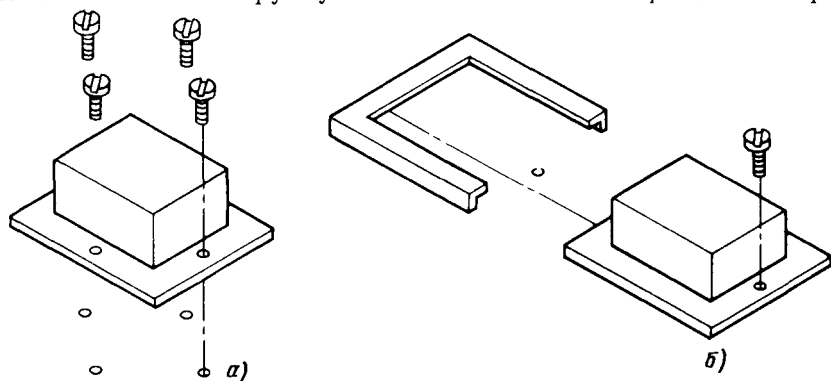


Рис. 1.8

Сокращение количества винтовых креплений:

а — конструкция, неудачная для автоматизации; б — предпочтительная конструкция

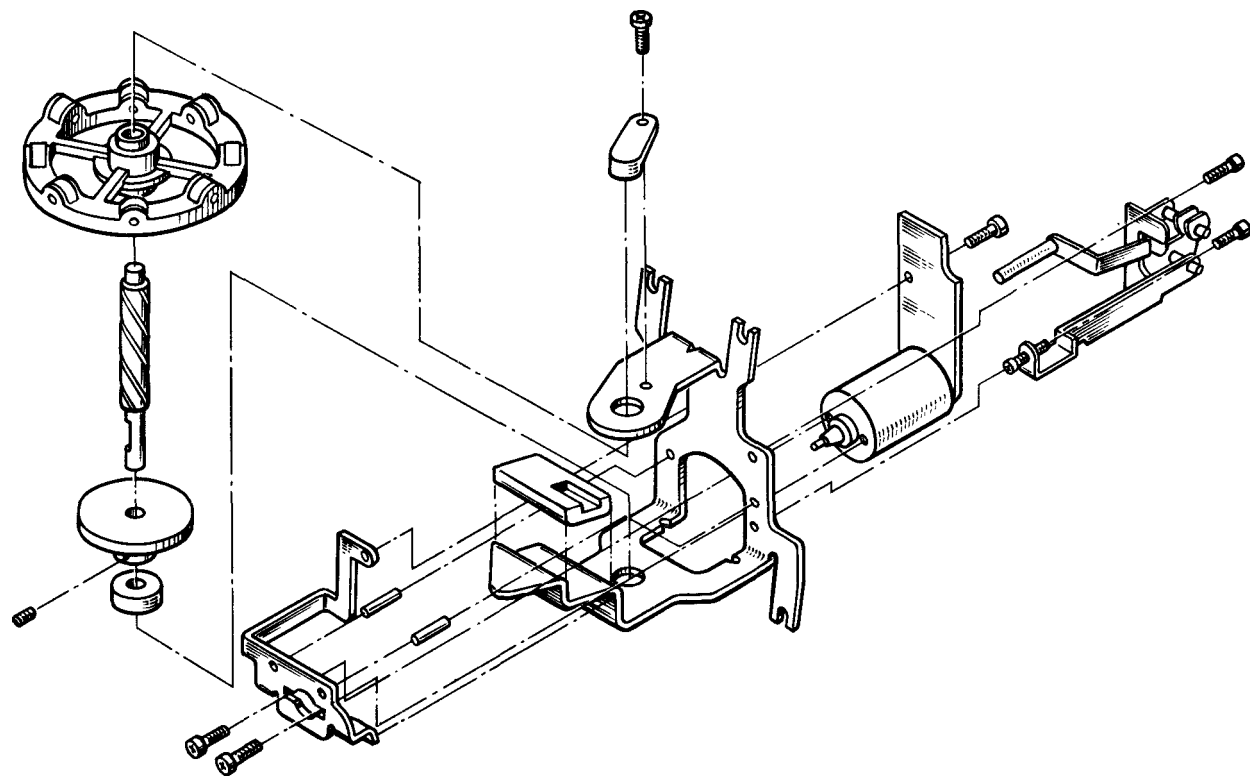


Рис. 1.9

Конструкция механизма для подачи по одному листов бумаги в печатающее устройство, выпускаемое на фирме ИБМ (г. Остин, шт. Техас)

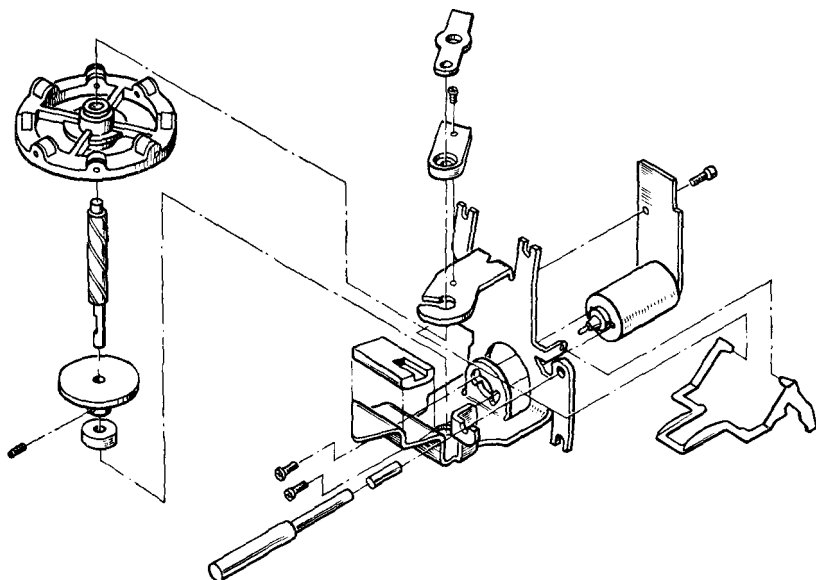


Рис. 1.10

Конструкция механизма, модернизированная для облегчения автоматизации сборки

на приобретение меньшего количества типов оборудования для автоматизации.

Для того чтобы посмотреть, как на практике реализуются принципы конструирования для автоматизации, сравним две конструкции одного и того же узла, изображенные на рис. 1.9 и 1.10. Узел предназначен для подачи листов бумаги в печатающее устройство, выпускаемое на заводе фирмы ИБМ, расположенном в г. Остин, штат Техас.

На рис. 1.9 приведена первоначальная конструкция узла, состоящего из 27 деталей. После переработки конструкции, результаты которой показаны на рис. 1.10, количество деталей сократилось до 14, а двигатель может быть установлен и закреплен за счет поступательного движения с поворотом. Следует обратить внимание на сокращение количества используемых винтов [4].

Из приведенного примера и иллюстраций принципов конструирования для автоматизации видно, что изменения конструкции, направленные на облегчение автоматизации, весьма выгодны и для неавтоматизированного производства. Действительно, после изменения конструкции узла печатающего устройства время на ручную сборку так сократилось, что применение робота стало невыгодным. Вероятно, самые большие выгоды в области автоматизации и роботизации получаются косвенно, от подготовки к этой работе.

1.3. СТАБИЛЬНОСТЬ ПРОЦЕССОВ

Несмотря на то, что гибкость современных промышленных роботов и другого оборудования, автоматизирующего производство, ослабила требования к стабильности процесса, относительная стабильность изделия и процесса является необходимой предпосылкой автоматизации. Даже рыночный спрос на изделие должен быть стабильным или возрадать, чтобы обосновать значительные капитальные вложения в оборудование для автоматизации.

Частые смены выпускаемых моделей в лучшем случае означают необходимость разработки дополнительного программного обеспечения для роботов и автоматических систем, а в худшем — потребуют дорогостоящего переоснащения и ликвидации дорогого жесткого автоматизированного оборудования.

Приемы статистической обработки данных эффективно используются для контроля за переменными процесса, особенно для контроля качества продукции и для оценки стабильности. Основное предположение, на которое опирается статистический контроль, заключается в том, что отклонения, наблюдаемые в процессе производства, не являются результатом просто случайности, а по сути представляют собой результат воздействия многих независимых случайных величин, каждая из которых оказывает малое влияние на процесс в целом. При этих условиях центральная предельная теорема [33] утверждает, что закон распределения суммы случайных величин будет иметь колоколообразную форму (гауссово, или нормальное, распределение). На рис. 1.11 изображена кривая нормального распределения случайной величины. Этой кривой аппроксимируется диаграмма распределения частот большой выборки значений нормально распределенной переменной величины. Кривая поэтому идеализирована и является теоретической, причем по вертикальной оси отложена плотность вероятности для всех значений переменных, откладываемых по горизонтальной оси.

Хотя одни нормально распределенные переменные изменяются в широком диапазоне, а другие меняются мало, любая нормально распределенная переменная может быть вписана в кривую, изображенную на рис. 1.11, соответствующим выбором масштаба горизонтальной оси. Выбор масштаба производится

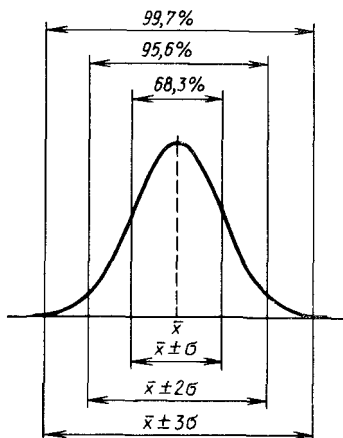


Рис. 1.11

Гауссово, или нормальное, распределение переменных величин процесса

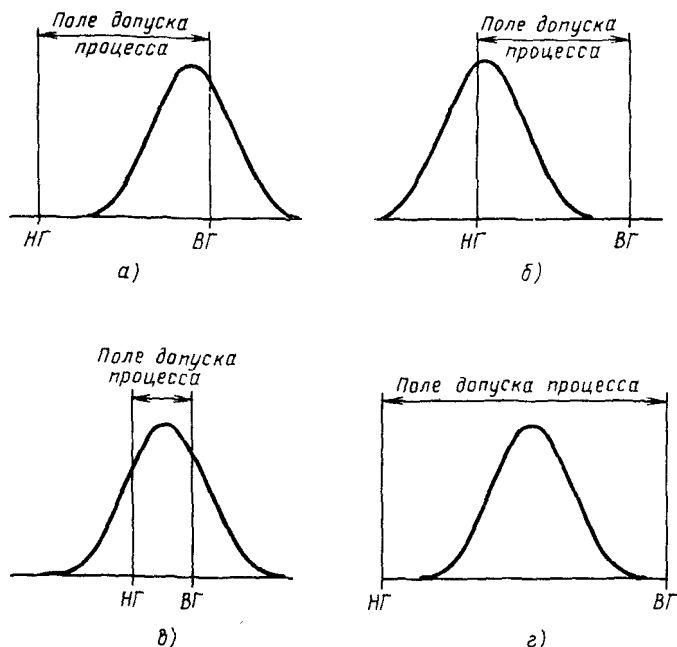


Рис. 1.12

Соотношения между переменной величиной процесса, имеющей нормальное распределение, и полем допуска:

а — процесс настроен слишком близко к верхней границе поля допуска и выходит за нее; *б* — процесс настроен слишком близко к нижней границе поля допуска и выходит за нее; *в* — разброс значений переменной процесса слишком велик, и он выходит из границ поля допуска; *г* — переменная величина процесса находится под контролем, и он вписывается в поле допуска

вычислением стандартного отклонения для выборки данных по формуле

$$s = \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 / (n - 1)},$$

где x_i — наблюдаемые значения; n — объем выборки; $\bar{x}$ — среднее арифметическое наблюдаемых значений; $\bar{x} = \sum_{i=1}^n x_i / n$.

Выборочное стандартное отклонение s служит оценкой теоретического стандартного отклонения σ для случайной нормально распределенной переменной. Доля плотности вероятности, приходящаяся на диапазон $\pm n\sigma$, откладываемый от среднего значения переменной, для нормального закона распределения известна. Это является ключом к использованию техники статистического контроля. Если наблюдаемое значение выходит за границы диапазона $\bar{x} \pm 3\sigma$, это обычно считается признаком принципиального изменения переменных, характеризующих процесс, а не случайным отклонением. Появление значения наблюдаемой величины,

выходящего за границы диапазона $\bar{x} \pm 3\sigma$, является случайностью и маловероятно, если $\bar{x}$ действительно является средним арифметическим значением нормально распределенной переменной. Это с очевидностью следует из рис. 1.11.

Соотношение между полем допуска на деталь и стандартным отклонением важно с точки зрения как качества продукции, так и возможности автоматизации процесса. На рис. 1.12 изображено несколько соотношений между полем допуска на деталь и переменными процесса, имеющими нормальное распределение. Только соотношение, изображенное на рис. 1.12, г, свидетельствует об удовлетворительном контроле за переменными процесса за счет применения автоматизации на заводе. Даже процесс, соответствующий рис. 1.12, г, может легко оказаться в положении, соответствующем рис. 1.12, в, если введены допуски, необходимые для применения промышленного робота или другого оборудования. Вероятность этого инженер по автоматизации должен оценить перед началом работы по роботизации или автоматизации.

Существуют способы для корректирования неудачных соотношений между изменениями в процессе и допуском на деталь, показанные на рис. 1.12. В промышленности зачастую для удовлетворения требований, предъявляемых автоматическим сборочным оборудованием, по повышенной цене закупается сделанные на заказ специальные высококачественные винты и другой крепеж. Единственной причиной дополнительных затрат на такие закупки является стремление облегчить автоматизацию изготовления, а совсем не повышение качества продукции. Средние значения и стандартные отклонения процесса должны быть заранее известны или определены и сравнены с установленными допусками. Сами допуски тоже должны быть критически проанализированы, чтобы выяснить, удовлетворяют ли они строгим требованиям автоматизации. Наряду с этим, для определения соответствия между допусками на деталь и возможностями процесса изготовления должны быть подробно изучены возможности предполагаемого к закупке оборудования для автоматизации.

1.4. ОБЕСПЕЧЕНИЕ НАДЕЖНОСТИ ОБОРУДОВАНИЯ

Надежные оборудование и процессы изготовления нужны всем, но автоматизация еще больше обостряет эту необходимость. С автоматизацией в производство приходят технологические процессы, не требующие внимания оператора. Автоматизация также приносит значительную взаимозависимость между установками, входящими в последовательность технологического процесса обработки.

В гл. 4 перечислены некоторые из драматических последствий отказов оборудования и заедания частей, что может полностью сокрушить проект автоматизации, в котором оборудование не обладает достаточно обусловленной надежностью. Этот вопрос

довольно сложен, но те, кто пережил последствия великих аварий электросети на северо-востоке в 1965 или 1967 г., прочувствовали, какие драматические последствия могут возникать в автоматических системах, объединяющих оборудование во взаимосвязанный процесс.

Когда отказывает один элемент, это может вывести из строя всю систему, если взаимозависимость между элементами критична. Процессы, выполняемые вручную, тоже не свободны от этих недостатков, но с введением автоматизации проблема обостряется. Инженеру по автоматизации необходимо оценить оборудование и производственный процесс на заводе с точки зрения надежности и удостовериться, что они обладают минимальным запасом, необходимым для их объединения в автоматизированную систему. Этот минимум зависит от числа элементов в цепи и от степени зависимости между элементами. Более подробно это освещено в гл. 4 и в последующих главах книги.

Тесно связана с надежностью и почти также важна для автоматизации ремонтпригодность оборудования. Важным понятием является эксплуатационная готовность — выражаемое в процентах отношение времени, в течение которого оборудование способно функционировать, ко всему времени наблюдения. Готовность оборудования является функцией того, как часто оборудование отказывает (надежности) и сколько времени потребуется на его восстановление (ремонтпригодности). При ручном серийном производстве допустим простой оборудования в течение 5 % времени (готовность 95 %). Оборудование может легко компенсировать простой, когда оно снова встанет в строй. В то же время для высокоавтоматизированной системы из взаимозависимого оборудования, готовность 95 % недопустима, поскольку это означает, что оборудование будет простаивать не 5 % времени, а большую часть времени!

1.5. СЕРИЙНОСТЬ ПРОИЗВОДСТВА

Хотя положение о том, что автоматизация оправдывает себя при высокой серийности производства, остается до сих пор справедливым, за последние десять — двадцать лет положение несколько изменилось. Автоматизация всегда требует капитальных вложений, которые должны амортизироваться за счет снижения себестоимости каждой единицы выпускаемой продукции. Однако для этого больше не обязательно выпускать в громадных количествах абсолютно одинаковые изделия, поскольку в распоряжении современного инженера по автоматизации имеется два основных типа автоматизации.

Жесткая автоматизация. При классическом типе автоматизации, характерном для 40-х и 50-х годов, используется специальное (жестко ориентированное) оборудование, обычно изготавливаемое по заказу и спроектированное для обеспечения произ-

водства конкретной продукции. С помощью жесткой автоматизации можно достигнуть очень высокой производительности выпуска, но ее стоимость весьма высока. Это сказывается очень болезненно, когда происходит смена модели или конструкции выпускаемого изделия. Специальная, или жесткая, автоматизация, как это следует из самого названия, плохо приспособлена к изменению выпускаемой продукции или технологии. Поэтому для обоснования решения об установке специального оборудования для жесткой автоматизации требуется по меньшей мере определенный объем производства при условии стабильности рынка, а гарантировать правильность такого решения могут лишь громадные объемы производства.

Хотя в 40-х и 50-х годах жесткая автоматизация была единственным типом автоматизации, она ни в коей мере не вышла из употребления сейчас. Рассмотрим, например, производство электроламп. Одна только фирма «Дженерал Электрик» выпускает приблизительно 2 млрд. электроламп в год [7]. При таком объеме выпуска легко обосновать применение высокопроизводительного оборудования жесткой автоматизации. В дополнение к большому объему производства выпуск электроламп достиг высокого уровня стабильности. Нет признаков того, что лампы накаливания выйдут из употребления в течение нескольких следующих десятилетий. Когда это произойдет, вместе с ними выйдет из употребления специальное оборудование для жесткой автоматизации стоимостью миллионы долларов.

Гибкая автоматизация. В противовес жесткой автоматизации современный инженер по автоматизации имеет еще один, новый тип автоматизации: гибкую автоматизацию. Мы не решились назвать этот тип автоматизации «мягкой» автоматизацией, потому что в нее входит как оборудование (hardware — термин, составной частью которого является слово «жесткий»), так и программно-математическое обеспечение (software — термин, включающий слово «мягкий»). Однако гибкая автоматизация действительно «мягкая» по сравнению с жесткой автоматизацией.

Самой заметной и определяющей чертой оборудования для гибкой автоматизации является возможность его программирования, и, следовательно, перепрограммирования. В настоящее время это обычно означает, что оборудование оснащено электронной цифровой вычислительной машиной. Первое оборудование, оснащенное программным управлением, можно рассматривать как оборудование для гибкой автоматизации, но это управление не базировалось на использовании ЭВМ. Возможность перепрограммирования оборудования для гибкой автоматизации является основным преимуществом по сравнению с жесткой автоматизацией. Для обоснования внедрения гибкой автоматизации нет необходимости иметь очень высокую серийность производства, поскольку после окончания изготовления партии изделий оборудование для гибкой автоматизации может быть использовано для

производства другой продукции. Оборудование может быть перепрограммировано или предварительно запрограммировано на выполнение ряда задач с вызовом соответствующей программы из подходящего программоносителя, например магнитной ленты или диска, при помощи поискового программно-математического обеспечения.

Гибкая автоматизация является основной темой данной книги. Гибкая автоматизация впервые появилась в начале 50-х годов вместе с внедрением станков с программным управлением типа NC. Позже станки с программным управлением стали оснащаться системами управления типа CNC, в состав которых входит малая ЭВМ, обычно микроЭВМ. В гл. 5 рассмотрены станки с программным управлением типа NC и CNC.

Промышленный робот является самым типичным образчиком гибкой автоматизации, а его разработка была естественным развитием концепции программного управления. В гл. 6—9 обсуждены вопросы, связанные с промышленными роботами. Гл. 6 является введением в эту тему. В гл. 7 объяснено, как промышленный робот может быть запрограммирован, что является ключом к его гибкости. На пути к внедрению робототехнических проектов могут встретиться подводные камни, и гл. 8 посвящена вопросу внедрения роботов. В гл. 9 рассмотрены примеры, иллюстрирующие применение роботов в промышленности, дана оценка различным случаям применения.

Промышленные роботы, как и все оборудование для гибкой автоматизации, должны быть оснащены системами, реагирующими на изменение условий, принимающими логические решения и выполняющими соответствующие действия. Эти системы называются промышленными логическими системами управления и описаны в гл. 10. В гл. 11 объяснено, как составлять схемы и анализировать работу логических систем. Оборудованием для внедрения промышленных логических систем управления являются ПК, описанные в гл. 12. Неспециалисты недостаточно хорошо понимают программируемые контроллеры, а сенсационные промышленные роботы, конечно, несколько затмили их. В то же время многие промышленные роботы сами управляются стандартными ПК. Кроме того, инженер по автоматизации может сконструировать и создать у себя на предприятии собственный робот, используя стандартный ПК в качестве его «мозга», что бывает даже практичнее.

Еще до разработки ПК для управления промышленными процессами применялись цифровые вычислительные машины. Большинство таких ЭВМ управляют непрерывными процессами, такими, какие имеют место на химических, нефтеочистительных, цементных заводах и бумажных фабриках, но производство дисковой продукции также может управляться вычислительными машинами, предназначенными для управления технологическими процессами. Большинство инженеров знакомы с пакетным режи-

мом обработки данных и программированием для ЭВМ, но программирование для ЭВМ, управляющих технологическими процессами, и их использование имеет несколько принципиальных отличий от обычной обработки данных на ЭВМ. Эти отличия рассмотрены в гл. 13, так что инженер по автоматизации сможет объединить промышленные роботы и оборудование для гибкой автоматизации во всеохватывающую заводскую систему, управляемую от ЭВМ.

Наконец, в гл. 14 выделен вопрос о микропроцессорах, выступающих в качестве движущей силы революции промышленных роботов и гибкой автоматизации. Создание микропроцессоров явилось одним из ключевых открытий столетия. Среди других потенциальных кандидатов можно выделить изобретение самолета, открытие атомной энергии, изобретение телевидения, полупроводников, ЭВМ и лазера. Микропроцессор является самым последним поступлением в этот элитарный список, и, как показано в гл. 14, есть основание верить, что его влияние на человечество затмит все остальные. Сейчас микрокомпьютеры служат питательной средой роботизации, и это обещает сделать начавшуюся новую эру эрой автоматизации производства и эрой инженеров, ее создающих.

В гл. 15 изложены результаты последних исследований, которые еще не полностью реализованы в производстве. В нее включены некоторые соображения о будущих тенденциях, наряду с размышлениями об этике и о том, куда мы движемся в свете взаимоотношений между роботами, автоматизацией и человечеством.

В ы в о д ы

В этой главе сделана попытка достичь двух целей: первая — дать возможность инженеру по автоматизации уяснить и оценить необходимость подготовки предприятия к автоматизации, вторая — дать общий обзор проблемы, чтобы подготовить читателя к восприятию остального материала книги.

Роботы и автоматизация производства едва ли нуждаются в рекламе. В сфере производства развитых стран, таких, как США, под давлением международной конкуренции уже ощутили срочную необходимость в автоматизации. Эту срочность можно сравнить с той, которую ощущает жертва удава. Удав потихоньку сжимает свои объятия каждый раз, когда жертва расслабляется и выдыхает, что характерно для экономических подъемов с обычным для них ростом заработной платы и терпимостью в вопросе о себестоимости.

Вселяющая ужас срочная необходимость принимать какие-то меры, вызываемая сжимающими кольцами удава мировой конкуренции, заставляет некоторых наивных и импульсивных инженеров и руководящих работников промышленности безоглядно заниматься проблемами автоматизации без предварительной под-

готовки изделий и технологии. Вероятно, наибольшая польза, извлекаемая из изучения возможностей применения в производстве промышленного робота, заключается в тщательной проработке конструкции изделия, оценке стабильности технологии и надежности имеющихся на производстве станков и другого оборудования. Такая предварительная проработка конструкции, анализ и совершенствование изделия и процесса могут быть настолько эффективными, что в конечном счете исключат необходимость применения роботов или другого автоматизированного оборудования.

Особого внимания при оценке необходимости использования роботов и автоматизации технологических процессов требуют объемы производства. Обычно считается, что для обеспечения успеха автоматизации необходимы огромные объемы производства, но эта старая истина начинает терять под собой почву. Современные научно-технические достижения позволили автоматизировать производства со среднесерийным выпуском изделий и использовать станки с ЧПУ различной степени гибкости для самых разных процессов. Гибкая автоматизация — главная тема этой книги, а как известно, основными элементами такой автоматизации являются промышленные роботы и управляющие ими промышленные логические системы управления.

Упражнения и вопросы

1.1. Почему применение промышленных роботов и автоматизация современных производств необходимы так срочно?

1.2. Можете ли вы назвать создание промышленных роботов крупным открытием в технологии? Поясните ответ.

1.3. Что является альтернативами автоматизации в условиях международной конкуренции?

1.4. Сформулируйте правила конструирования изделий, облегчающие возможность применения промышленных роботов и осуществления автоматизации производства.

1.5. Облегчает или усложняет автоматизацию симметрия деталей? Поясните, почему?

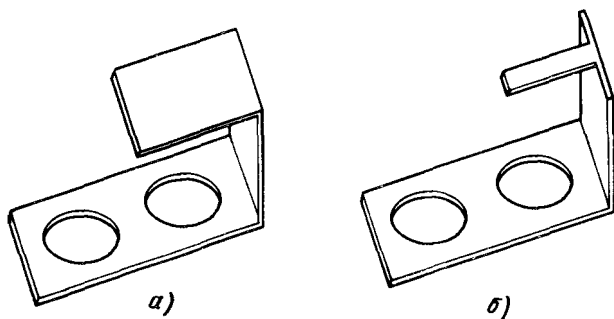


Рис. 1.13

Альтернативные конструкции изделия из упражнения 1.6

1.6. Рассмотрите, с точки зрения пригодности для автоматизации, конструкции двух деталей (рис. 1.13) при условии, что одинаковы их качество, стоимость и эффективность использования. Какая конструкция более пригодна для автоматизации? Почему?

1.7. Захват робота может поднимать детали с размерами $2 \pm 0,010$ дюйма ($50,8 \pm 0,254$ мм). Стандартное отклонение для этого размера, обеспечиваемое технологией, составляет 0,005 дюйма (0,127 мм). Удовлетворяет ли технология требованиям роботизации? Если средний размер деталей в партии составляет точно 2 дюйма (50,8 мм) и отклонения размера являются нормально распределенными, то какую часть деталей партии в процентах робот не может поднять?

1.8. Предположим, что в предыдущем случае средний размер деталей из-за износа инструмента увеличился на 0,01 дюйма (0,254 мм), а технология осталась прежней. Какой процент деталей партии робот сможет поднять при новом среднем размере, равном 2,01 дюйма (51,054 мм)?

2. ЭЛЕМЕНТЫ АВТОМАТИЗАЦИИ

Роботы и заводы-автоматы, управляемые ЭВМ, являются чудесами века, но автоматизация начинается не с роботов, а на рабочем месте. Для успешного решения проблемы повышения производительности обычно требуется выявить и устранить «узкие места», обеспечить эффективное использование рабочих мест, где решающую роль до сих пор играют люди, а не машины. Используя обширный набор хитроумных устройств — кубиков автоматизации — можно шаг за шагом решать вопросы более эффективного обнаружения, перемещения, позиционирования, ориентации, обработки и сборки изделий. До тех пор, пока работа технологического оборудования не проанализирована достаточно тщательно, а в состав установки не включены основные компоненты автоматизации, ее оснащение таким экзотическим оборудованием, как робот, проблематично. Ведь и сами промышленные роботы сконструированы из базовых компонентов автоматизации. Целью этой главы является изучение и классификация базовых компонентов автоматизации на уровне технологического оборудования. Эти компоненты будут появляться затем в примерах на протяжении всей остальной книги. Некоторые из объектов, изучаемых в данной главе, например зубчатые колеса, выключатели, двигатели, окажутся знакомыми почти всем читателям, но информация о них здесь приводится с тем, чтобы убедить всех в возможности их использования для автоматизации. К тому же, здесь изучается не конструкция этих устройств, а нечто другое. Наша цель — установить область применения и посмотреть, как их можно объединять в системы для повышения производительности.

В связи с разнообразием компонентов автоматизации необходимо ввести хотя бы грубую их классификацию. Классификация получится грубой потому, что наиболее употребительные компоненты могут попадать в разные категории в зависимости от их применения. Можно, однако, разделить все компоненты на следующие четыре класса: датчики, анализаторы, исполнительные механизмы и приводы.

Примерная взаимосвязь этих четырех широких категорий представлена на рис. 2.1. Следует отметить, что оператор на

рис. 2.1 — человек, а не робот. Промышленные роботы являются частью автоматизированной системы (верхняя часть рисунка). Промышленный робот сам по себе является системой, включающей компоненты всех четырех основных классов.

2.1. ДАТЧИКИ

Датчики (рис. 2.1) являются первым звеном, связывающим типичную автоматизированную систему и обычный технологический процесс. Они собирают информацию об обрабатываемом оборудовании, обрабатываемой детали и о человеке-операторе, если он имеется. Может показаться странным, что автоматизированная система получает информацию от оператора, однако этот канал, без сомнения, является самым важным каналом связи между автоматизированной системой и реальным миром. Первый датчик, с которым мы познакомимся, относится именно к таким датчикам.

Ручные выключатели. Самым известным датчиком является ручной выключатель. Большинству не приходит в голову, что выключатель является датчиком, однако выключатель связывает лампу с тем, кто хочет ее включить или выключить. Таким же образом автоматизированная система связана с оператором, который хочет включить или выключить ее, чтобы отрегулировать

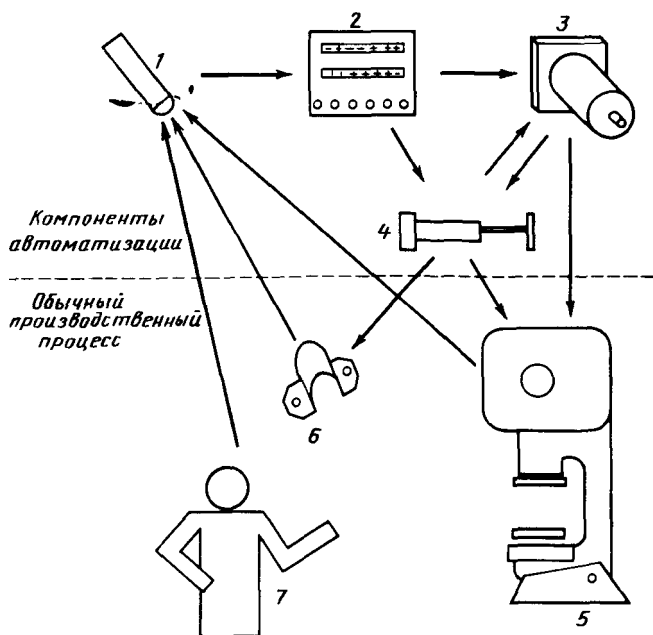


Рис. 2.1

Взаимосвязь компонентов автоматизации и технологического процесса:

1 — датчики; 2 — анализаторы; 3 — исполнительные механизмы; 4 — приводы;
5 — технологическое оборудование; 6 — деталь; 7 — оператор

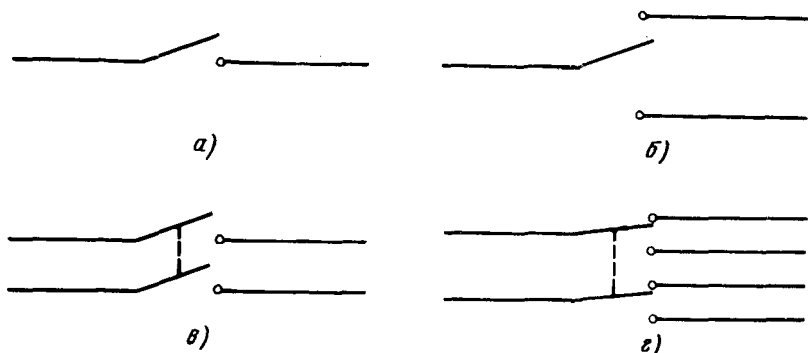


Рис. 2.2

Обозначения электрических выключателей:

а — однополюсный одноходовой выключатель; *б* — однополюсный выключатель для двух цепей; *в* — двухполюсный выключатель для одной цепи; *г* — двухполюсный выключатель для двух цепей

автоматический цикл. Практически все ручные выключатели являются электрическими, и мы не будем обращать внимание на исключения.

Включение выключателя означает, что цепь, которой он управляет, замкнута. Когда выключатель находится в положении «Выключено», цепь разомкнута. Большинство выключателей имеет два стабильных состояния — включенное (контакты замкнуты) и выключенное (контакты разомкнуты). Многие выключатели, однако, имеют лишь одно стабильное состояние. Такие выключатели снабжены пружиной, которая возвращает их в исходное (нормальное) состояние, когда внешняя сила не действует. В нормальном состоянии контакты могут быть замкнутыми или разомкнутыми, поэтому для описания контактов выключателей ввели термины замыкающие и размыкающие.

Обыкновенный стенной выключатель является примером рычажного переключателя и в простейшем случае только замыкает или размыкает одну электрическую цепь. Такие выключатели называются однополюсными выключателями для одной цепи, а их изображение на схемах приведено на рис. 2.2, *а*. Термин однополюсный для одной цепи предполагает существование многополюсных выключателей для нескольких цепей, и их работу необходимо изучить, чтобы понять название простейшего выключателя. Рычажные выключатели бывают самых различных конструкций. На промышленном оборудовании обычно устанавливают двухпозиционные выключатели, как те, что на электрических лампах, но бывают и трехпозиционные, отключающие цепь при их среднем положении. Это позволяет управлять двумя различными цепями.

Отклонение рычажка выключателя влево замыкает одну цепь, отклонение вправо — другую, в центральном положении обе цепи разомкнуты. Даже когда рычажный выключатель имеет

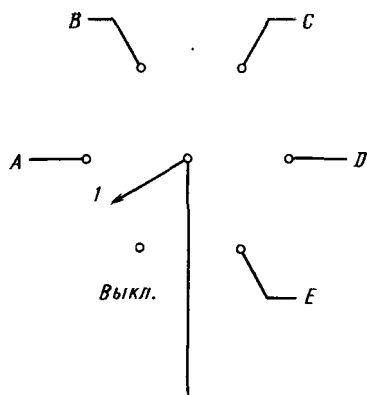


Рис. 2.3
Поворотный выключатель:
 I — скользящий контакт

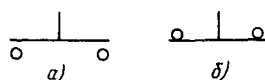


Рис. 2.4
Электрические кнопки:
 a — с замыкающими контактами, b — с размыкающими контактами

только два положения, он может управлять двумя цепями. На рис. 2.2, б показано, как такой выключатель с подведенным к нему общим проводом обеспечивает включение двух различных цепей. Такой выключатель называется однополюсным выключателем для двух цепей. Выключатель является однополюсным из-за наличия общего провода, но он обслуживает две цепи, подключая их попеременно к одному полюсу. Наряду с этим можно иметь и два провода, подводимых к выключателю, что позволило бы ему за одно движение соединить две цепи, не имеющие общего полюса. Выключатель, обеспечивающий такое соединение, называется двухполюсным выключателем для одной цепи, а его схема приведена на рис. 2.2, в. В еще одной конструкции каждый контакт с одной стороны выключателя может быть соединен с одним из двух контактов, имеющих с другой стороны. Такой выключатель называется двухполюсным выключателем для двух цепей, а его схема показана на рис. 2.2, г.

Можно добавлять сколько угодно полюсов, но использование рычажных выключателей более чем для двух цепей становится нецелесообразным. Для большого количества цепей более удобным является поворотный выключатель. На рис. 2.3 показан поворотный выключатель, который можно классифицировать как однополюсный выключатель для пяти цепей, однако обычно его называют просто шестипозиционным вращающимся переключателем. На каждой из шести позиций имеется механический фиксатор. Следует обратить внимание на то, что позиций шесть, а цепей всего пять. Одна позиция используется для отключения всех цепей. Рукоятки управления без фиксированных позиций обычно служат для бесступенчатого управления переменными параметрами автоматизированного процесса, которые могут принимать любое значение в некотором непрерывном диапазоне. Такими рукоятками можно менять напряжение, сопротивление или емкость электрической цепи, меняя тем самым такие величины, как скорость, положение или интенсивность автоматизированного процесса. Однако для того, чтобы автоматизировать производственные

системы и использовать роботы, понимать, как работают выключатели и другие дискретные селективные устройства значительно важнее, чем понимать работу устройств управления непрерывного типа.

Техника безопасности при эксплуатации роботов и других автоматизированных систем требует, чтобы большинство выключателей на панели ручного управления имели только одно стабильное положение: «Выключено». В этом случае для того чтобы выключатель оставался включенным, оператор должен непрерывно на него воздействовать. Достоинством выключателей с одним стабильным положением является то, что с его помощью в любой момент можно изменить состояние системы. Чтобы оценить это достоинство, предположим, что сложная автоматизированная система управляется одним рычажным выключателем с фиксированными включенным и выключенным положениями. Для отключения в аварийной ситуации оператору такой системы пришлось бы отыскивать этот единственный выключатель. В то же время при наличии мгновенного выключателя систему можно включить в одном месте, а затем выключать мгновенными выключателями, установленными в удобных местах. В промышленности эти мгновенные выключатели, располагаемые в разных местах, называются аварийными выключателями или просто кнопками «Стоп».

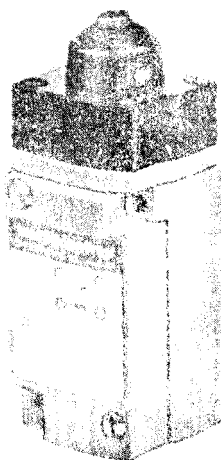
Мы до сих пор не обращали внимания на то, что оператор должен воздействовать на выключатель, чтобы поддерживать его во включенном состоянии. Такой характер работы выключателя может показаться неудобным, но следует отметить, что речь идет о выключателе, а не о самой системе. Для поддержания системы во включенном состоянии и для немедленного ее выключения любым из аварийных выключателей можно воспользоваться дополнительными логическими цепями. О том, как это делать, и о других областях применения логических систем будет рассказано в гл. 10.

Наиболее известным видом мгновенных выключателей являются нажимные кнопки. Они бывают «замыкающими» и «размыкающими». На рис. 2.4 показаны стандартные кнопки обоих типов. Существует и другой тип кнопочных выключателей, имеющих два стабильных состояния. Они предназначены для перевода цепи из любого исходного в противоположное состояние. Эти кнопки напоминают рычажные выключатели, но в отличие от них по внешнему виду кнопок непонятно, в каком из двух возможных состояний (разомкнутом или замкнутом) они сейчас находятся.

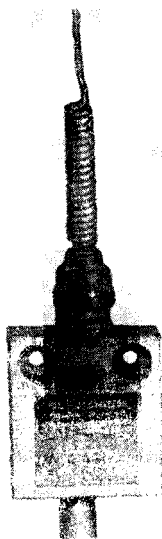
Концевые выключатели. Концевые выключатели, как и ручные, срабатывают от механического воздействия, но в отличие от них воздействие оказывает производственный процесс, материал или сама автоматизированная система, без участия оператора. Моделей и видов концевых выключателей существуют буквально



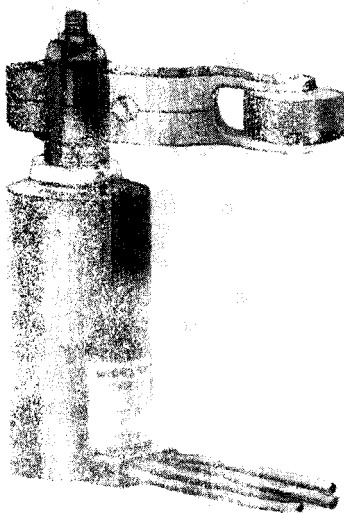
а)



б)



в)



г)

Рис. 2.5

Конструкции стандартных концевых выключателей:

а — стандартный выключатель с поворотным рычагом; б — стандартный выключатель с толкателем; в — концевой микровыключатель; г — концевой выключатель, применяемый в авиации

тысячи, даже больше, чем ручных выключателей. Это происходит из-за того, что конструкции выключателей должны по размерам, ходу рычага, силе воздействия и жесткости точно соответствовать требованиям, предъявляемым громадным количеством конкретных

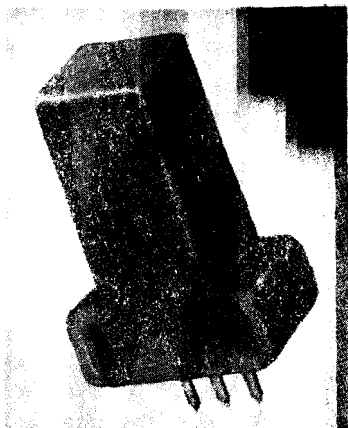
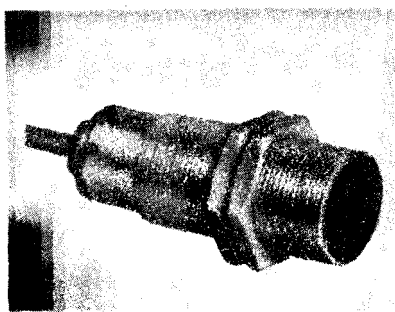
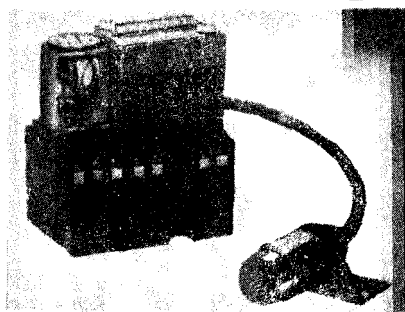


Рис. 2.6
Бесконтактные выключатели:
Слева вверху — реагирующие на присутствие
объектов из черных металлов; внизу — ре-
агирующие на присутствие объектов из лю-
бых металлов



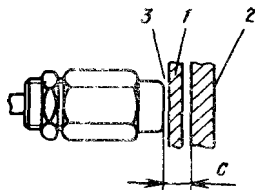
случаев автоматизации. Ручные выключатели проектируются под операторов, которые имеют относительно одинаковые физические характеристики. Кроме того, конструкции концевых выключателей часто должны обеспечивать их работоспособность в таких тяжелых производственных условиях, в которых работа операторов не допускается. Концевые выключатели приводятся в действие рычагами, штифтами, кнопками, плунжерами, роликами, контактными пружинами и вообще всем, что изобретатель может придумать при конструировании автоматической системы. На рис. 2.5 показаны конструкции концевых выключателей.

В роботизированных системах концевые выключатели применяются как на самих роботах, так и в периферийном оборудовании. Концевые выключатели могут использоваться для ограничения перемещения руки робота по любой из осей координат. Когда рука достигает до предела установленного перемещения по данной оси, электрическая цепь размыкается (или замыкается), что прекращает подачу энергии к соответствующему двигателю непосредственно или через систему управления роботом. В периферийном оборудовании концевой выключатель устанавливается, например, на двери барьера, ограждающего рабочую зону, и действует как аварийный выключатель, если кто-нибудь войдет.

Рис. 2.7

Барьер из цветного металла «прозрачен» для выключателя, реагирующего на присутствие объектов только из черных металлов:

1 — барьер; 2 — обнаруживаемый объект; 3 — зазор



Бесконтактные выключатели. Некоторые выключатели не требуют физического контакта с объектом, чтобы ощутить его присутствие. Воспринимать световое излучение от объекта им тоже для этого не требуется, поэтому такие выключатели называются бесконтактными. Человек не обладает такими возможностями, поэтому установка на роботах бесконтактных выключателей дает им определенные преимущества по сравнению с оператором-человеком.

На рис. 2.6 представлены два типа бесконтактных выключателей, первый из которых (рис. 2.6, а) реагирует в пределах своей чувствительности на присутствие объектов, изготовленных из черных металлов. Другой выключатель (рис. 2.6, б) реагирует на присутствие объектов как из черных, так и из цветных металлов. На первый взгляд, инженер, занимающийся автоматизацией или внедрением роботов, всегда предпочтет второй выключатель, реагирующий на все металлы. Однако вообразите применение выключателя для сортировки одинаковых объектов, отличающихся только по содержанию черных металлов. Другим преимуществом выключателя, который реагирует только на черные металлы, является то, что для него преграды из цветных металлов являются «прозрачными», поэтому он чувствует присутствие объектов из черных металлов за барьером из цветных металлов. На рис. 2.7 показан пример использования свойства бесконтактного выключателя чувствовать объекты из черных металлов за алюминиевым барьером.

Существуют физические явления, на основе которых работают выключатели, реагирующие на любые объекты — как металлические, так и неметаллические. В одном из типов выключателей имеется специальная антенна, работающая в диапазоне радиочастот. Антенна принимает сигнал, передаваемый другой целесобразно расположенной антенной, а помещение любого объекта в создаваемое поле вызывает помехи приему. Эти помехи улавливаются первой антенной, и при достижении ими определенного уровня выключатель срабатывает. К недостаткам (а для некоторых случаев к достоинствам) этих выключателей следует отнести то, что чувствительность антенны связана с электрическими свойствами материала объекта. Имеют значение также размеры объекта, поэтому систему можно превратить в подобие избирательной для определенных изделий.

Другим типом бесконтактных выключателей, реагирующих на неметаллические объекты, являются сонары. Сонары излучают

волны давления и улавливают отраженные от объектов волны. Когда частоты волн находятся в диапазоне частот, улавливаемых человеческим ухом, волны называются звуковыми. Большинство сонаров работают в ультразвуковом диапазоне, где частоты выше.

Работа весьма совершенных бесконтактных выключателей основывается на эффекте Холла, заключающемся в появлении небольшого напряжения в поперечном сечении проводника, по которому проходит электрический ток, если проводник помещен во внешнее магнитное поле. Значение этого напряжения, называемого напряжением Холла, пропорционально плотности потока магнитного поля, перпендикулярного направлению тока. Пропорциональность позволяет с помощью бесконтактных выключателей не только обнаруживать присутствие объекта, но и определять относительное расстояние до него.

Фотоэлектрические датчики. Более распространенными, чем бесконтактные выключатели, являются датчики, реагирующие на световое излучение, а именно фотоэлектрические датчики. Следует выделить два основных направления применения этих датчиков. Во-первых, можно просто использовать фотоэлемент для контроля за светом, естественно излучаемым объектом в процессе работы. Хорошим примером является использование фотоэлементов для включения освещения при наступлении сумерек и выключения его с рассветом. Рост стоимости потребляемой энергии привлек всеобщее внимание к этому типу автоматических систем.

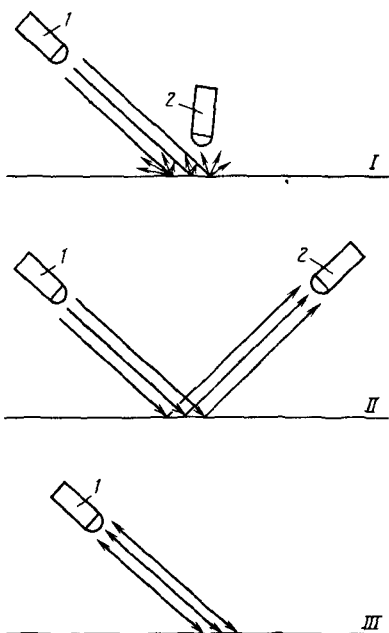
Во-вторых, можно использовать луч света, создаваемый искусственным источником. Основной задачей здесь является обнаружение объекта на пути прохождения луча. Источник света может быть установлен отдельно или встроен в датчик. Во втором случае возникает необходимость в естественном или искусственном отражателе для возврата луча к датчику.

Отражающие поверхности, используемые в фотоэлектрических системах, бывают трех типов: рассеивающие, зеркальные и обратного отражения (рис. 2.8). Самым дешевым и распространенным видом отражающих поверхностей являются рассеивающие поверхности. Даже обычный объект белого цвета имеет свойства рассеивающей поверхности, поэтому он отражает свет, а не изображение. Рассеивающие отражатели рассеивают так много света, что только малая его доля возвращается к фотоэлектрическому датчику. В связи с этим экономия, полученная за счет применения рассеивающих поверхностей, может обернуться убытком, если речь идет о применении более чувствительного и, возможно, более совершенного датчика. Фотоэлектрические системы, использующие рассеивающие отражающие поверхности, к тому же, более подвержены влиянию помех.

Зеркальные поверхности чаще всего ассоциируются со словом «отражающий». К ним относятся зеркала и очень хорошо полированные поверхности. На зеркальные отражающие поверхности

Рис. 2.8

Отражающие поверхности, используемые в фотоэлектрических системах:
 I — рассеивающая поверхность; II — зеркальная поверхность; III — поверхность обратного отражения; 1 — источник света; 2 — датчик



распространяется физический закон, по которому угол падения равен углу отражения. Очевидно, что источник света и датчик, взаимодействующие через зеркальные отражающие поверхности, должны быть более точно направленными, чем при использовании рассеивающих поверхностей. Для систем, в которых источник света и датчик конструктивно объединены, плоскость зеркальной отражающей поверхности должна быть перпендикулярной к направлению луча света. В противном случае луч будет потерян. Это может быть преимуществом или недостатком в зависимости от целей, которые ставит перед собой конструктор. Из всех трех перечисленных типов поверхностей поверхности обратного отражения являются наиболее сложными и дорогими. Эти поверхности способны отражать назад к источнику большую часть света вне зависимости от угла падения луча. Поверхности обратного отражения в основном нарушают закон равенства углов падения и отражения, за исключением случая перпендикулярности поверхности лучу от источника.

Общеизвестным примером поверхности обратного отражения является красный отражатель, устанавливаемый на заднем крыле велосипеда. Хотя отражатель не подсвечивается изнутри, он ярко светится под светом фары велосипеда, едущего сзади. Свет концентрируется в луч, возвращаемый назад к источнику, если тот находится в определенной зоне. Некоторые дорожные сигналы, полосы разметки, поверхности отражающих лент и киноэкранов покрыты слоем мелких бусинок, что заставляет часть света от источника отражаться обратно. По крайней мере частично описанная способность поверхностей обратного отражения может улучшить характеристики фотодатчиков. Очевидно, что точная установка в этом случае не требуется, чувствительность фотодиода может быть ниже, чем при рассеивающей отражательной поверхности, так что поверхности обратного отражения в некоторой степени соединяют достоинства зеркальных и рассеивающих, проигрывая им в цене.

Фотоэлектрические датчики очень полезны, но инженер по автоматизации должен избегать их применения в условиях, которые могут привести к их загрязнению, создавать помехи в работе или даже повредить фотоэлектрическую систему. Случайное освещение датчика, например, фонарем в руках рабочего может привести к его срабатыванию. Попадание на датчик света дуги электросварки может привести к более печальным последствиям, даже совсем вывести датчик из строя. Повредить фотоэлектрическую систему может также высокая температура, сопутствующая технологическому процессу. Если окружающая среда в производстве загрязнена твердыми частицами или испарениями, на линзах и зеркалах может образоваться налет пыли или конденсата. Необходимость очистки этих поверхностей вручную может лишить автоматизацию всех преимуществ. Собираясь проводить автоматизацию, необходимо учитывать возможность возникновения вибраций в связи с работой соседнего оборудования и принять соответствующие меры. Вибрации могут нарушить правильность взаимного положения источника света и датчика и снизить надежность ламп, особенно ламп накаливания.

Большинство проблем, возникающих при применении фотоэлектрических датчиков, поддается решению, особенно если инженер по автоматизации с самого начала готов к этому. Например, посторонние источники света могут быть экранированы, а направление луча света может быть изменено с помощью зеркал. Проблемы запыления и осаждения конденсата могут быть решены с помощью обдувки сжатым воздухом. Еще одним приемом, применяемым при неблагоприятных окружающих условиях, является применение устройств в преднамеренно суженном диапазоне. Например, если система рассчитана на диапазон 8—10 футов (2,5—3 м), используйте ее на расстоянии 2—3 футов (0,6—0,9 м). Для решения проблемы нарушения правильности взаимного положения датчика и источника вследствие вибраций могут быть применены поверхности обратного отражения. Иногда можно найти выход, используя инфракрасное излучение или волоконную оптику, о чем будет рассказано в следующих разделах.

Датчики инфракрасного излучения. Иногда полезно иметь возможность обнаруживать электромагнитное излучение за пределами видимого диапазона. Датчики инфракрасного излучения реагируют на излучение в диапазоне частот, примыкающем к видимому со стороны красного цвета. Горячие объекты излучают инфракрасный свет, и поэтому датчики инфракрасного излучения применяются для обнаружения источников теплоты. Реагирование на «естественное» инфракрасное излучение объектов используется в работе систем, диагностирующих неисправности. Датчики инфракрасного излучения, применяемые совместно с искусственными источниками инфракрасного света, еще более полезны при определении присутствия или отсутствия объектов, чем фотоэлектрические датчики. Некоторые преимущества использования источ-

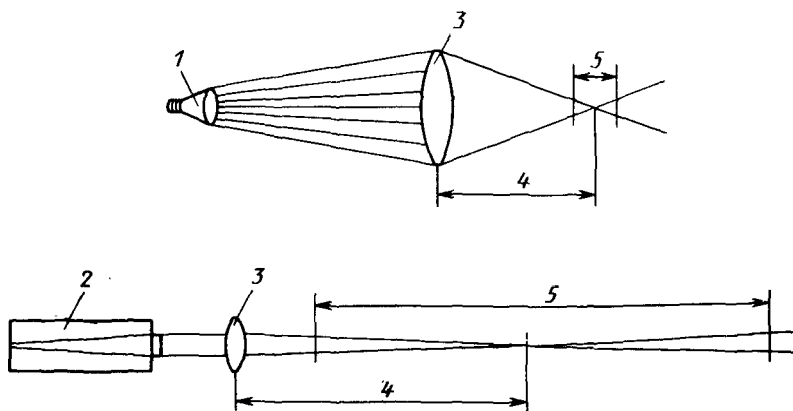


Рис. 2.9

Сравнение рабочих диапазонов, обеспечиваемых обычным и лазерным источниками света, при определении местонахождения объектов на расстоянии:

1 — обычный источник света; 2 — лазерный источник света; 3 — линза; 4 — фокусное расстояние; 5 — рабочий диапазон

ников и приемников инфракрасного излучения по сравнению с обычными фотоэлектрическими датчиками проистекают из невидимости инфракрасного излучения. Кроме того, датчики инфракрасного излучения практически не подвержены воздействию помех в диапазоне видимого света, что является очевидным их достоинством.

Наибольшее распространение получила методика использования модулированного инфракрасного излучения, когда источник генерирует пульсирующее излучение, обладающее повышенной интенсивностью, а датчик настроен на частоту модуляции. Модуляция излучения расширяет диапазон использования таких в принципе слабых источников инфракрасного излучения, как светодиоды. Светодиоды являются твердотельными приборами и поэтому обладают преимуществами перед лампами накаливания в потребляемой мощности и надежности.

Оптоволоконные устройства. Полезным дополнением к фотоэлектрическим или инфракрасным системам являются волоконные световоды, представляющие собой гибкие стеклянные или пластмассовые трубки, по которым можно пропускать лучи света, меняя их направление. Когда используются пучки волокон, могут быть переданы целые изображения объектов. Однако обычно в системах автоматизации применяются отдельные волокна для передачи луча, присутствие или отсутствие которого определяет датчик.

Преимуществом волоконной оптики является ее удивительная пропускающая способность. Она настолько велика, что при телефонизации оказалось выгодным трансформировать исходные электрические сигналы в модулированные световые для передачи их по световодам с последующей обратной трансформацией.

Лазеры. Прежде чем покончить с вопросом о фотоэлектрических датчиках, необходимо рассмотреть свет, испускаемый лазерами. Этот свет представляет собой концентрированный, усиленный луч коллимированного света. Такие лучи способны переносить на расстояние громадное количество энергии, сосредоточивая ее на маленькой площади, что с очевидностью предполагает его промышленное применение. В автоматизированных системах лазеры используются как источники очень длинных и калиброванных лучей света. Рис. 2.9 иллюстрирует преимущества узких концентрированных лучей, испускаемых лазерами. Свойства лазерных лучей позволяют с их помощью прекрасно обнаруживать малые объекты, способные прервать их на большом переменном расстоянии. Присутствие или отсутствие луча за объектом может быть использовано как логический вход в автоматизированную систему управления. Наряду с этим лазер может быть хорошим инструментом для измерения размеров.

2.2. АНАЛИЗАТОРЫ

Информация, воспринятая датчиком автоматизированной системы, должна быть зарегистрирована, ее содержание проанализировано, после чего система принимает решение о дальнейших действиях. И этот процесс, и исполнительные устройства слишком сложны, чтобы детально обсуждать их в данной главе. Но некоторые компоненты системы заслуживают упоминания здесь, они в дальнейшем помогут читателю в понимании составных частей станков с ЧПУ, роботов, программируемых контроллеров и других устройств автоматизированных систем.

ЭВМ. Цифровые ЭВМ являются первостепенным и в высшей степени универсальным средством анализа входных сигналов автоматизированных систем, поскольку возможности программирования для обработки данных безграничны. Продолжающаяся миниатюризация ЭВМ вместе со снижением цен, вызванным последними технологическими достижениями, позволяет расширить сферу их применения в автоматизации. Ни одно другое достижение не сказалось так на развитии промышленных роботов. Применение ЭВМ в качестве анализаторов настолько важно, что оно подробно рассмотрено в гл. 13 и 14. Важность использования ЭВМ для управления роботами и другим автоматизированным оборудованием подтверждается всем содержанием этой книги.

Счетчики. В автоматизированных системах зачастую требуется определять, сколько разных изделий накопилось в приемном устройстве или прошло обработку. Эта функция может быть осуществлена либо внутренними средствами с помощью управляющей ЭВМ или программируемого контроллера, либо извне с помощью специальных устройств, называемых счетчиками. Счетчик может быть механическим, но в большинстве автоматизи-

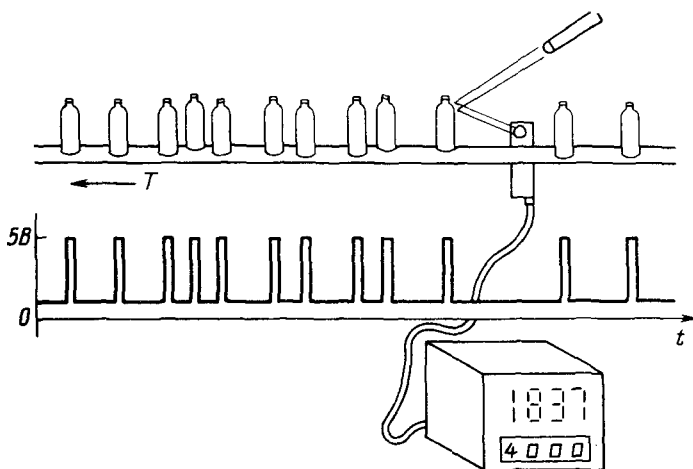


Рис. 2.10

Автоматическая фотоэлектрическая система для счета бутылок, движущихся на конвейере в направлении T

рованных систем применяются твердотельные электронные счетчики. Если счетчик представляет собой отдельное устройство, он обычно имеет индикацию, показывающую последний результат счета.

Обычно подсчитывается число импульсов напряжения, генерируемых датчиком, реагирующим на наличие физического объекта, с которым имеет дело автоматизированная система — например, стеклянных бутылок, проходящих по конвейеру. Изогнутое стекло обладает некоторыми зеркальными свойствами и при определенном положении бутылки она отразит луч прямо на датчик, как показано на рис. 2.10. Датчик преобразует импульс света в импульс электрического напряжения, который будет передан в счетчик.

Обратите внимание, что импульсы, изображенные на рис. 2.10, распределены неравномерно. Допускается также, чтобы сами пики варьировались по ширине (времени прохождения импульса) и амплитуде (напряжению). В определенных пределах это совершенно не имеет значения, потому что большинство промышленных счетчиков способно воспринимать пики и интервалы между ними длительностью менее 100 мкс. На значение напряжения также имеется некоторый допуск, в пределах которого счет осуществляется.

Показанный на рис. 2.10 счетчик имеет два регистра индикации. Верхний регистр, представляющий собой электронный дисплей из светодиодов, показывает текущий отсчет. На нижнем регистре механического типа установлено некоторое заданное число, при совпадении с которым текущего значения на верхнем

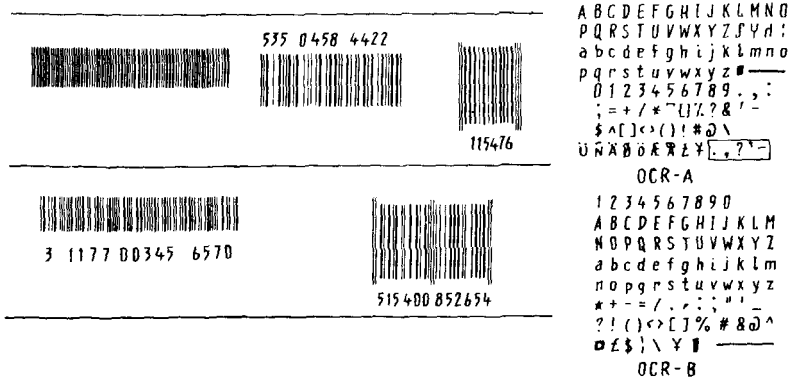


Рис. 2.11

Примеры стандартных полосковых (а) и буквенно-цифровых (б) кодов, предназначенных для оптического распознавания

регистре формируется выходной сигнал. Эта особенность конструкции позволяет счетчику некоторым образом воздействовать на производственный процесс. Хорошим примером может служить, например, автоматическое комплектование изготавливаемых деталей в партии определенного объема. Другим полезным свойством многих счетчиков является возможность счета в двух направлениях, что позволяет им воспринимать два вида входных сигналов — увеличивающих и уменьшающих результат. Это может оказаться полезным при контроле качества выпускаемой продукции и при транспортировании изделий.

Счетчик, показанный на рис. 2.10, является специальным устройством, но в гл. 12 показано, что программируемый контроллер может также выполнять функцию счетчика. Наряду с этим ЭВМ, управляющая производственным процессом, способна выполнять эту функцию при наличии соответствующего программного обеспечения.

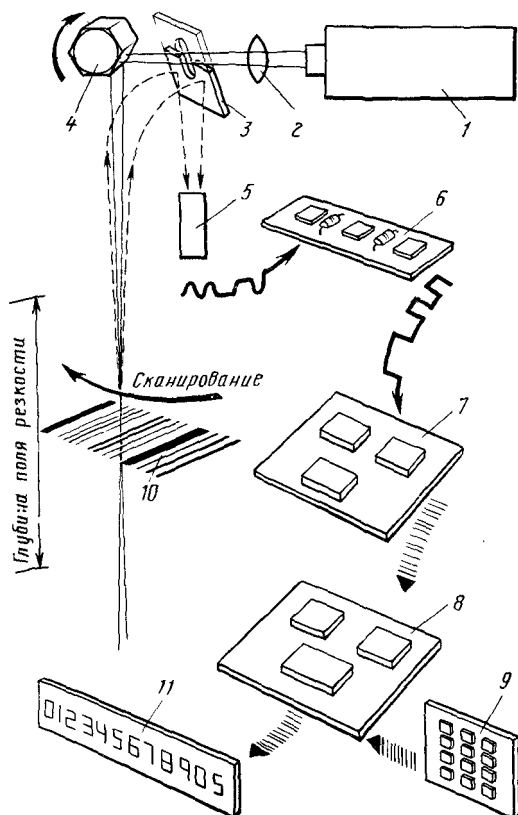
Таймеры. Если на вход подаются точные временные импульсы, счетчик, который их считает, становится таймером, эквивалентом часов. Промышленный таймер больше напоминает будильник, чем обычные часы. Когда прошедшее время подходит к заданному значению, в таймере формируется выходной сигнал — отсюда и аналогия с будильником. Как и счетчики, промышленные таймеры могут работать в двух направлениях, т. е. имеют прямой и обратный счет времени.

Еще одним свойством таймеров является возможность приостанавливать их работу, что позволяет им суммировать периоды времени, когда напряжение имеется, и пропускать периоды, когда напряжение отсутствует. Это будет пояснено в гл. 11, где описано построение циклограмм работы промышленных логических систем управления. Таймеры оказываются весьма полезными в таких системах.

Рис. 2.12

Лазерное считывающее устройство сканирует табличку с полосковым кодом, обрабатывает сигнал и выдает его в цифровой форме, используемой в автоматизированных системах обработки данных:

1 — лазер; 2 — система линз; 3 — полупрозрачное зеркало; 4 — вращающееся зеркало; 5 — фотодетектор; 6 — преобразователь сигнала; 7 — декодер; 8 — управляющая ЭВМ; 9 — клавиатура; 10 — полосковый код; 11 — табло



Использование промышленных таймеров в робототехнике и автоматизации производства даже шире, чем их ближайших родственников — счетчиков. Таймеры могут быть выполнены в виде отдельных приборов или являться составной частью программируемых контроллеров или управляющих ЭВМ.

Системы считывания полосковых кодов.

Системы считывания полосковых кодов могут рассматриваться как датчики, но правильное считать их анализаторами, состоящими из обычных фотоэлектрических или лазерных сканирующих устройств, соединенных с таймерами и счетчиками. Последовательность полосок различной ширины, подобная показанной на рис. 2.11, сканируется и подсчитывается в этих системах. Сканирование производится в направлении, перпендикулярном полоскам, после чего импульсы, принятые фотодатчиком, сравниваются для определения ширины каждой из полосок. Затем сочетания полученных сигналов анализируются, чтобы расшифровать код и перевести его в буквенно-цифровой вид, воспринимаемый автоматизированной системой (рис. 2.12).

Сканирование полосковых кодов обычно производится с помощью лазеров, потому что концентрированный когерентный луч лазера имеет большую рабочую зону по глубине поля резкости (см. рис. 2.9). При этом нет необходимости в точной фиксации таблички с кодом на определенном расстоянии от сканирующего устройства. Глубина поля резкости лазерной системы позволяет обеспечить сканирование коробок или предметов различного

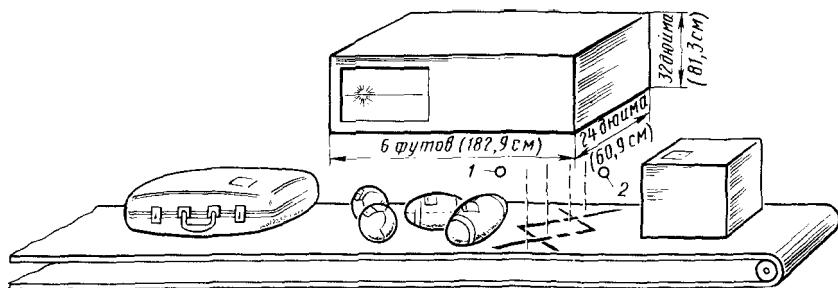


Рис. 2.13

Лазерное сканирующее устройство имеет большую глубину поля резкости и производит перекрестное сканирование, обеспечивая считывание табличек с полосковыми кодами, имеющих произвольную ориентацию и находящихся от него на различных расстояниях:

1 — входной фотозлемент; 2 — выходной фотозлемент

размера, проходящих на различном от нее расстоянии. Это свойство лазеров иллюстрируется на рис. 2.13. Обратите внимание, что сканируемые предметы имеют случайную ориентацию и не обязательно имеют форму параллелепипедов, особенно футбольные и бейсбольные мячи.

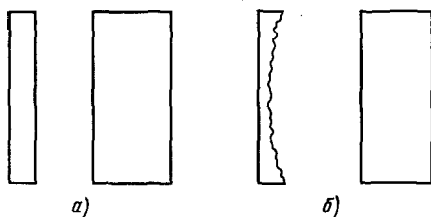
Зигзагообразный путь сканирующего луча позволяет лазерному устройству отыскать ярлык с полосковым кодом при любом его положении.

Другие коды, отличные от полосковых, может быть, легче воспринимаются человеком, но представляют проблему для автоматизированных систем. На рис. 2.11 представлены два буквенно-цифровых кода, предназначенных для оптического распознавания в автоматизированных системах. Другими альтернативами могут быть код «Марк — Сенс», широко используемый для подсчета экзаменационных баллов и опросных листов, или печатание магнитными чернилами, применяемое почти исключительно в банковских операциях для надпечатки чеков. К сожалению, автоматическое распознавание этих знаков является значительно более сложной операцией, чем в случае полосковых кодов. При этом необходимо обеспечить точное положение знаков, а некоторые знаки незначительно отличаются друг от друга. Подумайте хотя бы о проблеме разработки надежного метода сканирования для различения букв *E* и *F*, *O* и *Q*, *P* и *R*. Очевидно, что необходимо многократное сканирование, а нечеткая печать или случайное пятно на ярлыке может привести к неправильному считыванию. Программы сканирования содержат проверки на достоверность, отрицательные результаты которых приводят к выдаче команды на повторное сканирование, чтобы удостовериться в правильности считывания. Программа может включать выдачу сообщения «СБОЙ СЧИТЫВАНИЯ», если попытка повторного считывания была безуспешной. Некоторые дефекты кода могут привести сканирующее устройство к неправильному считыванию, что обычно хуже, чем сбой.

Рис. 2.14

Сочетание элементов полоскового кода:

а — узкая полоса и широкий интервал; б — поврежденная кромка узкой полосы



Вероятность неправильного считывания сканирующим устройством для полосковых кодов может быть сведена к минимуму соответствующей конфигурацией кода. Обычный код для одной цифры или буквы состоит из пяти полосок с четырьмя интервалами между ними. Имеется два варианта ширины полосок и интервалов, что дает количество возможных сочетаний, равное девятизначному двоичному числу. Поскольку $2^9 = 512$, а количество используемых букв, цифр и знаков менее 50, подавляющее большинство из 512 комбинаций полосок будут признаны неправильными, что вызовет повторное сканирование.

Некоторые погрешности в изображении кода вообще не вызывают ошибок считывания, что еще более повышает надежность полосковых кодов. Рассмотрим сочетание узкой полосы с широким интервалом (рис. 2.14, а). Если кромка полосы повреждена (рис. 2.14, б), это не приведет к сбою, потому что повреждение только делает полосу уже, а интервал шире. Если повреждение кромки полосы настолько велико, что полоска полностью разрушена (считывается как интервал) или две полоски соприкасаются (интервал считывается как полоска), считывание будет признано неправильным из-за суммарной ошибки.

Если сканирующее устройство дает большой процент повторных считываний (обнаружений неправильных кодов), вероятность неправильного считывания кодовой комбинации возрастает, потому что это означает возрастание частоты отказов системы из-за состояния приборной части, ярлыка с кодом или ошибок в методе считывания. Однако, даже если количество отказов считывания с первого раза достигает 10 %, вероятность неправильного считывания незначительна: менее одного на несколько сотен тысяч.

Оптические датчики положения. Концентричное расположение светлых и темных полос на диске вместе с возможностью быстрого их считывания открывает новые возможности для автоматизации. На рис. 2.15 изображен диск, который можно жестко закрепить на валу и снабдить оптическими датчиками для каждого кольца. Это устройство называется оптическим датчиком положения (рис. 2.16) и применяется для определения углового положения вала. Информация о положении вала может быть передана в ЭВМ или другое управляющее устройство для контроля частоты вращения или положения вала. Подобные устройства применяются в конструкциях роботов и станков с ЧПУ.



Рис. 2.15
Оптический кодирующий диск



Рис. 2.16
Оптический датчик положения для контроля угловых перемещений вала и других устройств автоматизации, работающий в абсолютной системе координат

Оптические датчики могут быть двух типов: работающие в абсолютной или в относительной системе координат (в приращениях). Последние подают серии импульсов напряжений, пропорциональных углу поворота вала. Управляющая ЭВМ должна знать предыдущее положение вала, чтобы рассчитать новое. Датчики, работающие в абсолютной системе координат, передают набор напряжений, соответствующий положению вала в каждый момент времени. На кольце, расположенном ближе всего к центру, темные и светлые полосы чередуются через 180° , на следующем — через каждые 90° , затем — через 45° и т. д., в зависимости от количества колец на диске. Результирующее битовое сочетание, вызываемое датчиком, соответствует точному угловому положению вала, как это будет показано в примере 2.1.

Пример. 2.1 Оптический датчик положения, работающий в абсолютной системе координат. Оптический датчик положения имеет восемь колец и восемь фотодиодных датчиков. На выход он подает восьмибитовый сигнал. Предположим, что на выходе имеется сочетание 10010110. Каково угловое положение датчика?

Формула для вычисления угла поворота

$$A = \sum_{i=1}^n m_i A_i,$$

где i — число колец; $m_i = 0$, если кольцо светлое (прозрачное); $m_i = 1$, если кольцо темное (непрозрачное); A_i — угловая дискрета для кольца; n — суммарное число колец.

Имеются оптические датчики с 12 кольцами. Их разрешающая способность равна 4000 импульсам на оборот.

Решение.

№ кольца датчика	Дискрета, градус	Сочетание на выходе	Угол поворота, градус
1 (внутреннее)	180	1	180
2	90	0	—
3	45	0	—
4	22,5	1	22,5
5	11,25	0	—
6	5,625	1	5,625
7	2,8125	1	2,8115
8 (внешнее)	1,40625	0	—
		Итого:	210,94

2.3. ИСПОЛНИТЕЛЬНЫЕ УСТРОЙСТВА

После того как информация о состоянии процесса воспринята датчиком и проанализирована, приходит время выполнять соответствующие действия. Как раз на этом и заканчивается автоматизация многих систем, потому что принято, чтобы вмешательство и принятие решения о выполнении действий оставалось за человеком-оператором. Такие системы можно назвать «информирующими», если они просто воспринимают и визуально представляют данные или регистрируют их. Другие системы этого типа можно назвать «ассистирующими», если они, кроме того, анализируют данные и дают рекомендации по последующим действиям или способствуют оператору в быстром принятии решений. Однако все больше автоматизированных систем обходятся без вмешательства оператора, выполняя воздействие на процесс автоматически.

Характер воздействия может быть чисто физическим, как у штанги, по команде ЭВМ или другого анализатора сталкивающей детали с ленты конвейера. В другом случае исполнительный механизм физически замыкает электрическую цепь, что в свою очередь воздействует на процесс. Здесь примером может служить воздействие электрического реле — исполнительного механизма, который выключает питание нагревательного элемента электропечи.

Пневматические и гидравлические цилиндры. Когда требуется автоматически осуществить прямолинейное движение, обычно применяют цилиндр. Наиболее популярны пневмоцилиндры, поскольку магистрали сжатого воздуха широко распространены на производственных предприятиях. Давление сжатого воздуха обычно варьируется в диапазоне от 80 до 100 фунт/кв. дюйм (от 552 до 690 кН/см²), что достаточно для большинства зажимных, пере-мещающих, позиционирующих и силовых устройств. На рис. 2.17 иллюстрируется использование пневмоцилиндров для автомати-

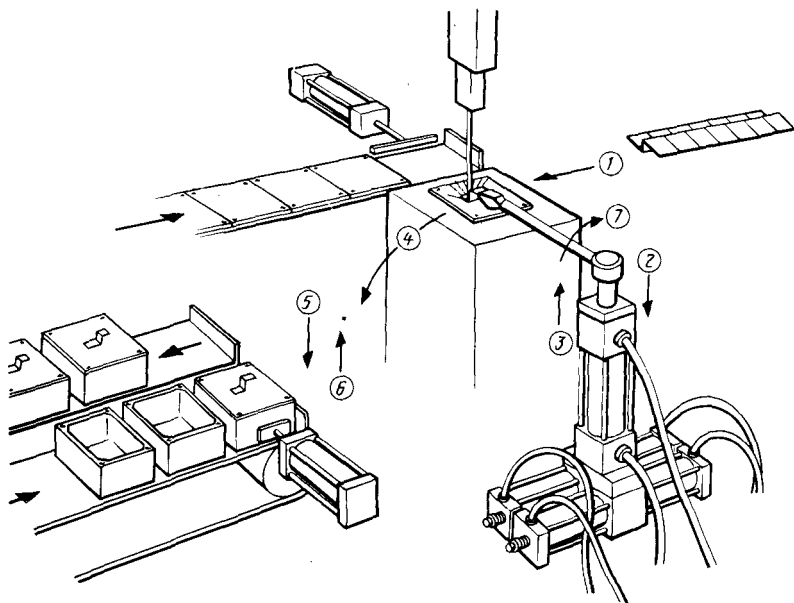


Рис. 2.17

Использование пневмоцилиндров для приведения в движение робота, обслуживающего промышленную установку

зации промышленных установок. Управление пневмоцилиндрами осуществляется клапанами, приводимыми в действие электрическими импульсами или пневматическими логическими устройствами.

Когда производственный процесс требует приложения сил свыше 200 фунтов (890 Н), предпочтение отдается гидроцилиндрам. В гидросистемах может свободно применяться давление свыше 2000 фунт/кв. дюйм (13 790 кН/м²), что значительно выше давления, обычно используемого в пневмосистемах (от 80 до 100 фунт/кв. дюйм). Пневмоцилиндры большого диаметра могут развивать такие же усилия, как и гидроцилиндры, но, учитывая габариты и другие преимущества, для получения больших усилий предпочтение отдается гидроцилиндрам. Наиболее мощные промышленные роботы приводятся гидравлическими исполнительными механизмами.

К преимуществам гидроцилиндров, кроме большой мощности, можно отнести удобство управления ходом. Кроме того, они не шумят, хотя этого нельзя сказать о насосах и сосудах. Из недостатков следует упомянуть высокую стоимость, сложность обслуживания и проблему утечек из цилиндров.

К вопросу о выборе пневматического или гидравлического исполнительного механизма следует подходить с осторожностью, поскольку необходимо учитывать одновременно требования к дав-

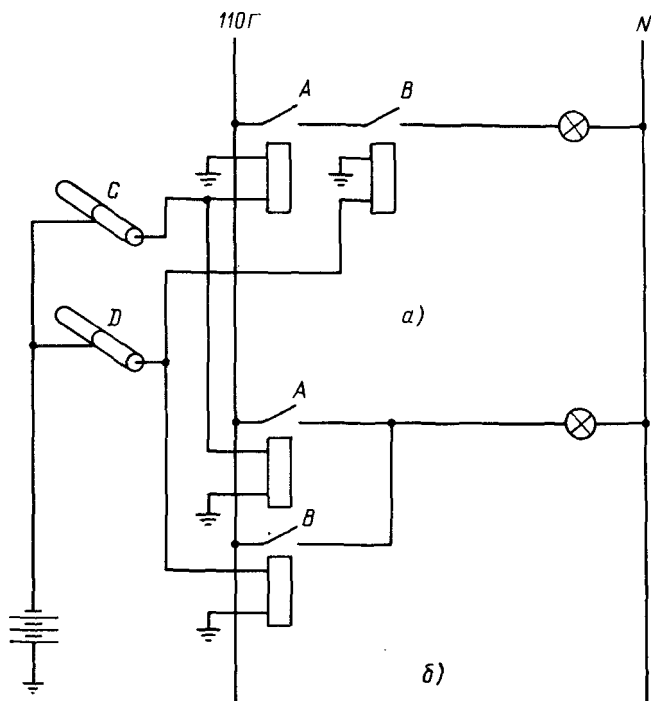


Рис. 2.18

Использование реле *A, B* для автоматического управления электрическими цепями. В схеме *a* для замыкания силовой цепи необходимо, чтобы она датчика *C* и *D* подали управляющее напряжение. В схеме *б* для этого достаточно срабатывания одного из датчиков

лению и расходу. Система, способная обеспечивать давление, достаточное для приведения в действие цилиндров или других исполнительных механизмов, может оказаться не в состоянии поддерживать это давление во время быстрых перемещений. Это явление можно наблюдать на частично автоматизированных заводах, где пневмосистемы используются для приведения в действие механизированных винтовертов, скобочных машин и транспортного оборудования. Конструкция системы, которая рассчитана на максимальный расход, потребляемый гидравлическим или пневматическим оборудованием, позволит избежать этих неприятностей.

Электромагниты. Когда требуется осуществить небольшое ненагруженное быстрое линейное перемещение, логичнее всего остановить свой выбор на электромагнитах. Из курса физики мы знаем, что принцип действия электромагнита основан на использовании свойств магнитного поля, возникающего при прохождении электрического тока по катушке из проволоки. При этом сердечник втягивается в катушку.

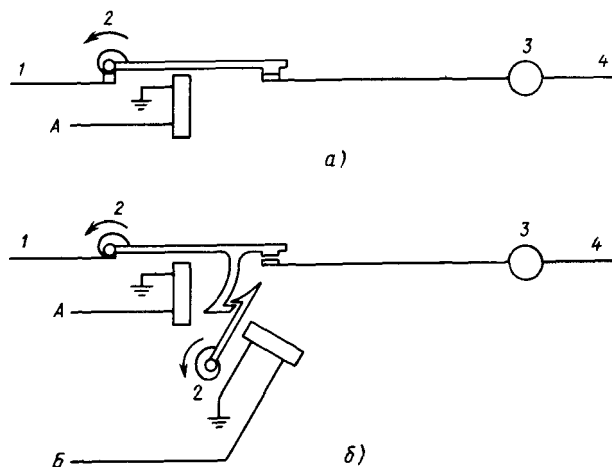


Рис. 2.19

Сравнение обычных реле и реле, снабженных блокировкой:

а — реле без блокировки поддерживает цепь в замкнутом состоянии, если по обмотке проходит ток, подаваемый через управляющий вход *А*; *б* — реле с блокировкой после подачи напряжения на вход *А* удерживается во включенном состоянии механической защелкой; импульс, подаваемый на вход *Б*, освобождает реле от защелки; *1* — провод; *2* — пружина; *3* — нагрузка; *4* — нейтраль

При отсутствии тока в катушке сердечник может автоматически возвращаться в исходное положение под действием пружины. В отличие от движения штока гидроцилиндра перемещение сердечника электромагнита плохо поддается контролю, однако при автоматизации часто надо обеспечить только короткое быстрое дискретное перемещение, а не медленный контролируемый ход.

Реле. Из всех электромагнитов наиболее известны те, что включают (выключают) электрические цепи, т. е. обычные реле. Цепи управления обычно работают при пониженном напряжении и значительно меньшей силе тока, чем силовые. На рис. 2.18 показан пример использования реле для замыкания электрических цепей по команде датчиков. Сравним логику, заложенную в две показанные на рисунке цепи.

В схеме *а* на рис. 2.18 для того, чтобы силовая цепь была замкнута, должны сработать реле, подключенные к обоим датчикам. В схеме *б* для этого достаточно, чтобы сработало реле *А* или реле *В*. Громадное количество способов, которыми можно скомбинировать реле в управляющих цепях, образует основу классического подхода к автоматизации производственных систем. В гл. 10 и 11 показано, каким образом логические цепи позволяют решать задачи автоматического управления. Такие цепи могут включать в себя наборы реле, интегральные схемы или программируемые контроллеры для выполнения идентичных задач. Понимание сущности этих систем зависит от понимания принципа работы основного элемента — реле.

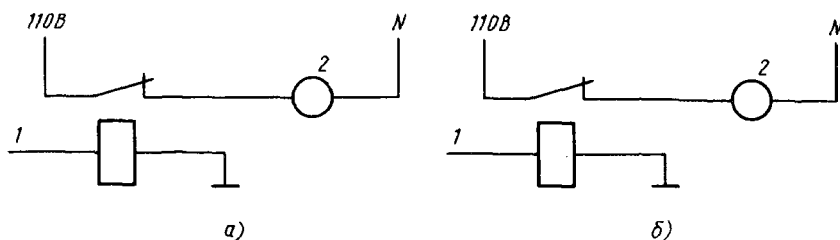


Рис. 2.20

Реле с замыкающими (а) и размыкающими (б) контактами:

1 — управляющий вход; 2 — нагрузка

Как это показано на рис. 2.19, реле бывают с блокировкой или без блокировки. Реле с блокировкой требуют только одного электрического импульса для того, чтобы замкнуть цепь и поддерживать ее в таком состоянии. Для отключения реле требуется подача импульса в другую цепь управления. Реле без блокировки удерживаются во включенном состоянии только при прохождении по катушке тока, и поэтому для их работы требуется непрерывный электрический сигнал. Прерывание сигнала приводит к немедленному отключению реле и разрыву цепи.

Выше рассмотрены реле, которые при срабатывании замыкают цепи, но они могут также и размыкать цепи при получении электрического сигнала. Когда срабатывание реле замыкает цепь, его контакты называют замыкающими (контакты реле на рис. 2.18 — замыкающие). В отличие от этого контакты реле, которые при срабатывании размыкаются, называют размыкающими. Из этого следует, что нормальным состоянием реле является выключенное состояние. На рис. 2.20 сравниваются реле с замыкающими и размыкающими контактами, включенные в неразветвленную силовую цепь. В цепи на рис. 2.20, а для замыкания цепи датчик должен подать управляющий импульс. В цепи на рис. 2.20, б для включения цепи управляющее напряжение должно быть отключено.

Обычные реле и электромагниты чаще всего питаются постоянным током низкого напряжения. Однако удобство и распространенность переменного тока напряжением 110 В привели к созданию реле и электромагнитов, работающих на переменном токе. Еще одним преимуществом таких электромагнитов является их сравнительная нечувствительность к электрическим наводкам, создаваемым силовым промышленным оборудованием. Чувствительные электромагниты могут быть непредумышленно включены сторонним напряжением, наведенным большими токами, проходящими по электрическому кабелю, проложенному вблизи цепи реле, особенно если оба проводника находятся рядом, лежат параллельно и имеют значительную протяженность.

По мере роста напряжения силовых цепей используются разные реле. Небольшие электродвигатели, загрузочно-разгру-

зочные устройства и приводы автоматизированных машин обычно питаются от силовых цепей, рассчитанных не более чем на 10 А, и реле, которые их обслуживают, действительно называются «реле». Однако, если сила проходящего тока от 10 до 30 А, используется термин «силовое реле». При больших токах реле можно называть контактором. Принципы, на которых основывается работа обычного реле, во всех случаях сохраняются, и инженер по автоматизации не должен быть смущен этой терминологией.

Особенно нужны реле для коммутации силовых цепей электродвигателей. Инженер по автоматизации услышит о пускателях электродвигателей — контакторах или реле, которые дополнительно осуществляют защиту от перегрузки, размыкая цепь питания двигателя, если механические нагрузки на двигатель слишком велики и вызывают прохождение слишком большого тока. Во время пуска двигателя, как известно, возникают перегрузки, которые пускатели должны в определенных пределах допускать. По этой причине обычно используют тепловые устройства защиты от перегрузок, что позволяет им быть до некоторой степени терпимыми к кратковременным перегрузкам.

2.4. ПРИВОДЫ

Приводы, как и исполнительные механизмы, воздействуют на процессы в соответствии с командами ЭВМ или другой управляющей машины. Для классификации мы здесь вводим разграничение между приводами и исполнительными механизмами, считая, что последние используются для выполнения законченных дискретных короткоходовых движений, обычно линейных. Приводы реализуют более продолжительные движения, по большей части вращательные. Исполнительные механизмы могут включать и выключать приводы, а приводы могут служить источником механической энергии для исполнительных механизмов. Некоторые устройства автоматизации, например мальтийские механизмы и качающиеся штанги, принадлежат, по-видимому, к обоим категориям.

Двигатели. Наиболее важной составной частью привода является двигатель. Инженер по автоматизации должен широко толковать термин «двигатель», имея в виду не только электродвигатели, но и гидро- и пневмомоторы. В то же время двигатели внутреннего сгорания сравнительно редко применяются в автоматизированном производстве. Гидро- и пневмомоторы являются машинами, обратными соответствующим насосам. Когда гидравлическое или пневматическое давление без труда может быть получено в производственной системе или автоматизированной установке, целесообразно использовать его для сообщения движения подсистемам, применяя гидро- или пневмомоторы. Гидромоторы заданного габарита развивают большую мощность, чем пневмомоторы. Менее мощные и более шумные пневмомоторы,

тем не менее, часто практичнее при автоматизации, поскольку магистрали сжатого воздуха часто имеются на предприятиях. Гидро- и пневмомоторы имеют преимущества по сравнению с электродвигателями с точки зрения электро- и взрывобезопасности.

Используя электродвигатели, инженер по автоматизации должен знать, какой тип двигателя ему нужен. Большинство серводвигателей являются двигателями переменного тока. Имеются два специальных типа электродвигателей постоянного тока, применяемых в приводах роботов и устройствах автоматизации производства, на особенностях которых следует остановиться.

Шаговые двигатели. Шаговые двигатели весьма удобны для автоматизации. Они управляются дискретно подаваемыми импульсами напряжения постоянного тока, являющимися обычным выходом цифровых ЭВМ и других систем управления. Шаговый двигатель идеален для осуществления точных угловых перемещений, что требуется для индексации и других целей автоматизации.

Шаговые двигатели прекрасно зарекомендовали себя в устройствах без обратной связи, где система управления только выдает команду, не проверяя ее отработки. Шаговые двигатели применяются в приводах некоторых промышленных роботов и станков с ЧПУ. В большинстве случаев эти приводы не имеют обратной связи, однако она может быть осуществлена путем контроля положения приводимого узла. Устройство в цепи обратной связи сравнивает действительное положение узла с заданным, определяя рассогласование. Устройство управления подает импульсы на шаговый двигатель до тех пор, пока рассогласование не будет сведено к нулю.

Серводвигатели постоянного тока. Серводвигатели постоянного тока применяются в станках с ЧПУ и промышленных роботах для управления движением. Используя контур обратной связи, система управления может подавать на двигатель напряжение постоянного тока, пропорциональное наблюдаемому рассогласованию. Когда рассогласование сведено к нулю, напряжение также снижается до нуля, и двигатель останавливается. Более развитые сервоприводы могут регулировать напряжение пропорционально скорости изменения рассогласования или результатам суммирования накопленного рассогласования по времени, а иногда по обоим параметрам сразу.

Важной особенностью серводвигателей постоянного тока и шаговых двигателей является способность сохранять вращающий момент в неподвижном состоянии. Поэтому эти двигатели используют не только для вращения вала, но и для удержания его в определенном положении. Станки с программным управлением и промышленные роботы являются оборудованием, для которого требуется сохранение положения исполнительных органов между командами на его изменение.

Кинематические цепи. Занимаясь автоматизацией производственного оборудования, легко упустить из вида, что обильный

и доступный источник механической энергии существует в самой установке. Например, инженер по автоматизации иногда использует отдельный электродвигатель для автоматизации загрузки установки, хотя дешевле и практичнее использовать для этого отводную кинематическую цепь. При этом упускается из виду явное преимущество таких цепей, заключающееся в синхронизации. Когда скорость установки возрастает, отводная кинематическая цепь немедленно на это реагирует. Это особенно удобно при обеспечении сбалансированности сборочных линий. Важно, чтобы погрузочно-разгрузочные и транспортные операции были скоординированы с работой установки, а отводные кинематические цепи обычно являются наилучшим средством достижения такой координации. Зубчатые колеса, кулачки, рычаги и храповые механизмы являются компонентами отводных кинематических цепей. В книге не рассматриваются конструкции этих механических устройств общего назначения. Здесь лишь указывается на их полезность при отборе мощности от производственной установки для транспортирования и позиционирования материала, осуществляемого автоматической системой, поддерживающей производственный процесс. Два механических устройства особенно важны для обеспечения прерывистых движений в автоматизированных поточных линиях. Это мальтийские и качающиеся штанговые механизмы.

Мальтийские механизмы. Мальтийские механизмы служат для осуществления прерывистого поворота делительных столов. На рис. 2.21 изображен такой механизм, обычно устанавливаемый снизу стола. Он состоит из двух дисков: ведущего (кривошипа) и ведомого (креста). Передача движения осуществляется штифтом кривошипа, входящим в радиальные прорези креста, имеющие форму арок. Кривошип вращается постоянно, передавая вращение кресту во время прохождения угла B (рис. 2.22). Всю оставшуюся часть поворота кривошипа стол остается неподвижным. Этот период — период выстоя — используется для выполнения работы на позициях делительного стола.

Следует обратить внимание на то, что при входе и выходе штифта из прорези вектор скорости его движения направлен строго по радиусу (см. рис. 2.21). Это необходимо для обеспечения плавного начала поворота и полной неподвижности стола во время выстоя. Фактически любое смещение стола по инерции во время выполнения работ (фаза выстоя) приведет к тому, что штифт кривошипа не войдет в следующий паз. Это является серьезной неисправностью, приводящей к заеданию привода установки и невыполнению операций на поверхности стола. Время индексации и время выстоя зависит от частоты вращения кривошипа и не меняется от такта к такту.

Тактовые столы с мальтийскими механизмами могут иметь различное количество позиций. Оно не может быть менее трех и редко превышает восемь. На рис. 2.22 приведена схема маль-

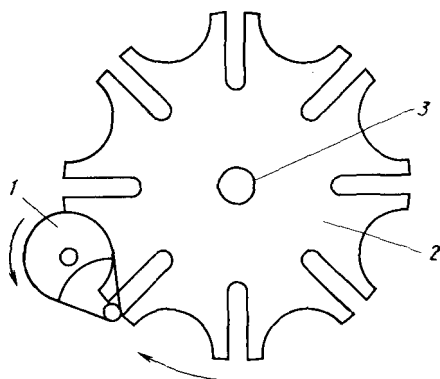


Рис. 2.21

Мальтийский механизм:

1 — ведущий диск; 2 — ведомый диск; 3 — ось стола

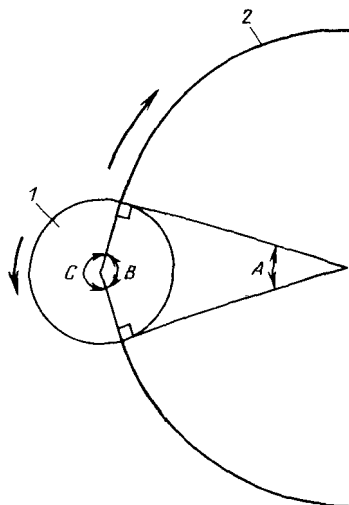


Рис. 2.22

К расчету мальтийского механизма:

1 — ведущий диск; 2 — делительный стол

тийского механизма, поясняющая расчет соотношений между числом позиций, временем выстоя, временем индексации и частотой вращения кривошипа. Из построения следует, что сумма углов A и B равна 180° . Сумма углов B и C равна 360° .

$$T_{\text{и}} = B^\circ \cdot 60 / 360^\circ;$$

$$T_{\text{в}} = C^\circ \cdot 60 / 360^\circ;$$

$$T_{\text{т}} = T_{\text{и}} + T_{\text{в}} = 60/n,$$

где $T_{\text{и}}$ — время индексации, с; $T_{\text{в}}$ — время выстоя, с; $T_{\text{т}}$ — время такта, с; n — частота вращения кривошипа, мин^{-1} ; 60 — переводной коэффициент, с/мин.

Пример 2.2. Расчет делительного стола. Задача: Мальтийский механизм делительного стола имеет шесть позиций. Частота вращения кривошипа 12 мин^{-1} . Определить: а) время индексации; б) время выстоя; в) идеальную производительность (количество тактов в 1 ч).

Решение:

$$A = \frac{360^\circ}{6} = 60^\circ;$$

$$B = 180^\circ - A = 120^\circ;$$

$$C = 360^\circ - B = 240^\circ;$$

$$T_{\text{и}} = \frac{B \cdot 60}{360} = \frac{120^\circ \cdot 60}{360^\circ \cdot 12} = 1,67 \text{ с};$$

$$T_{\text{в}} = \frac{C \cdot 60}{360} = \frac{240^\circ \cdot 60}{360^\circ \cdot 12} = 3,33 \text{ с};$$

$$T_{\text{т}} = T_{\text{и}} + T_{\text{в}} = 1,67 + 3,33 = 5 \text{ с};$$

Идеальная производительность

$$\Pi = \frac{1}{T_{\text{т}}} 3600 = 720 \text{ (тактов в 1 ч)}.$$

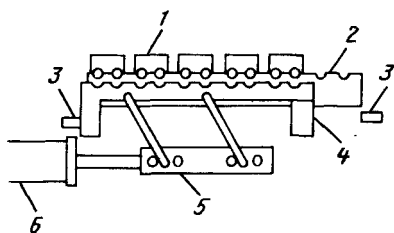
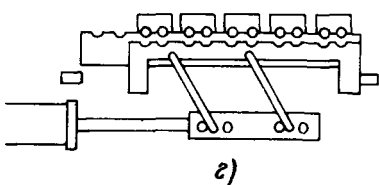
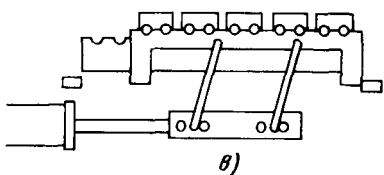
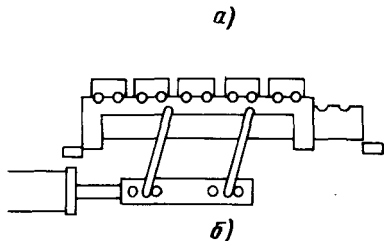


Рис. 2.23
Качающийся штанговый механизм:
 1 — поддоны для деталей; 2 — неподвижная штанга; 3 — упор; 4 — подающая штанга; 5 — ползун; 6 — поршень



Качающиеся штанговые механизмы создают повышенный шум и могут повредить продукцию при резком опускании ее на неподвижную штангу, происходящем в каждом такте. Однако по возможному числу позиций они превосходят делительные столы с мальтийскими механизмами. При большом количестве позиций линейные конвейеры позволяют создать более удобную планировку по сравнению с вращающимися.

Выводы

В главе рассмотрены «элементы» механизации и автоматизации. Автоматизация начинается с простой, а иногда и не очень простой механизации отдельных элементов операции конкретной установки. Хотя системный подход к полной автоматизации и хорошо иметь в виду, в реальности большинство предприятий автоматизируется постепенно, шаг за шагом.

Составные элементы механизации и автоматизации с трудом поддаются классификации, однако можно выделить четыре основ-

ных группы: датчики, анализаторы, приводы и исполнительные механизмы.

Оснащение выключателями и датчиками для решения конкретных задач автоматизации обычно не дает очевидных преимуществ, пока и другие элементы автоматизации не введены в систему. В меньшей степени сказанное относится к исполнительным механизмам и приводам, которые собственно реализуют автоматизацию.

Описание типов автоматизированных систем и их применения в производстве при дальнейшем изложении будет базироваться на элементах автоматизации и механизации, рассмотренных в данной главе.

Упражнения и вопросы

2.1. Что будет представлять собой, по Вашему мнению, конструкция одно-полосного шестипозиционного переключателя?

2.2. Изображенный на рис. 2.24, а полосковый код соответствует —609— Используя эту информацию, расшифруйте полосковый код на рис. 2.24, б:

а) чему соответствует код на рис. 2.24, б? (Напоминаем, что каждый знак изображается пятью полосками);

б) закодируйте цифру 6.

2.3. Фотоэлектрическая система предназначена для определения присутствия или отсутствия болтов диаметром 38 дюйма (9,525 мм) с прямоугольной резьбой и шлицованной потайной головкой на дорожке подающего устройства. При этом болты должны полностью прерывать световой луч диаметром $\frac{1}{2}$ дюйма (12,7 мм). Можно ли такую фотоэлектрическую систему практически реализовать? Возникновение каких проблем Вы предвидите? Каковы пути решения технических проблем, если они возникнут?

2.4. Сколько полюсов имеет выключатель, изображенный на рис. 2.25? На сколько цепей он рассчитан?

2.5. Абсолютный оптический датчик положения имеет на оптическом диске четыре кольца. Битовый сигнал на выходе соответствует 1001. Каково абсолютное угловое положение вала?

2.6. Предположим, что с помощью некоторого идентификационного кода можно реализовать 500 знаков, из которых распознаются всего 50. Значит ли это, что остальные 450 пропали напрасно? Поясните ответ.

2.7. Приблизительно вычислите вероятность неправильного считывания знаков для кода, описанного в упражнении 2.6, при случайной ошибке считывающего устройства.

2.8. Предположим, что Вы создаете автоматическую систему для идентификации палет автоматического склада при помощи идентификационных ярлыков, используя код, описанный в упражнении 2.6. На фирме имеется 500 таких палет,

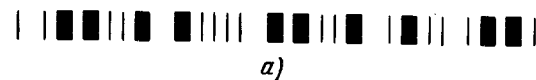


Рис. 2.24

Полосковые коды для упражнения 2.2

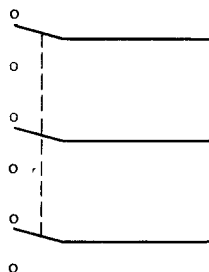


Рис. 2.25

Схема выключателя для упражнения 2.4

и Вам следует определить, по сколько знаков должно быть на ярлыках. Какое минимальное количество знаков необходимо для работы с 500 палетами?

2.9. Предположим, что при работе системы, описанной в упражнении 2.8, произошла случайная ошибка считывающего устройства. Какова вероятность того, что эта ошибка приведет к формированию недействительного кода и повторному сканированию, если на ярлыке имеются два знака? Какова вероятность того, что вместо повторного сканирования произойдет неправильная идентификация палеты?

2.10. Предположим, что обслуживающий персонал жалуется на надежность автоматизированной системы, описанной в упражнении 2.9. Как сократится вероятность неправильной идентификации палет при использовании ярлыков с четырьмя знаками вместо двух?

2.11. Обратите внимание, что верхняя часть букв кода OCR-A, изображенного на рис. 2.11, несколько уменьшена и буквы кажутся вытянутыми вверх. Можете ли Вы объяснить эту кажущуюся неоднородность в формате OCR-A?

2.12. Некий полосковый код состоит из пяти полосок и четырех интервалов между ними. В каждое сочетание входят две широкие и три узкие полоски при одном широком и трех узких интервалах. Сколько существует при этом неповторяющихся комбинаций? (Обратите внимание, что, поскольку система должна регистрировать при каждом считывании именно две широкие полоски, три узкие полоски, один широкий интервал и три узких интервала, считать неправильную комбинацию практически невозможно. Вместо этого будет выдано сообщение «Сбой считывания»).

3. МЕХАНИЗАЦИЯ МАНИПУЛИРОВАНИЯ ДЕТАЛЯМИ

Несомненно, читатель хотел бы сразу перейти к изучению роботов, этих очаровательных детей автоматизации. Однако до того как начать глубоко разбираться в аспектах механизации манипулирования изделиями и автоматизации сборки, важно уяснить, что промышленный робот может оказаться неспособным захватывать изделия в нужном положении или может только подавать их на станцию ручной сборки. В то время как последнее вполне приемлемо, неспособность захватить изделие может привести к полному краху в применении робота, что могут подтвердить многие инженеры по автоматизации в промышленности, которые поспешили обзавестись роботом, не разобравшись с вопросами, раскрываемыми в этой главе. Она посвящена манипулированию изделиями и их позиционированию для обеспечения сборочных и обрабатывающих установок.

3.1. ПОДАЧА ИЗДЕЛИЙ

Изготовленные детали, будь это поковки, отливки, детали, подвергнутые механической обработке, пластмассовые или резиновые молдинги, деревянные изделия или электрические компоненты, должны транспортироваться, отбираться, а также соответствующим образом ориентироваться и позиционироваться для сборки или последующих операций. Счастлив тот инженер по автоматизации, чье производство получает эти детали ориентированными, позиционированными и готовыми к сборке. Однако обычно поставка деталей по кооперации от субпоставщика или из другого цеха завода производится в большой таре навалом. Этот факт приводит к появлению первого условия автоматизации сборочных установок или любых других установок, обрабатывающих полученные заготовки: согласованности условий поставки.

Согласованность условий поставки. Перед тем как начать автоматизацию сборочной станции, инженер по автоматизации должен выяснить, откуда поступают на сборку компоненты. Он должен задаться вопросом: «Действительно ли необходимо, чтобы компоненты поступали на сборочную станцию вперемешку и навалом в таре?» Даже если компоненты закупаются где-то на стороне,

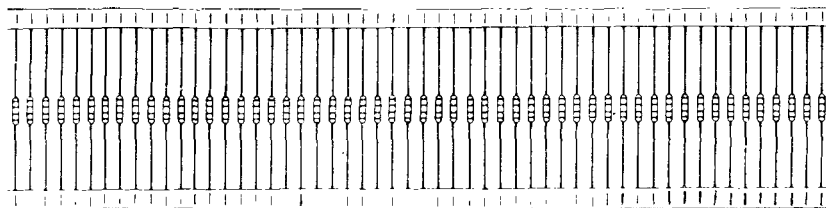


Рис. 3.1

Лента с резисторами для непрерывной подачи в сборочный автомат

иногда возможно договориться, чтобы они отгружались в магазинах, закрепленными на длинных лентах или, по крайней мере, в контейнерах с ячейками. Например, в промышленности распространена поставка резисторов и гвоздей, закрепленных на длинных лентах (рис. 3.1), что определенно способствует автоматизации процессов, в которых они применяются.

При разработке технических условий на закупаемые изделия и выборе их поставщика необходимо принимать во внимание упаковку и ориентацию изделий. Затраты, которых потребовали бы транспортирование материалов и организация собственного производства, могут оправдать часто более высокую стоимость покупных компонентов в магазинах, лентах или в рассортированном виде. Иногда стоимость сортировки удивительно невелика, а порою рассортированные детали могут оказаться даже дешевле. Рассмотрим изготовление пластмассовых молдингов, изображенных на рис. 3.2. Поставщику, может быть, пришлось бы вручную отделять изделия от литьевого каркаса, что увеличивает его стоимость. Обратим однако внимание на симметричное расположение молдингов на каркасе. При закупке молдингов вместе с каркасом их можно отделять от каркаса у себя вручную или роботом, сохраняя правильную ориентацию. Из этого следует, что инженер по

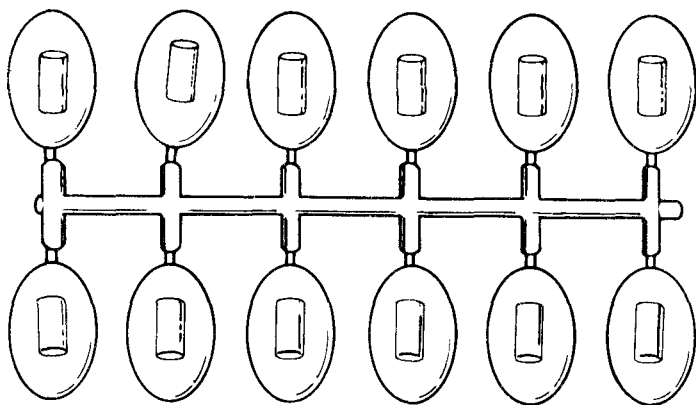


Рис. 3.2

Пластмассовые молдинги до их отделения от литьевого каркаса

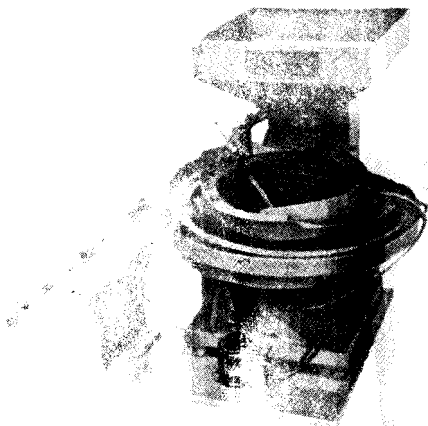


Рис. 3.3
Вибробункер

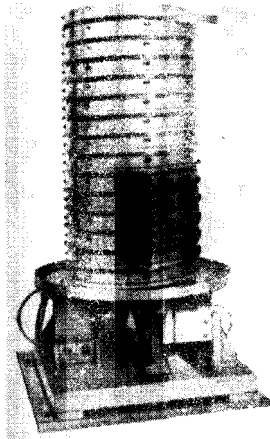


Рис. 3.4
Спиральный виброэлеватор

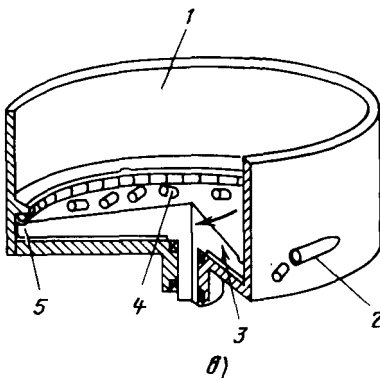
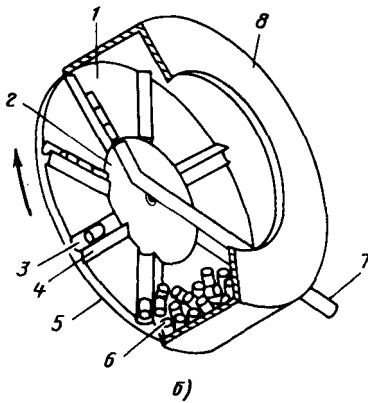
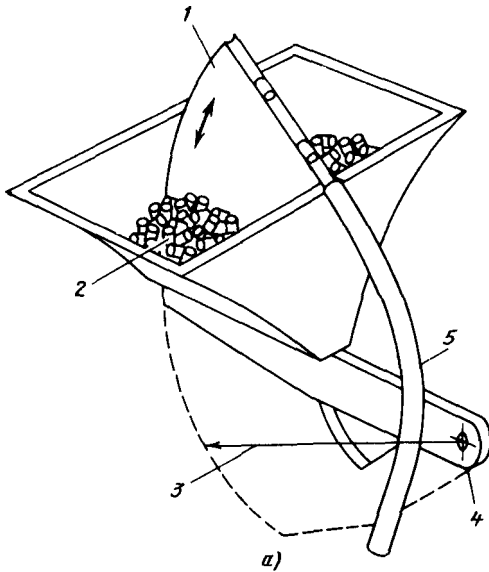
автоматизации существенным образом связан с закупкой комплектующих изделий, и это станет еще более очевидно из следующих разделов главы.

Перемещение и передача. Если самый плохой вариант оказался реальностью и детали получены в таре наваленными вперемешку, для обеспечения автоматической подачи деталей надо сначала заставить их перемещаться. В этом деле преуспели инженеры-механики, причем некоторые выдающиеся результаты в этой области опубликованы Д. Бутройдом с коллегами из Массачусетского университета [9, 11, 57] и Х. Е. ден Хамером из Нидерландов [17].

Деталям может быть придано движение за счет сил гравитации, центробежных сил при помощи перемешивания, сжатого воздуха или вибрации. Приводиться эти устройства могут от отдельных электродвигателей или от обслуживаемой установки. Второй вариант имеет преимущество по обеспечению синхронизации, как это показано в гл. 2. Самым универсальным подающим устройством для деталей малого размера является вибробункер, изображенный на рис. 3.3. Наиболее удивительной чертой вибробункеров является их способность заставлять вибрирующие детали подниматься вверх по наклонным дорожкам или направляющим, проходящим по спирали внутри бункеров, за счет разницы в ускорении во время цикла колебаний, при этом детали забрасываются вверх раньше, чем они успевают соскльзывать, пока новый цикл не забросит их еще дальше. Колебания происходят в направлении, наклонном к поверхности направляющей, по которой происходит перемещение деталей. Направление колебаний можно разложить на две составляющие: параллельную направляющей и перпендикулярную к ней (т. е. вертикальную).

Рис. 3.5

Способы подачи цилиндрических деталей:



а — бункер с качающейся подающей лопастью: 1 — лопасть в крайнем верхнем положении; 2 — детали; 3 — радиус качания; 4 — ось; 5 — желоб; 6 — толкатель с вращающимся дном: 1 — детали, соскальзывающие в спускной желоб; 2 — неподвижный диск; 3 — паз; 4 — выступ; 5 — вращающийся диск; 6 — детали; 7 — спускной желоб; 8 — неподвижный бункер; а — центробежный бункер: 1 — бункер; 2 — спускной желоб; 3 — вращающийся диск; 4 — детали; б — паз

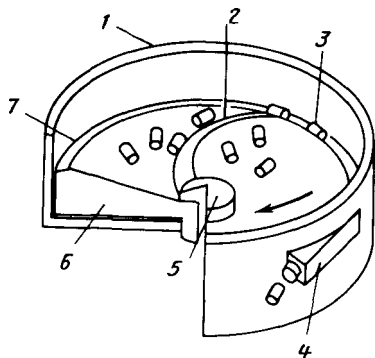
Частота вибрации может меняться, причем увеличение частоты в некотором диапазоне приводит к росту скорости подачи, а затем к ее снижению. Инженер по автоматизации обычно приобретает бункеры промышленного изготовления, не пытаясь разрабатывать собственные конструкции и изготавливать их у себя на предприятии. Одной из принципиальных особенностей вибробункеров является рециркуляция, т. е. возврат деталей к началу дорожки после падения с нее или возврат под воздействием ориентирующего механизма, расположенного вдоль дорожки или наверху.

Близким родственником вибробункера является спиральный виброэлеватор, изображенный на рис. 3.4. Такие устройства имеют некоторые преимущества, когда необходимо только подавать детали, не ориентируя их и не отсеивая детали с неправильной ориентацией.

Рис. 3.6

Бункер с вращающимся основанием, предназначенный для осторожной подачи чувствительных к ударам цилиндрических деталей:

1 — неподвижный бункер; 2 — неподвижный отсекаТЕЛЬ; 3 — деталь; 4 — спускной желоб; 5 — ступица; 6 — вращающееся основание; 7 — узкая кромка



Установка, показанная на рис. 3.4, используется в нагревательной камере для доведения электронных изделий до требуемой температуры во время их перемещения вверх по спиральной дорожке длиной 56 футов (17 м), перед тем как подвергнуть их испытаниям. Три способа подачи цилиндрических деталей показаны на рис. 3.5. В бункере с качающейся подающей лопастью используется возвратно-поступательное движение. В двух других устройствах используется вращение, но с разными целями. В центробежном бункере вращение служит для создания центробежной силы, в то время как в питателе с вращающимся диском детали просто перемешиваются до тех пор, пока не попадут под действием силы тяжести в ориентирующие щели. Для более осторожной подачи чувствительных к ударам цилиндрических деталей можно использовать бункер с вращающимся основанием (рис. 3.6). Еще два вида питающих устройств для цилиндрических деталей, бункер с крыльчатым колесом и бункер галтовочного типа показаны на рис. 3.7. Первое устройство внешне напоминает бункер с качающейся подающей лопастью, но принципиально эти устройства совершенно различные. Крыльчатое колесо не захватывает детали, а просто их перемешивает и отталкивает неориентированные детали с пути деталей, попавших в щель, обеспечивая им проход. Радиус лопастного колеса таков, что оно не касается деталей, которые правильно попали в паз. Бункер с лопастным колесом в сущности является устройством гравитационного типа.

Бункер галтовочного типа, напоминающий барабан для сушки белья, оснащен лопастями для захвата деталей во время вращения для того, чтобы они падали на направляющую в середине. Слишком большая частота вращения делает это устройство неработоспособным.

Бункер с вращающимися подающими лопастями (рис. 3.8) похож на бункер с крыльчатым колесом, но из рисунка видно, что лопасть фактически подхватывает детали, не сталкивая их обратно. П-образная деталь съезжает вниз и перемещается по медленно поворачивающейся лопасти до той точки, где она переходит на неподвижную направляющую.

Для деталей типа дисков можно использовать бункер с вращающимся крюком (рис. 3.9). Он похож на бункер с вращающимся

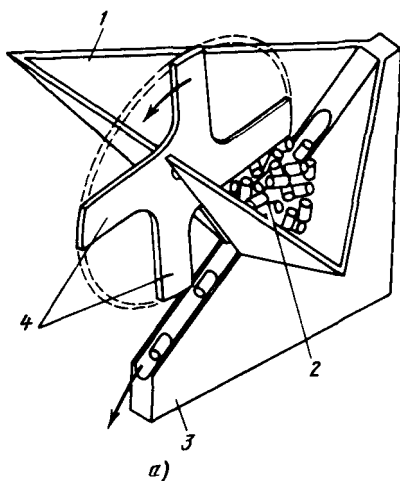
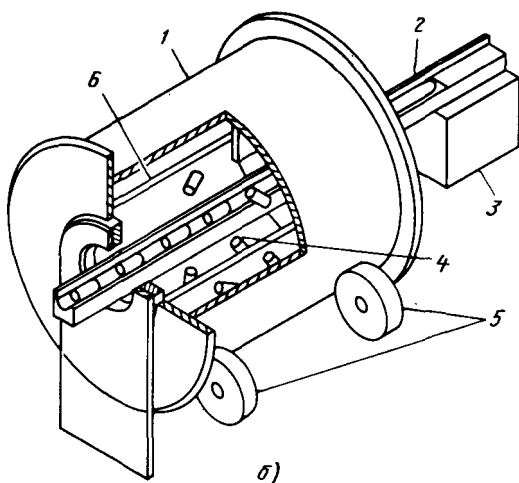


Рис. 3.7

Бункер с крыльчатим колесом и бункер галтовочного типа для подачи прочных цилиндрических деталей:

а — бункер с крыльчатим колесом: 1 — бункер; 2 — детали; 3 — скат; 4 — крыльчатка; *б* — бункер галтовочного типа: 1 — барабан; 2 — вибрирующая штанга; 3 — вибратор; 4 — детали; 5 — приводные ролики; 6 — лопасти



основанием, но его основание неподвижно. Работают эти устройства совершенно по-разному, поскольку вращающийся крюк, имеющий меньшую толщину, чем диски, скользит по поверхности основания, загоняя диски в паз, который отводит их к отверстию в центре. Если диски изготовлены из стали, их можно подавать, используя магнитное питающее устройство. На рис. 3.10 изображены два типа магнитных питающих устройств.

Методы ориентации. Следует отметить, что многие передающие механизмы, обеспечивающие перемещение деталей, одновременно ориентируют их перед тем, как поместить на направляющую или отправить в лоток. Однако приведенные ранее рисунки относятся к деталям простой формы, таким, как цилиндры и диски. Цилиндры могут быть слегка усложнены, иметь конические поверхности, но, скажем, винты с головками или другие детали,

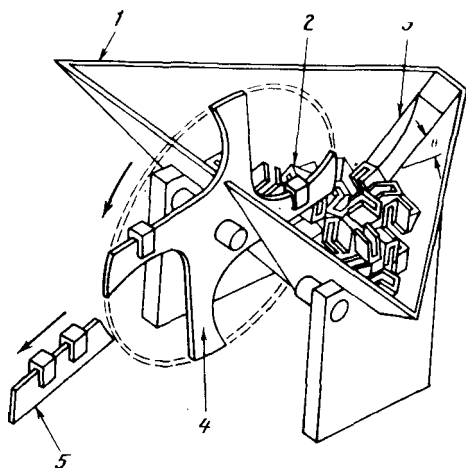


Рис. 3.8

Бункер с вращающимися подающими лопастями:

1 — бункер; 2 — детали; 3 — паз; 4 — лопастное колесо; 5 — выходной склнз

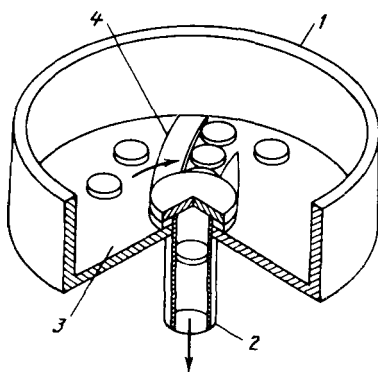


Рис. 3.9

Бункер с вращающимся крюком для дискообразных деталей:

1 — бункер; 2 — спускной желоб; 3 — неподвижное основание; 4 — вращающийся крюк

имеющие более сложную форму, требуют более сложных средств ориентации.

Если более внимательно рассмотреть самый универсальный подающий механизм, каким является вибробункер, обнаружится, что наклонные направляющие наверху бункера снабжены несколькими небольшими выступами, предназначенными для распознавания геометрии конкретной подаваемой детали. Эти выступы толкают, сбрасывают или, наоборот, высвобождают неправильно ориентированные детали для того, чтобы они упали обратно в бункер и вернулись в правильном положении. Ориентация винтов с головками или заклепок может служить хорошим примером. На рис. 3.11 показано несколько хитроумных приемов ориентирования заклепок с тем, чтобы обеспечить их попадание головками вверх, в паз, расположенный слева. Используя ту особенность заклепок, что их длина больше, чем диаметр головок, усик опрокидывателя сбрасывает все заклепки, стоящие на головках. Затем отражатель отводит поток заклепок, если приемная щель переполнена и не может принять новые детали в том темпе, в каком они поступают снизу. Отражатель также сужает дорожку направляющей, обеспечивая подачу заклепок к щели по одной. Однако даже в этой самой узкой зоне заклепки могут проходить как головкой, так и стержнем вперед. Обратите внимание, что паз достаточно широк для стержня, но не пропустит головку заклепки. Стержень окажется в пазу вне зависимости от того, как подается заклепка. Если заклепка расположена на направляющей диагонально и стержень в паз не попадает, скос на краю направ-

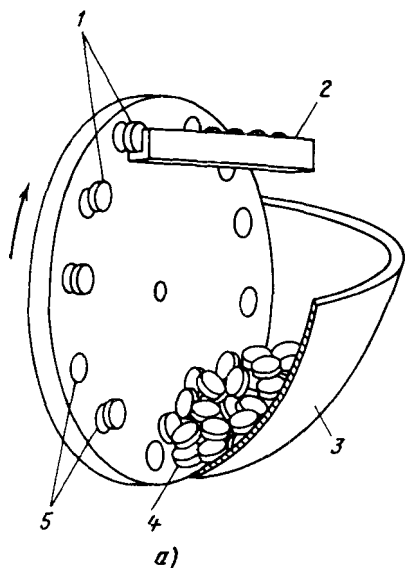


Рис. 3.10
Магнитные питающие устройства для относительно плоских деталей:

а — магнитный дисковый питатель: 1 — детали, захватываемые магнитами; 2 — спусковой желоб; 3 — неподвижный бункер; 4 — детали; 5 — магниты; 6 — магнитный элеваторный питатель: 1 — детали, проходящие по спусковому желобу; 2 — магниты; 3 — бесконечная цепь; 4 — стационарный бункер

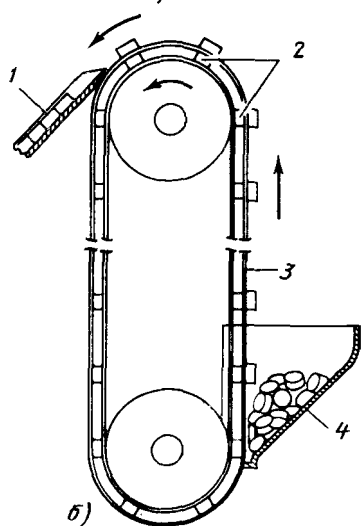
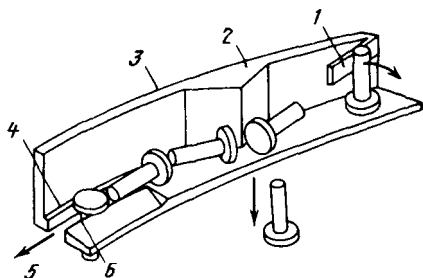


Рис. 3.11
Ориентирование винтов с головками или заклепок в верхней выходной части вибробункера:

1 — усик опрокидывателя сбрасывает все детали, не лежащие на боку; 2 — отражатель обеспечивает подачу деталей по одной и отводит поток, если приемная щель переполнена; 3 — стенка бункера; 4 — дорожка с щелью; 5 — спусковой желоб; 6 — приемная щель для ориентации деталей

ляющей заставляет заклепку падать в бункер для повторного прохождения.

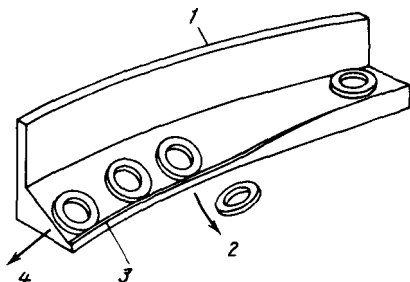
При подаче шайб или других дисков необходимо обеспечить их поступление одним слоем, сталкивая те из них, которые лежат сверху. Как показано на рис. 3.12, это обеспечивается за счет наклона направляющей и изготовления

на ней закраины с высотой несколько меньшей, чем толщина шайбы. При сортировке крышек для бутылок и других чашеобразных объектов усик опрокидывателя будет работать в том случае, если высота крышки меньше ее диаметра (рис. 3.13). Следует отметить, что это правило противоположно правилу, установленному ранее для винтов и заклепок. В случае крышек требуемой ориентацией является положение открытой стороной вверх. Специальный зубец сбрасывает все крышки, стоящие открытой стороной вниз. Наконечник, закрытая часть дорожки обеспечивает пово-

Рис. 3.12

Обеспечение в вибробункере подачи шайб одним слоем:

1 — стенка бункера; 2 — шайбы, не лежащие прямо на дорожке, падают на дно бункера; 3 — закраина, удерживающая шайбы, лежащие на дорожке; 4 — к спусковому желобу



рот крышки в требуемое положение, но некоторые крышки переворачиваются слишком рано, до подхода к закрытой части дорожки. Крышки должны

стоять одинаково открытой стороной вверх при подходе к последнему поворотному устройству.

Описанная выше система в первую очередь применима для небольших металлических деталей. Такие детали являются типовыми, однако этот метод применим не только для металлических объектов. Для разнообразия рассмотрим вращающийся барабан для ориентации пластмассовых бутылок (рис. 3.14). Поскольку миллиарды пластмассовых бутылок потребляются в год только в США, необходимость автоматизации здесь очевидна. Обратите внимание, что в устройстве, изображенном на рис. 3.14, имеются два лотка: один для бутылок, поступающих горлышком вперед, и другой для бутылок, поступающих дном вперед. Требуется только одно положение, для получения которого одну линию можно перевернуть и затем соединить с другой, образовав единую линию правильно ориентированных бутылок.

Габариты, жесткость и особенности материала пластмассовых бутылок предъявляют повышенные требования при производстве, сортировке и ориентации бутылок по сравнению с металлическими изделиями. Однако эти затруднения не идут ни в какое сравнение с возникающими при обработке и сортировании пищевых продуктов. Фрукты и овощи должны сортироваться, ориентироваться, а затем обрезаться, очищаться или нарезаться перед

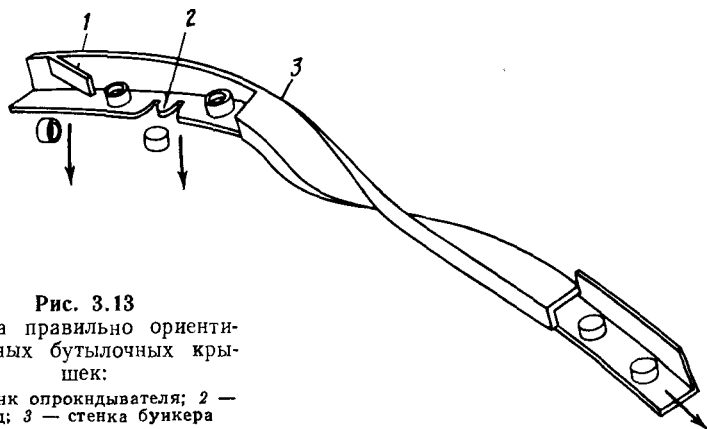


Рис. 3.13

Подача правильно ориентированных бутылочных крышек:

1 — усик опрокидывателя; 2 — зубец; 3 — стенка бункера

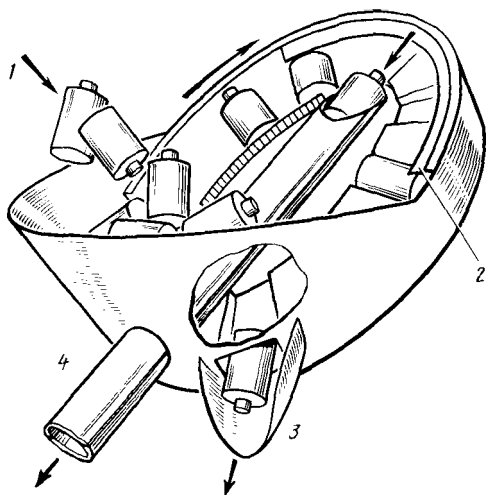


Рис. 3.14

Вращающийся барабан для ориентации пластмассовых бутылок:

1 — загрузка бутылок; 2 — предохранительный борт; 3 — первый лоток; 4 — второй лоток

консервированием или заморозкой. Фрукты и овощи разнообразны по форме и размерам, а также являются хрупкими и скоропортящимися. Попытка использовать обычные сортирующие и ориентирующие установки могут полностью провалиться. Можно, однако, в ка-

кой-то степени научно подойти к проблемам конструирования автоматического оборудования для ориентации фруктов и овощей, что будет сделано в следующем разделе.

3.2. ИССЛЕДОВАНИЕ ПРОЦЕССА ПОДАЧИ ДЕТАЛЕЙ

Без сомнения, возникновение идеи и разработка остроумной конструкции устройства подачи деталей являются проявлениями творческого гения конструктора. Однако при создании этих устройств, кроме творческого гения, требуется еще знание результатов экспериментальных исследований. Значительную работу в этой области проводят государственные и частные организации. Можно указать, например, на экспериментальный материал, представленный Р. Ф. Роджерсом и Дж. Бутридом [57], по исследованию влияния изменения ширины приемного паза на ориентацию детали заданной формы при подаче с помощью вибробункера (рис. 3.15). Задачей целевого ориентирующего устройства является обеспечение поступления деталей тяжелой стороной вперед вне зависимости от того, какой стороной она подходит к щели. Это кажется невыполнимым, но Роджерс и Бутрийд обнаружили, что существует некоторый рабочий диапазон, при котором все детали падают правильно, вне зависимости от первоначальной ориентации. Вне этого диапазона исследователи наблюдали различные явления. Сказанное иллюстрируется на рис. 3.15, где представлены также кривые вероятностей для наблюдаемых явлений, поскольку зоны их существования не имеют четких границ. Обратите внимание на линейность этих кривых, построенных по точкам, полученным экспериментально. Рабочий диапазон, обозначенный на рис. 3.15, является идеальным, однако изучение графиков показывает, что он может быть

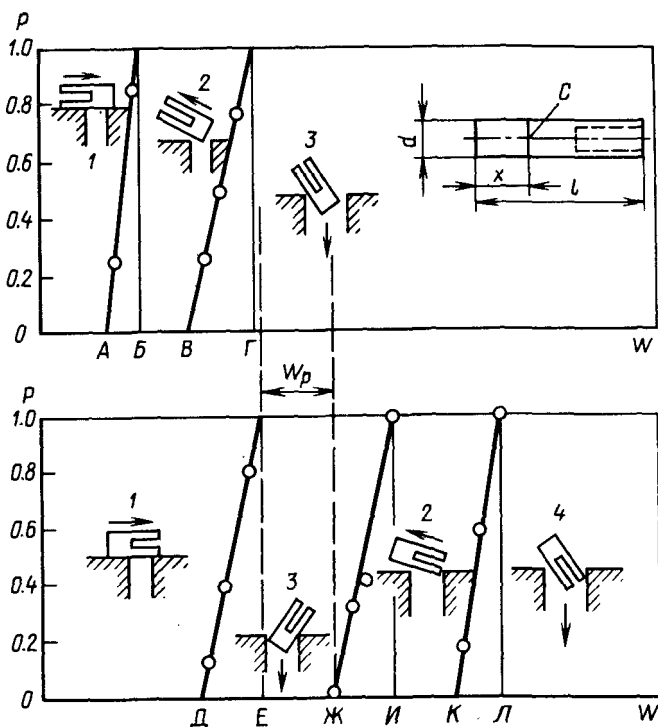


Рис. 3.15

Результаты экспериментального определения рабочего диапазона размеров паза при ориентировании детали заданной геометрической формы с помощью щелевого ориентирующего устройства. Параметры детали: $d = 0,25$ дюйма (6,35 мм); $l/d = 4,0$; $\bar{x}/l = 0,3$. Режим работы устройства: амплитуда колебаний $a_0 = 0,01$ дюйма (0,25 мм), угол наклона дорожки $\alpha = 0$; отклонение направления вектора колебаний от плоскости дорожки $\beta = 5,6^\circ$; P — вероятность наступления событий:

1 — проход детали над щелью; 2 — отбрасывание; 3 — падение тяжелой стороной вперед; 4 — падение легкой стороной вперед; W — ширина щели; W_p — рабочий диапазон; c — центр масс детали; верхний график — для поступления детали тяжелой стороной вперед, нижний — легкой стороной вперед

расширен, хотя и не будет таким эффективным, как при ширине паза, соответствующей диапазону $ЕЖ$. В диапазоне $ЖЗ$ все детали, поступающие тяжелой стороной вперед, упадут правильно, так же как и детали, поступающие легкой стороной вперед. Последние могут отбрасываться назад неопределенное количество раз, но в конце концов они упадут правильно. Ни одна деталь не упадет неправильно, если ширина щели не выйдет за границу, обозначенную точкой $И$. Действительно недопустимыми размерами щели являются размеры, соответствующие зонам $БВ$ и $ЗИ$. В зоне от $Б$ до $В$ все детали, поступающие тяжелой стороной вперед, отбрасываются назад, блокируя поток деталей. То же самое можно сказать о деталях, поступающих легкой стороной вперед при

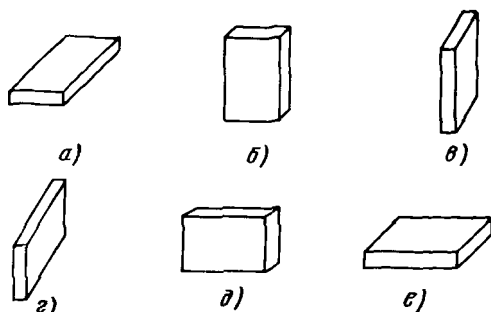


Рис. 3.16

Шесть возможных ориентированных положений параллелепипеда:

а — плашмя вдоль; б — стоя поперек; в — стоя вдоль; г — на ребре вдоль; д — на ребре поперек; е — плашмя поперек

ширине щели от $З$ до $И$. Это может привести даже к более серьезным проблемам, чем при ширине паза меньше $Д$, когда детали

просто проходят сверху. Исследование показывает, что проблемы подачи деталей являются более научными, чем кажется большинству. Безусловно, изменение размеров детали и смещение центра масс приводит к необходимости устанавливать новый рабочий диапазон. Однако для деталей стандартной формы, таких, как цилиндры, диски, кубы, призмы, винты и болты, изготовленных из конкретных материалов, можно вывести формулы, выражающие зависимость ширины паза от геометрии детали. Авторы исследования предлагают следующие уравнения для цилиндрических деталей при изготовленном из стали ориентирующем устройстве с пазом:

$$W_d = [(0,175d + \bar{x})^2 + d^2]^{1/2} + 5,7a_0; \quad (3.1)$$

$$W_g = l - \bar{x} - 0,32d, \quad (3.2)$$

где W_d — минимальная ширина щели в рабочем диапазоне; W_g — максимальная ширина щели в рабочем диапазоне; d — диаметр детали; $\bar{x}$ — расстояние от центра масс детали до ее ближайшего конца; l — длина детали; a_0 — амплитуда колебаний направляющей дорожки бункера.

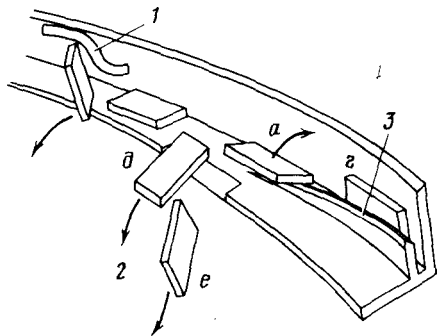
Для некоторых деталей вообще нет рабочего диапазона. Например, судя по рис. 3.15, если необходимо добиться ориентации легким концом вперед, такого диапазона не существует. Как показывают эксперименты, некоторые детали непригодны для подачи и ориентации, поэтому их конструкция должна рассматриваться как нежелательная при автоматизации. Успехи исследований в области подачи деталей таковы, что в зависимости от геометрии деталей их сразу можно разделить на идеально пригодные к автоматизации, условно пригодные и непригодные. Дж. Бутройд и П. Дьюхерст разработали диалоговую программу для ЭВМ, которая служит алгоритмом принятия решений при классификации и анализе их «собираемости». Программное обеспечение позволяет оценить стоимость и рассчитать эффективность автоматической и ручной сборки [9].

Эффективность сортировки. Используя теорию вероятностей, можно рассчитать эффективность и производительность для различных сочетаний сортировочных устройств, если известны «ве-

Рис. 3.17

Устройство для ориентации деталей в форме параллелепипеда, применяемое в вибробункерах:

1 — первая фаза ориентации — выступ, сбрасывает стоящие детали; 2 — вторая фаза ориентации — уклон, обеспечивает сбрасывание деталей, поступающих в поперечных положениях ($\theta \neq e$); 3 — третья фаза ориентации — направляющая, переводит детали из положения a в положение g



роятности перехода» для каждой фазы сортировки. Под термином «вероятность перехода» мы понимаем вероятность того, что деталь, подходя к сортировочному устройству в положении x , будет иметь положение y после взаимодействия с ним. Наиболее популярным видом сортировочного устройства является отражатель, который просто отбрасывает неправильно ориентированные детали в центр вибробункера или другого питателя. Идеальным отражателем называется устройство, которое обеспечивает 100 %-ную вероятность отбрасывания деталей, поступающих в некотором заданном положении. Устройство может быть идеальным отражателем для некоторых положений и неидеальным для других. Если устройство является идеальным отражателем для всех входных состояний, его можно назвать полным отражателем, поскольку оно сбрасывает все детали, соприкасающиеся с ним, вне зависимости от их ориентации. Понятие «полный отражатель» введено здесь только для пояснения термина «идеальный отражатель». В реальной автоматизации «идеальные» отражатели могут найти применение, а «полные» — нет.

В общем случае для сортировочного устройства может быть рассмотрена вероятность перехода детали из любого возможного положения в другое возможное положение, включая сбрасывание. Поэтому может быть составлена матрица вероятностей $n \times (n + 1)$, где n соответствует числу возможных положений. Например, параллелепипед имеет шесть возможных положений (рис. 3.16). Поэтому для полного описания общих характеристик устройства для ориентации деталей в форме параллелепипеда используется матрица 6×7 , в которой записываются 42 значения вероятностей. В реальных устройствах большинство вероятностей, записываемых в матрице, равно нулю или единице (0 или 100 %). Ниже рассмотрен пример сочетания устройств для ориентации и рассчитано распределение вероятностей получения на выходе той или иной ориентации.

Пример 3.1. Ориентация деталей в форме параллелепипеда. На рис. 3.17 изображено устройство для ориентации деталей в форме параллелепипеда в три этапа. Требуется так сориентировать деталь, чтобы она стояла вдоль на ребре, как это показано на рис. 3.16, приведенном выше. По заданным матрицам вероятностей перехода требуется рассчитать КПД и производительность устрой-

3.1. Матрица вероятностей перехода для первого ориентирующего приспособления

Исходное положение	Конечное положение						
	<i>a</i>	<i>b</i>	<i>в</i>	<i>г</i>	<i>д</i>	<i>е</i>	<i>x</i> (сбро- шено)
<i>a.</i> Плашмя вдоль	100	0	0	00	0	0	0
<i>б.</i> Стоя поперек	10	0	0	0	0	0	90
<i>в.</i> Стоя вдоль	0	0	0	0	0	0	100
<i>г.</i> На ребре вдоль	0	0	0	100	0	0	0
<i>д.</i> На ребре поперек	0	0	0	0	100	0	0
<i>е.</i> Плашмя поперек	0	0	0	0	0	100	0

3.2. Матрица вероятностей перехода для второго ориентирующего приспособления

Исходное положение	Конечное положение						
	<i>a</i>	<i>б</i>	<i>в</i>	<i>г</i>	<i>д</i>	<i>е</i>	<i>x</i> (сбро- шено)
<i>a.</i> Плашмя вдоль	100	0	0	0	0	0	0
<i>б.</i> Стоя поперек	0	100	0	0	0	0	0
<i>в.</i> Стоя вдоль	0	0	100	0	0	0	0
<i>г.</i> На ребре вдоль	0	0	0	100	0	0	0
<i>д.</i> На ребре поперек	0	0	0	0	0	0	100
<i>е.</i> Плашмя поперек	0	0	0	0	0	0	100

3.3. Матрица вероятностей перехода для третьего ориентирующего положения

Исходное положение	Конечное положение						
	<i>a</i>	<i>б</i>	<i>в</i>	<i>г</i>	<i>д</i>	<i>е</i>	<i>x</i> (сбро- шено)
<i>a.</i> Плашмя вдоль	100	0	0	0	0	0	0
<i>б.</i> Стоя поперек	0	100	0	0	0	0	0
<i>в.</i> Стоя вдоль	0	0	100	0	0	0	0
<i>г.</i> На ребре вдоль	0	0	0	100	0	0	0
<i>д.</i> На ребре поперек	0	0	0	0	0	0	100
<i>е.</i> Плашмя поперек	0	0	0	0	0	0	100

ства для различных распределений ориентаций и различной интенсивности подачи деталей на вход устройства.

Первым приспособлением для ориентации является выступ, который обеспечивает вероятность перехода детали из одного положения в другое в соответствии с матрицей, приведенной в табл. 3.1. Следует обратить внимание, что выступ является идеальным отражателем для положения «стоя вдоль», но срабатывает только в 90 % случаев для положения «стоя поперек». Оставшиеся 10 % относятся к переходу в положение «плашмя вдоль», что по-прежнему не является правильным положением. Деталь в остальных четырех положениях выступом не затрагивается.

Следующим приспособлением является уклон, обеспечивающий сбрасывание деталей, находящихся в положениях δ и ϵ , изображенных на рис. 3.16. Из матрицы вероятностей перехода, приведенной в табл. 3.2, ясно, что уклон является идеальным отражателем для этих положений, поскольку два последних ряда полностью заполнены нулями, за исключением графы x (сброшено), в которой стоит 100 %.

Последним приспособлением является наклонная направляющая, которая предназначена для перевода детали из положения a в положение $г$. Положим, например, что это приспособление срабатывает в 80 % случаев, а оставшиеся 20 % падают с дорожки, как это указано в матрице вероятностей перехода, приведенной в табл. 3.3.

Установив все 42 вероятности для каждой из трех матриц, можно непосредственно рассчитать распределение ориентаций на выходе после прохождения комбинации из трех приспособлений для каждого данного распределения ориентаций на входе и вычислить КПД системы, который определяется из уравнения

$$\text{КПД} = F_0/F_1, \quad (3.3)$$

где F_0 — производительность устройства по выходу правильно ориентированных деталей; F_1 — суммарная интенсивность подачи деталей на вход устройства.

Вычисление распределения ориентаций на выходе состоит в умножении вероятности каждой ориентации на входе на соответствующую вероятность перехода, указанную в матрице. Это делается последовательно для каждой фазы, пока не будет рассчитан выход всей системы.

Предположим следующее распределение ориентаций деталей на входе для примера 3.1:

Ориентация	Доля, %
a	30
b	10
$в$	5
$г$	5
δ	10
ϵ	10

Умножение этих цифр на вероятности перехода в первой фазе ориентации дает следующее распределение после первой фазы:

Ориентация	Вычисления	Новое распределение, %
a	$30 \cdot 1,0 + 10 \cdot 0,1$	31
b	10 · 0	0
$в$	5 · 0	0
$г$	$5 \cdot 1,0$	5
δ	$10 \cdot 1,0$	10
ϵ	$40 \cdot 1,0$	40
x (сброшено)	$10 \cdot 0,9 + 5 \cdot 10$	14

Продолжив вычисления по оставшимся фазам, получим результаты, приведенные в табл. 3.4. Любая такая таблица может быть проверена суммированием значений в каждом из первых двух столбцов. Сумма должна быть равна 100 %. Суммарная производительность (последняя колонка), просуммированная с общим числом сброшенных деталей (нижний ряд), тоже должна давать 100 %. Из итоговой табл. 3.4 можно установить, что КПД системы составляет 29,8 %.

3.4. Сводное распределение

Ориентация	Распределение, %			
	На входе	Первая фаза	Вторая фаза	Третья фаза (на выходе)
а. Плашмя вдоль	30	31	31	0
б. Стоя поперек	10	0	0	0
в. Стоя вдоль	5	0	0	0
г. На ребре вдоль	5	5	5	29,8
д. На ребре поперек	10	10	0	0
е. Плашмя поперек	40	40	0	0
х (сброшено)	14	14	50	6,2

Коэффициент полезного действия и эффективность. В примере 3.1 было подсчитано, что КПД набора ориентирующих устройств, установленных на дорожке вибробункера, составляет 29,8 %. Это может показаться слегка обескураживающим, но необходимо помнить, что устройство должно было автоматически обеспечить единственно правильную ориентацию из шести возможных и что распределение этих ориентаций на входе было очень неблагоприятным: только 5 % деталей во входном потоке сразу имели правильную ориентацию. Ориентирующая система устройств в примере 3.1 предназначена для того, чтобы привлечь внимание к проблеме ориентации. Эта проблема возникает в большинстве реальных примеров автоматической сборки в промышленности. Смотреть на то, как работает промышленный вибробункер, очень увлекательно. Совсем не редко неправильно ориентированные детали в больших количествах падают на дно бункера, чтобы повторить свой путь опять.

Другим способом оценки описанного устройства является использование понятия «эффективность» вместо КПД. Эффективность E определяется следующим образом:

$$E = F_0/F_T, \quad (3.4)$$

где F_0 — производительность устройства по выходу правильно ориентированных деталей (аналогично определению для КПД); F_T — суммарная интенсивность выхода деталей.

Эффективность сортировочного устройства нагляднее всего определяется долей правильно ориентированных деталей. Сортировочное устройство в примере 3.1 имеет эффективность 100 %, поскольку все детали в выходном потоке правильно ориентированы. Чтобы применяться в реальных производственных условиях, система сортировки и ориентации фактически должна иметь эффективность 100 %. Любое отклонение от идеальной эффективности 100 % в реальной системе приводит к заклиниванию детали, проблеме с неожиданно серьезными последствиями, как это показано в гл. 4.

Хотя вся система ориентации обычно эффективна почти на 100 %, сортировочные приспособления, из которых система состоит, значительно менее эффективны. Фактически первое сортировочное приспособление в примере 3.1 было только на 6 % эффективным для данного распределения на входе, что вычисляется следующим образом:

$$F_0 = 30 \cdot 0 + 10 \cdot 0 + 5 \cdot 0 + 5 \cdot 1,00 + 10 \cdot 0 + 40 \cdot 0 = 5;$$

$$F_T = 30 \cdot 1,00 + 10 \cdot 0,10 + 5 \cdot 0 + 5 \cdot 1,00 + 10 \cdot 1,00 + 40 \cdot 1,00 = 86;$$

$$E = \frac{5}{86} = 6 \, \%.$$

Эту цифру можно получить непосредственно из второй колонки табл. 3.4 как отношение производительности по ориентации к сумме производительностей по всем ориентациям для первого приспособления, исключая, конечно, сброшенные детали, которые вообще не появляются в выходном потоке.

Эффективность второго приспособления, составляющая 14 %, немногим больше. Изыщество интегрированной системы сортировочно-отбраковочных приспособлений заключается в суммировании их эффектов при последовательном применении. В одиночку ни одно из приспособлений примера 3.1 не очень эффективно, но в сумме устройство эффективно практически на 100 %.

Эффективность является интересным критерием при анализе, особенно для отдельной сортировочной позиции. Однако, поскольку эффективность большинства комбинированных сортировочных устройств приближается к 100 %, КПД является более удобным показателем для их оценки.

Износ и повреждение деталей. Наконец, следует рассмотреть возможность заклинивания деталей на одном из приспособлений, когда детали поднимаются, вибрируя, и падают на дно, снова и снова сталкиваемые приспособлением. Если эффективность практически 100 %, можно использовать КПД системы для вычисления вероятности того, что деталь будет отброшена k раз до того, как приобретет допустимую ориентацию. Формула расчета

$$P_k = \frac{E}{100} \left(1 - \frac{E}{100} \right)^k, \quad (3.5)$$

где P_k — вероятность того, что деталь будет отброшена k раз; E — эффективность; k — число отбрасываний.

Для примера 3.1

$$P_0 = \frac{E}{100} = \frac{29,8}{100} = 0,298;$$

$$P_1 = \frac{E}{100} \left(1 - \frac{E}{100} \right)^1 = 0,298 \cdot 0,702 = 0,209;$$

$$P_2 = \frac{E}{100} \left(1 - \frac{E}{100} \right)^2 = 0,298 \cdot 0,702^2 = 0,147;$$

3.5. Вероятности отбрасывания для примера 3.1

Число отбрасываний k	Вероятность k отбрасываний $\frac{E}{100} \left(1 - \frac{E}{100}\right)^K$	kP_k	$\sum_{i=0}^k iP_i$
0	0,298	0	0
1	0,209	0,209	0,209
2	0,147	0,294	0,503
3	0,103	0,308	0,812
4	0,072	0,289	1,102
5	0,051	0,254	1,356
6	0,036	0,214	1,570
7	0,025	0,175	1,745
8	0,018	0,141	1,886
9	0,012	0,111	1,997
10	0,009	0,087	2,083
...	...	...	...
x	0	0	2,356- k

$$P_3 = \frac{E}{100} \left(1 - \frac{E}{100}\right) = 0,298 \cdot 0,702^3 = 0,103;$$

$$P_{10} = \frac{E}{100} \left[1 - \frac{E}{100}\right]^{10} = 0,298 \cdot 0,702^{10} = 0,009.$$

Таким образом, приблизительно одна деталь из 100 будет отброшена 10 раз, прежде чем получит допустимую ориентацию. Инженер по автоматизации должен решить, повредит или нет такое обращение изделию.

Среднее число отбрасываний $\bar{k}$ для каждой детали, проходящей устройство ориентации, обладающее эффективностью 100 %, также является функцией эффективности и определяется следующим выражением:

$$\bar{k} = \sum_{i=0}^{\infty} iP_i = \sum_{i=0}^{\infty} i \frac{E}{100} \left(1 - \frac{E}{100}\right)^i. \quad (3.6)$$

Можно показать, что бесконечный ряд, который представлен в (3.6), эквивалентен конечному выражению

$$k = \frac{1 - \frac{E}{100}}{\frac{E}{100}}. \quad (3.6a)$$

Для условий, приведенных в примере 3.1, значение $\bar{k} = 2,356$ может быть вычислено по (3.6a). Уравнение (3.6) позволяет рассчитывать все вероятности отбрасывания, как это показано в табл. 3.5.

Ориентация пищевых продуктов. В консервной промышленности сортировка, ориентация и подача пищевых продуктов может представлять даже большие трудности с точки зрения анализа

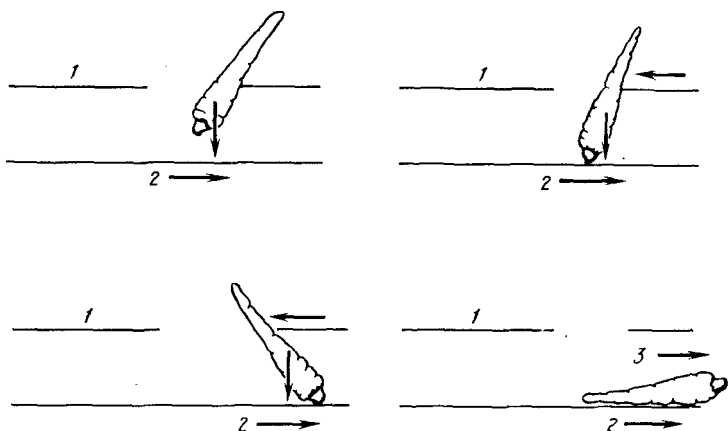


Рис. 3.18

Схема работы устройства для сортировки замороженных овощей:

1 — вибростол; 2 — подающий конвейер; 3 — направление подачи под резак

из-за имеющихся неизбежных случайных отклонений. Исследования в области автоматизации, выполненные Квеем в Арканзасском университете [30], привели к разработке оптимальной конструкции подающего устройства для нарезки овощей в процессе заморозки (рис. 3.18). В месте, где морковь переходит в ботву, имеется часть, которая рассматривается как нежелательная. Механические резаки могут срезать эти верхушки, если под них подан соответствующий конец моркови. На рис. 3.18 показан механизм для сортировки и транспортирования, задачей которого является подача моркови под резак верхним концом вперед. Морковь высыпается с обычного конвейера на вибростол с большим количеством отверстий. Верхний конец моркови тяжелее, и она, скорее всего, провалится в отверстие этим концом вперед, вне зависимости от того, как она лежала на столе. Что с морковью происходит дальше, в принципе зависит от соотношения между ее длиной и зазором между подающим конвейером и вибростолом. Другими факторами являются скорость подающего конвейера и наклон всего устройства. Если принять, что вся морковь имеет одинаковую длину, проблема сильно облегчается. Однако при анализе конструкции подающей системы необходимо учитывать случайные отклонения в размерах моркови. Несколько упрощенный случай расчета приведен в примере 3.2.

Пример 3.2. Конструкция подающего устройства для замороженных овощей. На рис. 3.18 показана часть автоматической системы для приготовления замороженной моркови. Здесь показано, как морковь проваливается в отверстия вибростола и попадает на подающий конвейер, приобретая при этом правильную ориентацию. Если морковь слишком коротка по отношению к зазору между вибростолом и конвейером, она может упасть тонким концом вперед, а комлем в неправильном направлении. С другой стороны, если зазор слишком мал, морковь не пройдет и забьет всю систему. Это вызовет сбой автоматической линии, что

считается вдвое более серьезной неприятностью, чем ориентация «не тем концом». В качестве приближения к действительности примем, что забивание обязательно произойдет, если зазор на 1 дюйм (25,4 мм) меньше, чем длина моркови, и никогда не произойдет, если зазор на 1 дюйм (25,4 мм) больше. Приблизительно примем, что морковь всегда упадет неправильно, если зазор меньше, чем длина моркови.

Будем считать, что длина моркови распределена по нормальному закону со средней 3 дюйма (76,2 мм) и стандартным отклонением 0,5 дюйма (12,7 мм).

а) Каким должен быть зазор?

б) Какая доля поступающей моркови будет забивать конвейер?

в) Какая доля моркови будет неправильно обрезана?

Решение. Пусть L — длина моркови (нормальное распределение, $\mu = 3$ дюйма (76,2 мм), $\sigma = 0,5$ дюйма (12,7 мм)); $F(L)$ — интегральная функция распределения от L ; G — зазор, дюйм; G^* — оптимальный зазор; P — вероятность; $Q(G)$ — штрафная функция = P (неправильной обрезки моркови) + $2P$ (застревание);

$$Q(G^*) = \min G Q(G);$$

$$Q(G) = P(G \geq L) + 2(P[G \leq L - 1]) = P(L < G) + 2(P[L \geq G + 1]) = F[G] + 2(1 - F[G + 1]).$$

Пусть $z = (G - \mu)/\sigma$ — стандартная нормальная случайная переменная;

$$\Phi(z) = \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz — \text{интегральная функция распределения } z; z_1 = \frac{G - 3}{0,5} —$$

точка, в которой морковь будет обрезана неправильно; $j_1 = 2G - 6$; $z_2 = \frac{G + 1 - 3}{0,5}$ — точка, в которой морковь будет застревать, $j_2 = 2G - 4$;

$$Q(G) = \Phi[z_1] + 2(1 - \Phi[z_2]) = \Phi[z_1] + 2\Phi[-z_2].$$

Минимум этой функции может быть найден обычным путем с помощью дифференцирования. Интуиция подсказывает, что зазор должен быть больше двух дюймов, потому что застревание является более серьезной проблемой, чем обрезка не с того конца. Минимум штрафной функции наблюдается при зазоре около 2,7 дюйма (68,58 мм), поэтому $G^* = 2,7$.

Используя таблицы значений интегральной функции стандартного нормального распределения, имеем:

доля неправильно обрезанной моркови 27,43 %; доля застрявшей моркови 8,08 %;

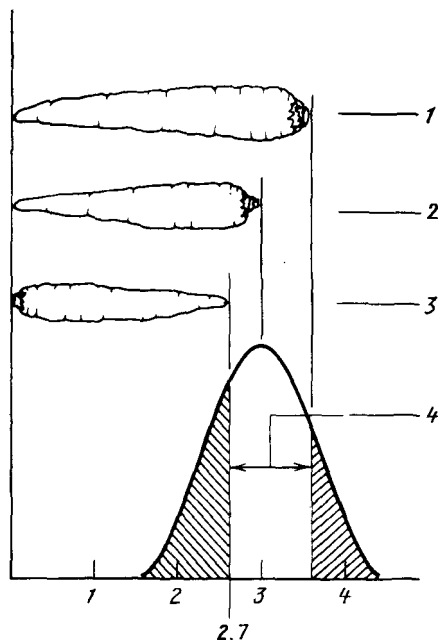
$$Q(G^*) = 0,2743 + 2(0,0808) = 0,4359.$$

Анализ приведенной в примере системы для переработки моркови потребовал некоторых знаний в области случайных переменных и теории статистики, но даже для читателей, не имеющих соответствующей подготовки, очевидно сходство поставленной проблемы с проблемой автоматической подачи деталей. Как и в случае целевого ориентирующего устройства, должен быть установлен рабочий диапазон. Для моркови с номинальной длиной три дюйма (76,2 мм) зазор между столом и конвейером может быть установлен от двух до трех дюймов (от 50,8 до 76,2 мм) без всяких проблем. Однако морковь бывает разного размера, и рабочий диапазон для моркови длиной в три дюйма не тот, что для моркови длиной пять дюймов. К счастью, большинство реальных статистических распределений имеет тенденцию группиро-

Рис. 3.19

Определение оптимального зазора и допустимого диапазона длин моркови при нормальном распределении размеров (пример 3.2). Оптимальный зазор 2,7 дюйма (68,6 мм): 1 — морковь слишком длинная и может забить отверстия; 2 — морковь среднего размера; 3 — морковь слишком коротка и может быть обрезана неправильно; 4 — допустимый диапазон длин

ваться вокруг центра, и конструкции систем автоматизации могут быть настроены на диапазон в середине распределения, охватывая основную массу объектов. Так было сделано в примере 3.2, что иллюстрируется на рис. 3.19, хотя допустимый диапазон несколько сдвинут в сторону больших длин из-за большего вреда от забивания устройства.



Частичная автоматизация. График на рис. 3.19 подводит и к другой проблеме. Такое большое количество моркови попадает в те зоны распределения, которые относятся к забиванию устройства и неправильной обрезке моркови, и становится очевидной неудовлетворительная работа устройства. На реальном предприятии, откуда был почерпнут пример 3.2, точно так и случилось (за исключением того, что это была не морковь). Разнообразие размеров продукции заставило инженеров пойти на компромисс и ограничиться частичной автоматизацией. По обеим сторонам вибростола были предусмотрены места для работников, которые вручную переворачивали неправильно упавшие овощи до их поступления под обрезку.

Частичная автоматизация применяется не только в пищевой промышленности, где переменность характеристик продукции является главной проблемой. Ориентация и выверка различных видов продукции и деталей машин может потребовать интеллектуального технического зрения или какого-либо другого технического ухищрения, что может сделать данный производственный процесс неэффективным с технической или экономической точки зрения. Частичная автоматизация, при которой рабочий и машина делают свое дело в меру своих способностей, является альтернативой, которую инженер по автоматизации не должен упускать из виду.

В этой главе рассмотрена проблема перемещения материалов или деталей в позицию для обработки или сборки. Хотя изученные операции кажутся сложными, а иногда построены очень

изобретательно, полезных изменений с продукцией в результате не происходит. Подача деталей и их ориентация, без сомнения, являются полезными процессами, но сейчас мы готовы рассмотреть автоматизацию машин, которые обрабатывают или собирают эти детали в полезный продукт.

В ы в о д ы

Успех проекта автоматизации сборки зависит не столько от в высшей степени заметных движений промышленного робота, перемещающего детали, сколько от маленьких хитростей позиционирования и ориентации деталей для сборки. Иногда этих мелких проблем можно избежать, сохраняя исходную ориентацию во время обработки или с помощью специальной тары, в которой отгружаются покупные детали.

Если необходимо обеспечить ориентацию деталей, можно воспользоваться хитроумными механическими устройствами, основным представителем которых является вибробункер с ориентирующими приспособлениями на выходной дорожке. Детали, которые сортирующая система бункера не может пропустить или сориентировать, отбрасываются назад для повторного прохождения. Отброшенные детали не пропадают, теряются только время и производительность. Потеря производительности от отбрасывания неправильно сориентированных деталей прямо зависит от КПД вибробункера и сортирующих приспособлений. Суммарная эффективность системы может быть рассчитана с учетом матриц вероятностей сортировки для каждого из приспособлений, расположенных по дорожке вибробункера. Если необходимо повысить производительность, можно повысить уровень вибраций, что до некоторого момента повышает подачу. Наука об ориентации и подаче объектов охватывает не только мелкие металлические детали, но и пластмассовые бутылки, молдинги, отливки и даже пищевые продукты. Пищевые продукты представляют собой проблему для инженера по автоматизации из-за разброса размеров и форм объектов.

У п р а ж н е н и я и в о п р о с ы

3.1. Опишите условия, при которых можно заказать поставку покупных изделий с обеспечением определенной ориентации для роботизированной сборки и при этом по ценам ниже обычных (при поставке в ящике навалом).

3.2. Назовите несколько методов сообщения движения деталям при подготовке к ориентации и сборке. Какой из них наиболее популярен?

3.3. Бункер с вращающимся основанием и центробежный бункер предназначены для подачи деталей цилиндрической формы. В чем их различие?

3.4. Рассмотрев график влияния ширины приемного паза на ориентацию детали, изображенной на рис. 3.15, определить рабочий диапазон ширины паза для подачи деталей легким концом вперед, если такой диапазон имеется.

3.5. По графику на рис. 3.15 рассчитайте, какая доля деталей вероятно упадет правильно (тяжелым концом вперед), исходя из условий, приведенных в табл. 3.6, а и 3.6, б.

3.6. а. Условия упражнения 3.5

Положение	Входной поток		Ширина паза, дюйм
	тяжелый конец вперед, %	легкий конец вперед, %	
<i>а</i>	20	80	0,50
<i>б</i>	80	20	0,50
<i>в</i>	20	80	0,65
<i>г</i>	80	20	0,65
<i>д</i>	20	80	0,40
<i>е</i>	80	20	0,40

3.6. б. Условия упражнения 3.5

Точки на рис. 3.15	Ширина паза, дюйм
<i>А</i>	0,12
<i>Б</i>	0,19
<i>В</i>	0,29
<i>Г</i>	0,43
<i>Д</i>	0,31
<i>Е</i>	0,45
<i>Ж</i>	0,59
<i>И</i>	0,73
<i>К</i>	0,85
<i>Л</i>	0,96

3.6. В табл. 3.7 приведена матрица вероятностей перехода для ориентирующего устройства, установленного на дорожке вибробункера: а) Является ли устройство отражателем? б) Если да, является ли отражатель идеальным? в) Является ли отражатель абсолютным? г) Какова требуемая ориентация? д) Будет ли полезной установка второго аналогичного ориентирующего устройства дальше вверх по дорожке?

3.7. Если реально возможные варианты ориентации деталей на входе в устройство, описываемое в табл. 3.7, равномерно распределены между четырьмя показанными вариантами, каков будет КПД сортировочной установки? Какова эффективность установки?

3.8. Выполните задание 3.7, рассчитав эффективность системы из двух идентичных сортировочных установок, соединенных последовательно.

3.9. Можно ли достигнуть 100 % эффективности последовательным соединением сортировочных устройств, работающих в соответствии с данными, указанными в табл. 3.7.

3.10. Предположим, что необходимо обеспечить эффективность 100 % в ориентирующей системе, показатели которой указаны в табл. 3.7. Это, конечно, невозможно, но мы можем добавить еще одно сортировочное приспособление, являющееся идеальным отражателем для одного или более положений. Этот идеальный отражатель может быть помещен до или после имеющегося приспособления. Для какого исходного состояния (или состояний) дополнительное сортирующее приспособление должно быть идеальным отражателем, чтобы получить общую эффективность системы 100 %? Где отражатель должен быть помещен, до или после сортирующего приспособления? Сравните КПД системы для этих двух вариантов.

3.11. Заполните пропущенные элементы в матрице вероятности перехода, помещенной в табл. 3.8, и вычислите затем эффективность сортирующего приспособления при следующих распределениях входных ориентаций, если требуемая ориентация а: а) 50 %, б) 10 %; в) 30 %; г) 5 %; д) 5 %.

3.7. Матрицы вероятностей перехода для упражнения 3.7

Положение	<i>а</i>	<i>б</i>	<i>в</i>	<i>г</i>	<i>х</i>
<i>а</i>	100	0	0	0	0
<i>б</i>	20	0	0	0	80
<i>в</i>	20	0	10	0	70
<i>г</i>	10	0	0	30	60

3.8. Матрица вероятностей перехода для упражнений 3.11 и 3.12

Положение	а	б	в	г	д	х
а	40	30	20	10		
б	10	10	10	10	10	
в						100
г	20	20	20	20		20
д	30	25		20	0	10

3.12. Приняв одинаковое распределение по пяти входным ориентациям, определите, по каким выходным ориентациям сортирующее устройство, показателя которого указаны в табл. 3.8, наиболее эффективно? Какова эффективность ориентирующего приспособления для этой ориентации? Каков КПД для этой ориентации?

3.13. Вибрационная сортирующая система обладает эффективностью 100 % при КПД 80 %. Какова вероятность того, что деталь будет сброшена в бункер 5 раз до того, как пройдет систему? Каково среднее количество сбрасываний детали для этой системы?

3.14. Техническими требованиями к конструкции некоторой системы подачи деталей оговаривается, что вероятность сбрасывания детали 10 раз и более должна поддерживаться на уровне ниже 1 %. Система практически идеальна в отношении эффективности, но имеет КПД только 40 %. Отвечает ли данная конструкция техническим требованиям?

4. АВТОМАТИЧЕСКОЕ ПРОИЗВОДСТВО И АВТОМАТИЧЕСКАЯ СБОРКА

Целью ориентации и подачи деталей, о которых рассказывалось в гл. 3, является их подготовка к автоматической обработке или сборке. В производственных линиях и сборочных установках используются разнообразные пневматические и электрические приводы и исполнительные механизмы общего применения, описанные в гл. 2. Хотя в данной главе основной упор сделан на операцию сборки, многие принципиальные положения применимы и к другим автоматическим процессам, например выполняемым многопозиционными прессами с автоматической передачей обрабатываемой заготовки с одной позиции на другую, автоматизированными сверлильными станками или токарно-револьверными автоматами.

4.1. СБОРОЧНЫЕ УСТАНОВКИ

Конструкция сборочных установок настолько тесно связана с конкретными сборочными операциями, что их трудно классифицировать в объеме этой книги. Существуют, однако, некоторые операции, которые являются общими для большинства сборочных процессов. Одна из них — завертывание винтов — рассматривается ниже.

Автоматические винтоверты. Большинство из нас представляют автоматические винтоверты в виде приводных инструментов, которые оператор держит в руках и которые предназначены для ускорения завинчивания винтов вручную. Однако существуют машины, которые полностью выполняют все операции. Винты навалом высыпаются в вибробункера, где они ориентируются и подаются на выходную дорожку, ведущую к месту, откуда приводной винтоверт автоматически перемещается в позицию, где завертываются винты. На рис. 4.1 изображена конструкция, заимствованная из работы [12]. Она имеет много общих черт с человеком — ее конкурентом в завертывании винтов. Механические захваты или «пальцы» забирают винт с подающей дорожки, держат его прямо, совмещая ось винта с осью винтоверта, и затем опускаются вместе с ним, продолжая удерживать винт до тех пор, пока лезвие винтоверта не попадет в прорезь винта и не повернет

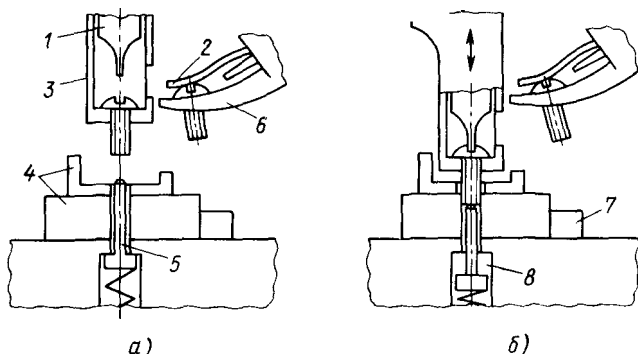


Рис. 4.1

Автоматическое завинчивание винтов:

а — захваты получают винт из устройства для подачи и ориентации; б — захваты опускаются вместе с инструментом до тех пор, пока винт не войдет в отверстие, отжав базирующий штифт. Как только винтоверт входит в зацепление со шлицом винта и начинает прилагать определенный вращающий момент, захваты расходятся; 1 — вращающееся лезвие; 2 — удерживающая пружина; 3 — захваты; 4 — соединяемые детали; 5 — подпружиненный базирующий штифт; 6 — держатель; 7 — базирующий упор; 8 — крепежное приспособление

его на несколько оборотов. Обратите внимание на то, что человек-оператор аналогичным образом держит винт, когда завертывает его вручную.

Сам привод может быть электрическим или пневматическим. Каждый вариант имеет свои преимущества, что показано в гл. 2. Поскольку привод при достижении определенного вращающего момента должен отключаться, применяют муфту или какое-либо другое средство для отключения. Пневмодвигатель может сам обеспечивать приблизительное постоянство момента за счет правильной регулировки давления воздуха. Если же требуется повысить давление, чтобы обеспечить частоту вращения, диктуемую технологией, или если применяется электрический винтоверт, используют фрикционную или храповую муфту.

Некачественное изготовление собираемых деталей или винтов, которые их скрепляют, может привести к таким проблемам, как недостаточное усилие затяжки, незавернутые винты, сорванные резьбы и поврежденные головки винтов. Это вопрос контроля качества, а в целом важность контроля качества компонентов возрастает при автоматизации сборки. Но даже при качественных деталях автоматическое заворачивание винтов и другие операции автоматической сборки требуют точной фиксации деталей.

Сборка гайки с болтом весьма похожа на операцию свинчивания. Болт может быть установлен в нужное положение роботом или линейным приводным механизмом. Для наворачивания гайки можно использовать механизмы, аналогичные винтоверту на рис. 4.1. Для более крупных деталей и корпусов целесообразно использовать специальные крепежные приспособления. Промышленные роботы со специальными захватами также используются для этих объектов.

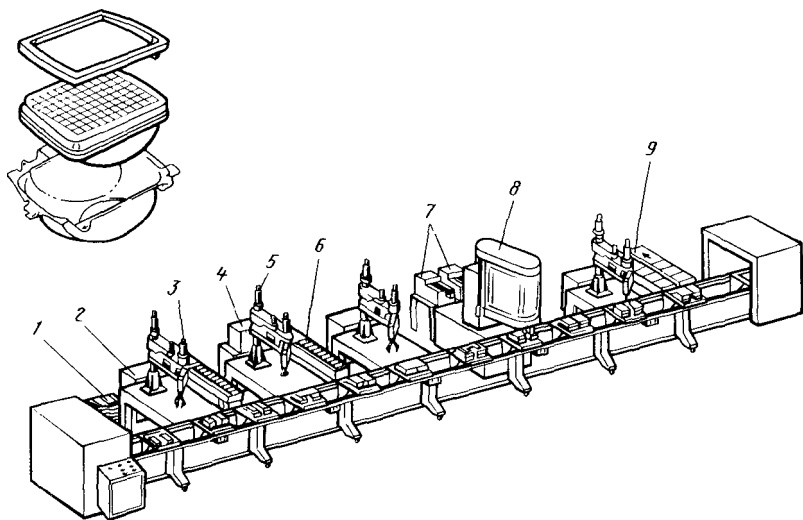


Рис. 4.2

Асинхронная автоматическая линия для сборки плафонов, устанавливаемых в салонах автомобилей:

1 — задние крышки; 2 — контроллер; 3 — сигнальная лампа; 4 — устройство программирования; 5 — роботы модели «Чарли 4»; 6 — окантовки; 7 — устройства для подачи винтов; 8 — восьмишпильный винтовёрт; 9 — выгрузка собранных плафонов

Тактовые установки. Автоматический гайковерты, винтовёрты и другие автоматические сборочные инструменты полезны для автоматизации, но их не называют сборочными установками. Термин «сборочная установка» обычно подразумевает существование нескольких связанных между собой позиций, иногда со складами между ними. Сборочные установки принципиально отличаются друг от друга методом организации такта или передачи частично собранных изделий из одной позиции в другую.

Существует два основных типа тактовых установок: линейные (рис. 4.2) и карусельные. Примером линейного построения является линия сборки автомобилей, хотя в таких линиях сохраняется очень много ручных операций. Линейное построение обладает большей гибкостью в том смысле, что ограничений по количеству позиций нет. Еще более важным свойством, вероятно, является простота, с которой эти линии могут дооснащаться межоперационными буферными складами. Эта возможность является ключом к повышению производительности линий, состоящих из установок со значительно отличающимся индивидуальным временем цикла.

Карусельные сборочные установки чаще применяются для сборки небольших деталей. Эти установки иногда называют круговыми тактовыми установками. На круглом столе имеется несколько позиций, в которых установлена собираемая продукция в различных стадиях сборки. Изделие собирается по мере переме-

щения по кругу. Движение вращающегося стола может быть непрерывным, как у установок для розлива в бутылки, но при этом требуется, чтобы сборочное устройство перемещалось вместе со столом. Большинство сборочных установок на заводах работают по принципу прерывистого шага. Одна позиция используется для установки базовой детали, основания или под сборки на вращающийся стол. Эта операция может выполняться вручную или промышленным роботом. После того как стол почти полностью заканчивает поворот, собранное изделие снимается с позиции, соседней с позицией для установки, а иногда для этого используется сама установочная позиция, которая в этом случае служит как для установки, так и для снятия изделия.

4.2. ПРОИЗВОДИТЕЛЬНОСТЬ И ВРЕМЯ ПРОИЗВОДСТВА

Автоматические сборочные установки как карусельного, так и линейного типа выпускают собранное изделие каждый раз, когда завершается такт, вне зависимости от того, сколько позиций в процессе сборки. Чтобы подсчитать производительность (идеальную) автоматической сборочной машины, необходимо знать только время такта, а количество позиций несущественно. Действительно, время производства, т. е. время, необходимое для сборки изделия от начала до конца, зависит и от числа позиций и от времени такта, а производительность не зависит.

Пример 4.1. Производительность и время производства. Автоматическая сборочная установка круговой тактовой конструкции имеет восемь позиций ($N = 8$). Стол поворачивается с помощью мальтийского механизма, у которого кривошип вращается с частотой $n = 30$ об/мин. Какова производительность установки и время производства?

Решение. В гл. 2 описан принцип работы мальтийского механизма. За каждый поворот кривошипа стол перемещается на одну позицию, поэтому производительность $P = 30$ изделий/мин.

Время выпуска продукции

$$t = 1/P = 60/30 = 2 \text{ с/изделие.}$$

Время производства

$$T = tN = 2 \cdot 8 = 16 \text{ с.}$$

Мы сконцентрировали внимание на идеальной производительности без учета неполадок, но идеальная производительность никогда не может быть достигнута. Неполадки в работе на одной из позиций при автоматической сборке могут серьезно повлиять на производительность, о чем будет рассказано далее.

Неполадки в машине. Подводные камни на пути к автоматизации легко упустить из виду. Одним из наиболее ярких примеров может служить попытка наладить автоматическую многопозиционную сборочную установку без учета потенциального влияния неполадок и сбоев. Рассмотрим восьмипозиционную ($N = 8$) карусельную установку, приводимую мальтийским механизмом, у которого время индексации $t_1 = 3$ с, а время выстоя $t_2 = 5$ с.

Справедливость отношения времени индексации к времени выстоя можно проверить, используя принципы, рассмотренные в гл. 2. При идеальных условиях работы (без сбоев) длительность цикла восьмипозиционной установки при выпуске законченной сборочной единицы $T = 8$ с, что соответствует производительности $P_1 = 450$ сборок/ч.

Предположим, что на каждой позиции происходит один сбой на каждые 100 циклов. Этот уровень кажется допустимым. Сбой на одной или нескольких позициях немедленно вызовет перебой в работе тактовой установки, что потребует вмешательства оператора для наладки и повторного запуска. Предположим, что на это потребуются примерно 10 мин. Даже такое минимальное проявление законов Мерфи выливается в значительное уменьшение производительности автоматической сборочной машины, как будет показано далее в расчетах.

Если вероятность сбоя на позиции $p = 0,01$, то вероятность того, что сбой не произойдет в данном цикле, $p_1 = 99\% = 0,99$. Чтобы сборка была успешно выполнена, все восемь позиций должны работать без сбоев. Поэтому вероятность отсутствия сбоев в данном цикле равна произведению вероятностей отсутствия сбоев на каждой позиции во время выполнения цикла. Перемножая эти вероятности, получим

$$p_2 = p_1^N = 0,99^8 = 0,9227.$$

Следовательно, за 10 000 циклов без сбоев будет изготовлено $N_1 = 9227$ сборок с длительностью цикла $T = 8$ с, для этого потребуется время

$$t_3 = N_1 T = 9227 \cdot 8 = 20,5 \text{ ч.}$$

Во время остальных $N_2 = 10\,000 - 9227 = 773$ циклов по крайней мере на одной позиции произойдет сбой, требующий времени $t_4 = 10$ мин на восстановление. На это уйдет время

$$t_5 = N_2 t_4 = 773 \cdot 10 = 128,83 \text{ ч.}$$

Суммарное время на производство 9227 сборок t_6

$$t_6 = t_3 + t_5 = 20,5 + 128,83 = 149,33 \text{ ч,}$$

включая производственное время и время простоев. Обратите внимание, какое из чисел больше. Время простоев t_7 в долях всего времени на производство

$$t_7 = t_5/t_6 = 128,83/149,33 = 0,863 = 86,3\%.$$

Суммарная производительность

$$P_2 = N_1/t_6 = 9227/149,33 = 61,8 \text{ сборок/ч.}$$

Коэффициент полезного действия сборочной машины, представляющий собой отношение фактической производительности к идеальной,

$$\text{КПД} = P_2/P_1 = 61,8/450 = 0,137 = 13,7\%.$$

При незначительном (1 %) уровне сбоев при сборке наблюдается удивительно большое (на 86,3 %) падение КПД. Таков мир автоматизации, и инженер, работающий в этой области, планируя использование новой автоматической сборочной установки, должен учитывать реальные сбои и простои.

Контроль качества компонентов. Выше показано, какое большое отрицательное значение имеют сбои в работе автоматических сборочных установок. Самой важной причиной, вызывающей сбой при автоматической сборке, являются случайные отклонения параметров компонентов собираемых изделий, при которых автоматическая сборка не может быть произведена. Если могут быть ужесточены допуски или введен более жесткий контроль качества компонентов по существующим допускам, инженер по автоматизации сможет добиться удивительного повышения производительности сборочных установок. Предположим, что в предыдущем примере 90 % сбоев машины происходит из-за некачественных компонентов. Решение проблем их качества сократит уровень сбоев с одного на каждый 100 циклов до одного на каждые 10 000 циклов. Пересчитаем производительность и долю простоев для новых условий.

Вероятность того, что за типичный цикл произойдет по меньшей мере один сбой, составит

$$p' = 0,999^8 = 0,9920.$$

Значит, за 10 000 циклов будет собрано $N_3 = 9920$ изделий при длительности цикла $T = 8$ с. На это потребуется время

$$t_8 = N_3 T = 9920 \cdot 8 = 22,04 \text{ ч.}$$

Во время оставшихся $N_4 = 10\,000 - 9920 = 80$ циклов по крайней мере на одной позиции произойдет сбой, что потребует $t_9 = 10$ мин на восстановление. В целом на это потребуется время

$$t_{10} = N_4 t_9 = 80 \cdot 10 = 13,33 \text{ ч.}$$

Суммарное время за сборку 9920 изделий

$$t_{11} = t_8 + t_{10} = 22,04 + 13,33 = 35,37 \text{ ч,}$$

куда включено оперативное время и время простоев. Отношение времени простоев к общему времени сократится с $t_7 = 86,3$ до

$$t_{12} = t_{10}/t_{11} = 13,33/35 = 38 \text{ \%}.$$

Производительность за 10 000 циклов

$$P_3 = N_3/t_{11} = 9920/35,37 = 280,46 \text{ изделий/ч,}$$

$$\text{КПД} = P_3/P_1 = 280,46/450 = 0,62 = 62 \text{ \%}.$$

КПД = 62 % не очень высок, но он значительно выше, чем 13,7 %, полученные без контроля качества компонентов. Контроль качества компонентов может легко решить вопрос об успехе или провале во внедрении автоматической сборочной установки. Следующий пример служит для повторения вопросов, раскрытых в двух предыдущих разделах.

Пример 4.2. Автоматическая сборка при сбоях в установке и заклинивании деталей. Автоматическая сборочная линия состоит из тридцати последовательно соединенных позиций ($n = 30$). Линия оснащена качающимся штанговым конвейером, с тактом $t_1 = 4$ с и высотой $t_2 = 20$ с. Каждая позиция будет работать без сбоев с вероятностью $p_1 = 0,999$, если в бункере находятся только качественные компоненты. Любой некачественный компонент вызовет заклинивание на позиции, что приведет к перебою в работе всей линии, поскольку межпозиционные накопители не предусмотрены. На устранение сбоя или заедания требуется время $t_3 = 10$ мин.

а) *Отсутствие сбоев.* Какова идеальная производительность линии при полном отсутствии сбоев? Каково время производства?

Решение. Продолжительность цикла

$$T_{\text{ц}} = t_1 + t_2 = 4 + 20 = 24 \text{ с.}$$

Идеальная производительность

$$P_1 = 3600/24 = 150 \text{ изделий/ч.}$$

Время производства

$$T_{\text{п}} = 30 \cdot 24 = 720 \text{ с} = 12 \text{ мин.}$$

б) *Сбои сборочных устройств на позициях.* При идеальном качестве компонентов какова производительность линии, если сбои сборочных устройств имеют место? Какую долю составляет время простоев? Каково время производства?

Решение. Вероятность безотказной работы $p_1 = 0,999^{30} = 0,9704$; вероятность отказа $p_2 = 1 - 0,9704 = 0,0296$; время, затрачиваемое на сборку 9704 изделий ($N_1 = 9704$), t_3 , при суммарном времени безотказной работы t_4 и времени на устранение отказов t_5

$$t_3 = t_4 + t_5 = N_1 T_{\text{ц}}/3600 + N_2 t_3/60 = 9704 \cdot 24/3600 + \\ + 296 \cdot 10/60 = 64,69 + 49,33 = 114,02 \text{ ч.}$$

Производительность

$$P = N_1/t_3 = 9704/114,02 = 95 \text{ изделий/ч.}$$

Доля простоев

$$D = t_5/t_3 = 49,33/114,02 = 0,43 = 43 \text{ \%}.$$

Время производства

$$T_{\text{п}} = 60n/P = 60 \cdot 30/95 = 18,95 \text{ мин.}$$

в) *Заклинивание из-за некачественных деталей.* Предположим, что 0,5 % количества деталей в партии — брак. Как это повлияет на производительность?

Решение. Вероятность того, что данная позиция не заклинится на данном цикле,

$$p_3 = p_1 (1 - 0,005) = 0,999 (1 - 0,005) = 0,994.$$

Вероятность того, что цикл не прервется

$$p_4 = p_3^n = 0,994^{30} = 0,8349;$$

вероятность того, что цикл прервется,

$$p_5 = 1 - 0,8349 = 0,1651 (N_2 = 1651);$$

время, затрачиваемое на сборку $N_1 = 8349$ изделий,

$$t'_3 = t'_4 + t'_5 = N_1 T_{\text{ц}}/3600 + N_2 t_3/60 = 8349 \cdot 24/3600 + \\ + 1651 \cdot 10/60 = 55,55 + 215,16 = 330,82 \text{ ч.}$$

Производительность

$$P = N_1/t'_3 = 8349/330,82 = 25,24 \text{ изделий/ч.}$$

Доля простоев

$$D = t'_3/t'_3 = 275,16/330,82 = 0,83 = 83 \, \%.$$

$$\text{КПД} = P/P_1 = 0,17 = 17 \, \%.$$

Время производства

$$T_{\Pi} = 60n/P = 60 \cdot 30/25,24 = 1,19 \text{ ч.}$$

Таким образом, всего 0,5 % некачественных деталей, как в нашем случае, сокращает производительность с 95 изделий/ч до 25,24 изделий/ч, увеличивает время простоев с 43 % до 83 % и время производства приблизительно с 20 мин до более чем 1 ч.

Качество и единообразие компонентов настолько важны для успеха автоматизации сборки, что некоторые фирмы переплачивают в несколько раз по сравнению с обычной ценой, приобретая винты, гайки и болты наивысшего качества для этих целей. Простой из-за заклиниваний обходится так дорого, что дополнительные затраты на качество компонентов легко окупаются.

Сборка с использованием некачественных компонентов. Одним из способов решения проблемы качества компонентов является автоматизация процесса контроля с тем, чтобы дорогая сборочная операция не выполнялась впустую. Это хорошая стратегия, но по результатам, полученным при решении примера 4.2, можно предположить, что, поскольку компоненты уже попали в сборочную машину, лучшей стратегией будет умышленно дособрать изделие с некачественным компонентом, а не останавливать линию. В некоторых случаях, например при сборке электронных печатных плат, практичнее отбирать некачественные компоненты уже после окончания сборки, несмотря на то, что это потребует дополнительных затрат на частичную разборку. Достоинства этой стратегии проиллюстрированы в примере 4.3.

Пример 4.3. Экономия при сборке с использованием некачественных компонентов. Автоматическая сборочная станция, состоящая из $n = 20$ позиций, работает в две смены ($k = 2$), включая обслуживание, $t = 350$ дней в год. Предупредительный ремонт и обслуживание осуществляются в третью смену и во время работы. Линия останавливается по требованию для ремонта машины или для замены некачественных компонентов. 1 % компонентов имеет дефекты, а применяющаяся стратегия заключается в том, чтобы постоянно выявлять и удалять дефектные компоненты до окончания сборки, хотя это и требует остановки линии каждый раз в среднем на время $t_1 = 15$ мин. Стоимость автоматической сборочной установки в год $C = 490\,000$ долл. и обеспечивает идеальную продолжительность цикла $T_{\Pi} = 10$ с.

Предлагаемая стратегия состоит в том, что линия работает без остановки для удаления некачественных компонентов, хотя это требует времени $t_2 = 0,6$ ч на некачественную сборку при стоимости $C_1 = 20$ долл./ч. Скомпенсирует ли экономия времени затраты на дорогое исправление изделий?

Решение. Анализ состоит в сравнении стоимости простоев по стратегии C_A со стоимостью исправлений по стратегии C_B :

$$C_A = [t_1/(60k)] [C/t] = [15/(60 \cdot 16)] (490\,000/365) = 21\,875 \text{ долл.},$$

$$C_B = t_2 C_1 = 0,6 \cdot 20 = 12 \text{ долл.}$$

Следовательно, лучшей альтернативой является преднамеренная сборка с использованием некачественных компонентов, а не остановка линии.

Обратите внимание, что в примере 4.3 понятия производительности и продолжительности цикла не использовались при анализе. Самым дешевым вариантом в данном примере была сборка с использованием некачественных компонентов с последующим исправлением, однако необходимо рассмотреть и другие альтернативы. Время простоя сборочной установки при извлечении некачественных деталей в действительности не теряется, если его можно компенсировать работой в третью смену. Эквивалентная годовая стоимость установки при этом существенно не изменится, поскольку большая часть этой стоимости состоит из возврата средств, первоначально вложенных в установку. Двумя другими альтернативами являются автоматическая предварительная отсортировка некачественных компонентов до того, как они попадут на установку, и улучшение исходного качества компонентов во время их изготовления.

В примере 4.3 условия были подобраны таким образом, чтобы сделать преднамеренную сборку с использованием некачественных компонентов жизнеспособной альтернативой. Однако обычно такая стратегия не является лучшей. Если есть возможность использовать установку дополнительно, хотя бы в ночную смену, обычно выгоднее компенсировать простои за счет этого, чем нести убытки из-за преднамеренной сборки с использованием некачественных компонентов.

Читателю, которому условия примера 4.3 покажутся чересчур искусственными, следует знать, что бывают случаи, когда брак целесообразно пропускать.

Например, при пошиве верхнего платья дефекты материала часто обнаруживаются персоналом раскройного цеха во время раскладки материала, выполняемой перед раскроем. Раскраивается сразу стопа до 250 слоев и дефекты пропускаются, чтобы избежать нарушения процесса раскладки или повреждения других частей кроя, которые будут изготавливаться из слоя с дефектами. Обычно рассчитывают на то, что дефектные части раскроя будут обнаружены и изъяты в пошивочном цехе до наметки или далее при выходном контроле, когда исправление брака приводит к неприятным последствиям. Любая из этих альтернатив может оказаться предпочтительной по сравнению с удалением дефектного материала в раскройном цехе.

4.3. БУФЕРНЫЙ СКЛАД

Если вероятностные показатели операций на отдельных позициях автоматической сборочной линии характеризуются большим разбросом, необходимо обеспечить некоторую степень независимости между позициями с тем, чтобы обеспечить хотя бы некоторую производительность. Это достигается помещением между позициями буферных складов. Такие системы обычно заставляют применять несинхронные средства транспортирования материалов,

поскольку жесткие промежуточные средства транспортирования не приспособлены к изменениям производственного времени на позициях. Несинхронные средства транспортирования характеризуются выражением «подал и оставил».

Чтобы осознать преимущества буферных складов между позициями, рассмотрим линию, у которой идеальное время цикла 1 мин. Если каждая позиция на линии отказывает один раз в 60 циклов, а на восстановление каждый раз требуется 1 ч, каждая станция будет готова к работе приблизительно 50 % времени. Говоря предположительно, если бы удалось синхронизировать отказы на всех позициях, линия теоретически имела бы производительность, равную половине от 60, т. е. 30 изделий/ч. Но природа отказов такова, что время их возникновения случайно. Это обстоятельство сильно снижает производительность линии, поскольку, если хоть одна позиция в линии отказала, отказала и вся линия, как было показано при анализе работы тактовых сборочных установок. При ситуации, описанной выше, когда каждая позиция готова к работе только 50 % времени, готовность всей тактовой линии $0,5^n$, где n равно числу позиций в линии. Такая линия всего из 5 позиций будет простаивать 97 % времени! Разность между этой долей простоев и гипотетической долей готовности, полученной при условии синхронизации отказов:

$$97 - 50 = 47 \%,$$

представляет собой потерю производительности из-за взаимной зависимости работы позиций. Зависимость называется «блокировкой», когда позиция i не может передать свое изделие на позицию $i + 1$, и «голоданием», когда позиция i не может получить изделие из позиции $i - 1$. В любом случае позиция i простаивает, хотя она исправна и могла бы работать при наличии изделия.

Встройка буферного склада между позициями позволяет каждой позиции создать «запас». Сначала начинает работать первая позиция, которая до начала работы на второй позиции заполняет полуфабрикатами приблизительно половину емкости своего склада. Вторая позиция начинает работать, забирая изделие из склада, заполняемого первой позицией, в то время как первая продолжает работать. Последующие позиции включаются в работу, когда склад предыдущей позиции заполнен почти наполовину, при условии, что все позиции имеют до некоторой степени аналогичные эксплуатационные характеристики и продолжительность простоев. Когда все склады заполнены, линия, благодаря этому, может работать при некоторой способности к восстановлению. Если буферные склады достаточно велики, чтобы предупредить блокировку и голодание, производительность описанной линии может быть доведена до 30 изделий в минуту.

Пример 4.4. Работа сборочной линии с буферным запасом. В примере 4.2 показано влияние отказов отдельных позиций линии и заклинивания деталей на автоматическую сборку. Предположим, что та же сборочная операция выполняется на линии из $n = 30$ позиций, оснащенной буферными складами, обеспе-

чивающей тот же цикл сборки $T_{\Sigma} = 20$ с (время такта $t_1 = 4$ с — не упоминается, поскольку в данном случае каждый склад образует запас изделий). Рассчитайте повышение производительности линии.

Решение. Вероятность того, что данная позиция не заклинит, $p_3 = 0,994$ (см. пример 4.2, в); время на сборку $N_1 = 994$ изделий

$$t_3 = t_4 + t_5 = N_1 T_{\Sigma} / 3600 + (1 - p_3) t_3 / 60 = 994 \cdot 20 / 3600 + \\ + 6 \cdot 10 / 60 = 5,52 + 1 = 6,52 \text{ ч.}$$

Производительность (с резервным запасом)

$$P_1 = N_1 / t_3 = 994 / 6,52 = 152,4 \text{ изделий/ч.}$$

Производительность (без резервных запасов из решения примера 4.2, в)

$$P_2 = 25,24 \text{ изделия/ч.}$$

Повышение производительности

$$P_1 / P_2 = 152,4 / 25,24 = 6,04.$$

Пример 4.4 показывает, как введение соответствующего буферного запаса может шестикратно повысить производительность линии. Эта информация отрезвляет, поскольку она бросает тень на типичные сборочные автоматические линии тактового типа, являющиеся сгустком автоматизации. Производство на автоматической линии, оснащенной буферными складами между всеми позициями, вырождается в серийное, типичное для предприятия до проведения автоматизации.

Рассмотрим недостатки, связанные с введением буферных складов между позициями. В примере 4.4 были опущены расчеты времени производства, которое сильно возрастает с введением буферных складов. Характер увеличения этого времени зависит от количества полуфабрикатов в складах, как это будет показано в примере 4.5.

Пример 4.5. Влияние буферных запасов на время производства. Сборочная линия из тридцати позиций ($n = 30$) с буферным запасом между позициями, составляющим в среднем $N' = 50$ изделий на каждую позицию, имеет производительность $P = 152,4$ изделия/ч (аналогично примеру 4.4). Рассчитайте время производства.

Решение. Время производства включает в себя время нахождения каждого изделия в буферных складах между позициями. Среднее время нахождения изделия на складе T_1 равно *произведению* среднего времени цикла для позиции (включая простои) T_{Σ} на среднее количество изделий в складе N . Поэтому для $N = 50$

$$T_1 = T_{\Sigma} N = N / P = 50 / 152,4 = 0,328 \text{ ч.}$$

Поскольку при $n = 30$ в сборочной линии будет $n - 1$ буферных складов, т. е. $n_1 = 29$ складов, суммарное время пролеживания для изделий в сборочной линии с буферными запасами

$$T'_1 = T_1 n_1 = 0,328 \cdot 29 = 9,51 \text{ ч.}$$

Суммарное время производства

$$T_{\Sigma} = n / P + T'_1 = 30 / 152,4 + 9,51 = 0,2 + 9,51 = 9,71 \text{ ч.}$$

Альтернативное решение. Другой способ определения времени производства для сборочной линии с буферными запасами состоит в том, что определяется количество позиций n' , через которое изделие должно проходить по линии при интенсивности $(1/152,4)$ ч на позицию:

$$n' = n + n_1 N' = 30 + (29 \cdot 50) = 1480.$$

Время производства

$$T_{\Sigma} = n + n_1 N' / P = 30 + (29 \cdot 50) / 152,4 = 9,71 \text{ ч.}$$

Время производства, полученное в примере 4.4 и равное 9,71 ч, можно сравнить с временем производства, определенным в примере 4.2, в, где рассматриваются отказы и заклинивания. Это время составляет приблизительно 1 ч. Поэтому, хотя использование буферных складов увеличивает производительность линии шестикратно (см. пример 4.3), оно же увеличивает время производства почти в 10 раз (см. пример 4.4). Это заключение базируется на допущении о том, что среднего буферного запаса в 50 изделий будет достаточно для того, чтобы предотвратить блокирование или голодание позиций в сборочной линии с буферными запасами.

Кроме значительного роста времени производства, буферные запасы увеличивают сложность манипулирования изделиями и их транспортирования. При обычном серийном производстве, которое можно назвать вырожденным состоянием сборочной линии с буферными запасами, вся работа по позиционированию, известная нам из гл. 2, пропадает впустую на каждой позиции, поскольку каждый раз собранные полуфабрикаты ссылаются в приемный бункер следующей позиции.

Очевидно, что главная проблема организации штучного производства или сборки заключается в правильном решении дилеммы, состоящей в том, что стремление достичь наибольшей производительности противоречит попыткам обеспечить минимальную стоимость манипулирования и минимальное время производства. Однако именно здесь на сцену выступает промышленный робот. С одной стороны, он способен сохранять ориентацию и положение деталей, а с другой — может в зависимости от обстоятельств помещать продукцию на склад или подавать ее дальше по технологической цепочке. Это лишь немного из тех преимуществ, которые обеспечиваются гибкими промышленными роботами, о чем будет подробнее рассказано позже. Но сначала мы должны завершить ознакомление с автоматическими процессами изучением станков с программным управлением, являющихся предшественниками промышленных роботов, что и сделано в гл. 5.

В ы в о ды

Как только детали правильно сориентированы и поданы на установку, они готовы к автоматической обработке или сборке. Такие операции, как завинчивание винтов и закручивание гаек, могут быть полностью автоматизированы. Сборочные машины распределяются по позициям, устанавливаемым в линию или по кругу. Любая компоновка может оказаться неэффективной из-за простоев системы, вызываемых отказами или заклиниванием деталей. Сокращение КПД и производительности тем более существенно, чем больше число позиций.

Проблемы, связанные с отказами и заклиниванием некачественных компонентов, могут быть частично решены при помощи введения буферных складов на позициях автоматической линии.

Эти склады могут обеспечить большое повышение производительности линии, но за счет значительного увеличения времени производства. Одновременно установка складов может создать трудности в сохранении ориентации и в позиционировании деталей при автоматической сборке.

Важность обеспечения качества компонентов для сборки возрастает с внедрением автоматизации. Главной причиной этого является возможность заклинивания из-за низкого качества компонентов, что приводит к остановке всей линии. Эта проблема может быть настолько важной, что в некоторых случаях становится более выгодным собирать изделие, используя вместе с качественными некачественные компоненты, удаляя последние по окончании сборочного процесса, чем останавливать линию.

У п р а ж н е н и я и в о п р о с ы

4.1. Продолжительность цикла сборки на одной позиции 12-позиционной автоматической сборочной линии составляет 18 с, включая время такта. Рассчитайте идеальную производительность линии и время производства.

4.2. 8-позиционная автоматическая линия оснащена буферными складами между позициями с емкостью 50 изделий каждый. Все позиции сравнительно одинаковы по эксплуатационным характеристикам и простоям. Сколько изделий можно рекомендовать иметь на складах при нормальной работе линии?

4.3. Какую последовательность запуска можно рекомендовать при запуске и загрузке линии, описанной в упражнении 4.2?

4.4. Рассчитайте среднее время производства для линии, описанной в упражнении 4.2.

4.5. Шестипозиционная сборочная установка не имеет буферных складов. Интенсивность отказов составляет 2 % на каждую позицию. Среднее время на восстановление и перезапуск установки составляет 10 мин. При идеальном времени цикла 10 с, включая продолжительность такта, определите, какова действительная производительность. Рассчитайте коэффициент использования.

4.6. В упражнении 4.5 предположим, что интенсивность отказов может быть снижена наполовину за счет проведения работ по повышению качества компонентов, включающих более жесткую приемку покупных изделий, отбор поставщиков и введение более жестких допусков на покупные компоненты и компоненты собственного производства. Как это повлияет на коэффициент использования сборочной установки?

4.7. Рассчитайте и сравните идеальное и действительное время производства для сборочной машины, описанной в упражнении 4.5.

4.8. 6-позиционная карусельная тактовая установка приводится мальтийским механизмом с кривошипом, вращающимся с частотой 18 об/мин. Каково время выпуска для этой установки и время производства?

4.9. Предположим, что автоматическая линия, состоящая из 50 позиций без буферных складов и резервов, планируется к внедрению как основа автоматизации производства, где Вы работаете в качестве инженера по автоматизации. Для того чтобы обеспечить выпуск 2000 изделий за смену 8 ч, было принято время цикла 12 с. Управляющий заводом очень беспокоится об этом новом проекте и планирует пригласить руководство корпорации, чтобы продемонстрировать плоды автоматизации, как только Вы запустите линию. Разделяете ли Вы оптимизм управляющего в отношении этого проекта? Реалистично ли ожидать выпуска 2000 изделий в смену? Согласуется ли этот объем выпуска с заданным временем цикла? Если все обслуживание проводится во вторую смену и теоретически линия может работать всю первую смену, какова будет идеальная производительность этой линии? Какая максимальная интенсивность отказов на цикл может считаться допустимой, если среднее время восстановления и перезапуска линии составляет 6 мин?

5. ЧИСЛОВОЕ ПРОГРАММНОЕ УПРАВЛЕНИЕ, АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ И ИЗГОТОВЛЕНИЕ С ПОМОЩЬЮ ЭВМ

Предшественниками промышленных роботов являются станки с числовым программным управлением (ЧПУ), представляющие собой автоматизированный вариант обычных станков. Обычные фрезерные, токарные, сверлильные, расточные и другие станки управляются вручную маховиками и рукоятками (рис. 5.1). Режущий инструмент удерживается в нужном положении станком, что обеспечивает точное резание заготовки, обычно изготовленной из металла. Обычные станки еще применяются при очень маленьких размерах партий или в инструментальных цехах, где отдельные специальные инструменты изготавливаются высококвалифицированными рабочими-инструментальщиками, с тем, чтобы использоваться затем на станках в серийном производстве.

Чтобы автоматизировать последовательность операций, выполняемых на обычных станках при обработке деталей, инженеры по автоматизации сначала использовали кулачки и механические связи. Этот прием хорошо зарекомендовал себя в отношении обычных станков, производительность которых возрастает при автоматизации управления. Одновременно сокращается потребность в рабочей силе. Еще одним преимуществом автоматизации явилось повышение стабильности выполнения операций, что положительно сказывается на качестве. Примером автоматизации на базе кулачков и механических связей является токарный копировальный полуавтомат (рис. 5.2), а своей вершины этот тип автоматизации достиг с разработкой пруткового автомата (рис. 5.3). Прутковые автоматы, которые иногда называют винторезными станками (по одному из видов изготавливаемой на них продукции), способны выполнять все виды обработки для деталей, изображенных на рис. 5.4, обеспечивая при этом высокую производительность. Это достигается одновременным выполнением шести операций, а шпиндель с деталью перемещается из позиции в позицию. Этот тип станков был последним словом в автоматизации 40-х годов.

Несмотря на наличие прутковых автоматов и других станков с последовательностью операций, обеспечиваемой механическими средствами, некоторых проблем станки-автоматы 40-х годов не решали. Серийность некоторых изделий была слишком мала,

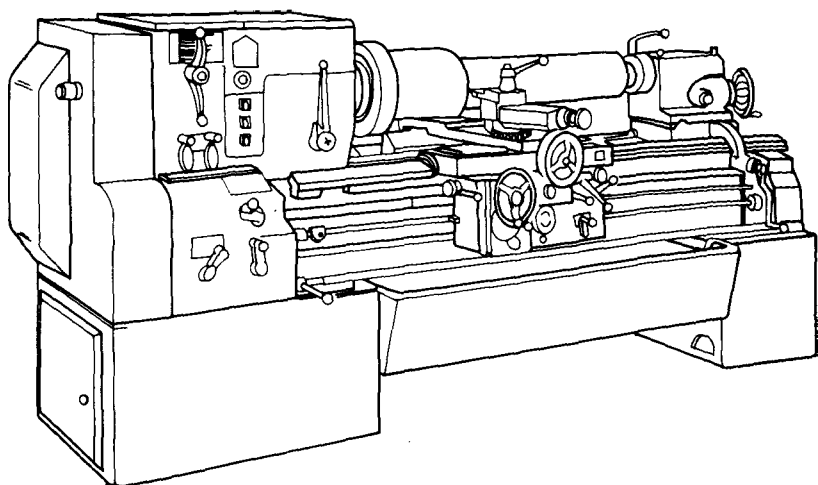


Рис. 5.1
Обычный универсальный токарно-винторезный станок

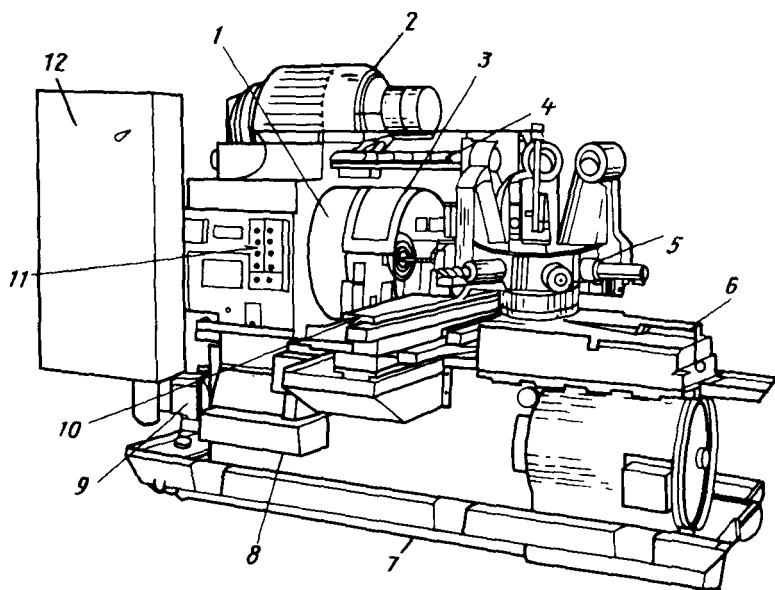


Рис. 5.2
Токарный копировальный полуавтомат:

1 — передняя бабка; 2 — приводной двигатель; 3 — ограждение патрона; 4 — верхняя направляющая цапфа; 5 — револьверная головка; 6 — салазки револьверной головки; 7 — основание; 8 — распределительная коробка; 9 — насос системы охлаждения; 10 — передний поперечный суппорт; 11 — кнопочная панель; 12 — шкаф управления

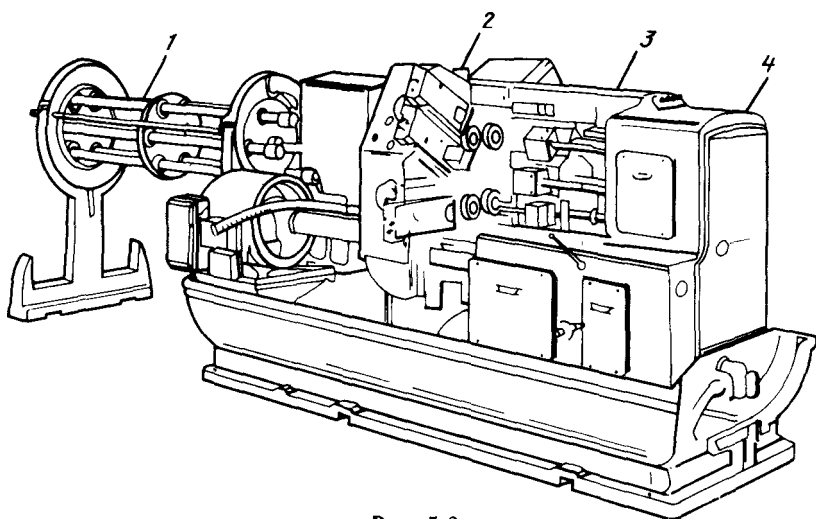


Рис. 5.3

Многошпиндельный прутковый автомат:

1 — прутковый магазин; 2 — шпиндельный блок; 3 — инструментальная наладка;
4 — коробка передач

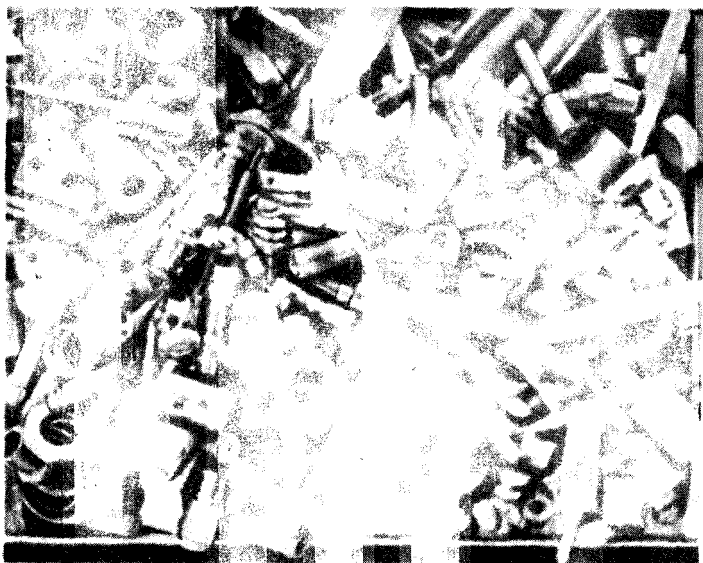


Рис. 5.4

Детали, изготавливаемые на прутковых автоматах

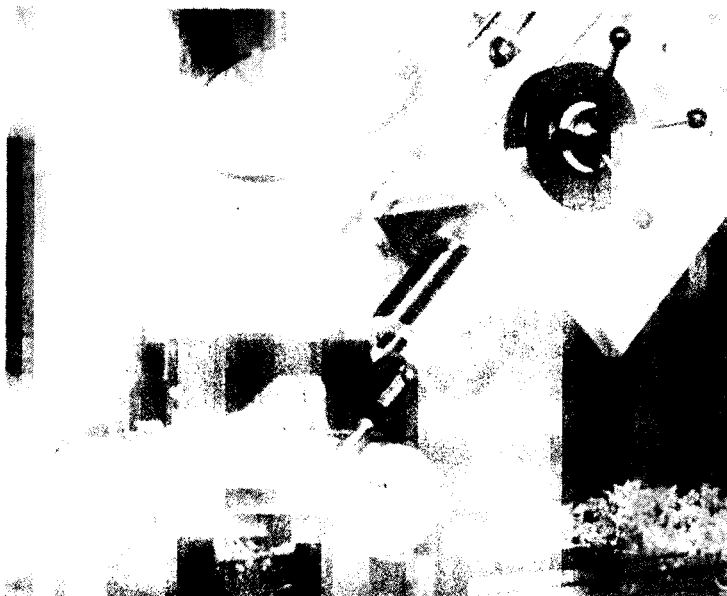


Рис. 5.5

Обработка цапфы для космической лаборатории НАСА (Национальное управление по аэронавтике и исследованию космического пространства) на фрезерно-расточном станке

чтобы оправдать капиталовложения в обеспечение последовательности операций механическими средствами. Единственной возможностью решить эту проблему было использование опытных токарей, фрезеровщиков и других рабочих-станочников, чтобы управлять станками вручную, несмотря на потери в производительности и качестве. Изготовление деталей для авиационной и космической промышленности начала 50-х годов потребовало привлечения в высшей степени квалифицированных станочников для изготовления больших и поразительно сложных алюминиевых деталей самолетов и космических кораблей.

Кроме других навыков, от станочника, фрезеровавшего громадные детали шасси или лонжероны крыльев из алюминиевых отливок, требовалось умение читать чертежи. Операции по обработке детали, подобной изображенной на рис. 5.5, были настолько разнообразны, что их трудно было разбить на ряд технологических переходов и приходилось рассчитывать только на опыт станочника, способного разработать для себя технологию прямо с чертежа. От такого станочника высшей квалификации требовался также навык обращения с прецизионным мерительным инструментом. В процессе изготовления, зависящем от опыта оператора, необходимо осуществлять частые проверки, чтобы удостовериться в достижении требуемого уровня точности. Труд-

ность выполнения таких проверок возрастает из-за малой серийности сложных деталей.

Все это и привело в 1949 г. к организации проекта, финансируемого ВВС США, по которому Массачусетскому технологическому институту поручалась разработка опытного образца станка для обработки лонжеронов крыльев и обшивки военных самолетов. Основная идея состояла в том, что исполнительные перемещения станка должны управляться числовым кодом, воспринимаемым станком. Преимущество должно было заключаться в возможности хранения сложной последовательности операций, записанной на соответствующем носителе информации при обеспечении повторного воспроизведения, если возникала необходимость обработки той же детали через некоторое время.

Некоторые авторы связывают возникновение программного управления с изобретением ткацких станков и других машин, управляемых от перфокарт, появившихся столетие назад. Действительно, машины, работающие по этому принципу, существовали и раньше, однако они не назывались станками с ЧПУ и принадлежали к другому этапу эволюции, чем металлорежущие станки с ЧПУ, разработанные в 50-х годах для аэрокосмической промышленности.

5.1. ПРЕИМУЩЕСТВА ЧИСЛОВОГО ПРОГРАММНОГО УПРАВЛЕНИЯ

В предпосланном введении аэрокосмическая промышленность показана как колыбель современных станков с ЧПУ. Почему, однако, аэрокосмическая? Ответ на этот вопрос проливает свет на принципы и цели использования станков с ЧПУ. Аэрокосмическая промышленность является идеальной областью для применения станков с ЧПУ практически со всех точек зрения и по всем мыслимым причинам. Ниже рассмотрены преимущества использования станков с ЧПУ в аэрокосмической промышленности, идеальной области применения по сравнению с другими отраслями, которые также могут быть отличными кандидатами для этого.

Гибкость. Многие изделия изготавливаются в количестве 100 000 или даже 1 000 000 штук одной модели, но это не относится к самолетам или космическим кораблям. Значительно меньшая серийность производства в аэрокосмической промышленности требует такого типа автоматизации, которая могла бы быть легко перестроена при переводе производства с одной модели на другую и даже с одного изделия на другое без замены оборудования. Удовлетворение этих требований при негибком типе автоматизации, существовавшем до 1950 г., обошлось бы слишком дорого. Числовое программное управление благодаря своей гибкости дало ответ на этот вопрос и стало применяться в других отраслях промышленности после того, как аэрокосмическая промышленность проложила себе дорогу.

С другой стороны, гибкость, являющаяся преимуществом станков с ЧПУ, реализуется в сокращении времени наладки. Во время наладки персонал, станки и оснастка заняты, как и в операционное время, с той разницей, что во время наладки продукция не производится. Во время первого запуска в производство данной продукции время наладки станка с ЧПУ сравнимо с временем наладки оборудования с «жесткой» автоматизацией, поскольку требуется запрограммировать и проверить обработку. Зато когда в дальнейшем потребуется снова вернуться к изготовлению этой продукции, время переналадки станка с ЧПУ значительно уменьшается.

Возможность обработки сложных деталей. При «жесткой» автоматизации осуществление длинных и сложных последовательностей обработки не представляется возможным. Этот тип автоматизации предполагает использование специальных станков со сложными кулачками и механическими связями для осуществления немногих (около шести), часто повторяющихся операций. В противоположность этому станки с ЧПУ очень универсальны, и на них так же легко запрограммировать последовательность из шести операций, как и из 66 или даже 666 операций. После окончания цикла та же программа может быть повторена на другой заготовке или может быть заложена другая программа, по которой через несколько минут тот же станок начнет обрабатывать совершенно другую деталь, делая сотни операций, совершенно не похожих на предыдущие. Это как раз то, что требуется для аэрокосмической промышленности при изготовлении сложных лонжеронов крыльев и шасси. Другие отрасли последовали за аэрокосмической промышленностью и не в последнюю очередь само станкостроение.

Обработка крупных изделий. Специальные станки-автоматы обычно применяются для выполнения нескольких операций, после чего деталь передается на другой станок для продолжения процесса обработки. То же самое наблюдается и при использовании станков с ЧПУ, разница же состоит в количестве станков по технологическому маршруту. Вместо передачи детали на другой станок на станке с ЧПУ происходит замена инструмента и затем продолжение обработки. Когда детали большие и неудобные для транспортирования, как, например, детали самолета, это преимущество станка с ЧПУ может сократить стоимость обработки.

Сокращение стоимости инструментальной и крепежной оснастки. Возможность оставить деталь на станке для продолжения обработки тем же самым или другим инструментом создает еще одно преимущество станков с ЧПУ: сокращение стоимости инструментальной и крепежной оснастки. Это может быть даже важнее, чем простота обработки крупногабаритных и неуклюжих деталей. Гибкость и универсальность станков с ЧПУ позволяет производить обработку деталей под разными углами, в различных режимах, и с помощью различных инструментов для осуществле-

ния большого количества операций обработки без перемещения деталей в другие крепежные приспособления.

Качество. Выполнение многих операций при одном закреплении деталей позволяет достигнуть более высоких и стабильных качественных показателей, чем при обработке на нескольких специальных станках, что требует нескольких перезакреплений деталей. К этому нужно добавить освобождение от нестабильности, вносимой людьми и разнообразием процессов. Это также приводит к повышению качества изделий, обрабатываемых на станках с ЧПУ.

Существовало мнение, что дополнительно к указанным преимуществам станки с ЧПУ снижают требования к уровню квалификации рабочих-станочников. Действительно, управление правильно запрограммированными станками с ЧПУ требует меньшей квалификации станочников, чем при выполнении тех же операций на обычных станках. Но для управления и обслуживания станков с ЧПУ требуются совсем другие навыки. При внедрении станков с ЧПУ возникает ситуация, аналогичная наблюдаемой при внедрении других средств гибкой автоматизации: вместо навыков по управлению станками и навыков ручной работы требуются навыки обращения с вычислительной техникой.

Говоря об управлении станками, было бы неразумно предполагать, что применение станков с ЧПУ снизит требования к квалификации операторов и одновременно оплату их работы. Автор с удивлением обнаружил, что на авиационных заводах в 60-х годах операторы станков с ЧПУ числились среди самых опытных и высокооплачиваемых станочников. Причина этой аномалии кроется скорее в традиционном приоритете, отдаваемом станкам с ЧПУ, чем в действительной необходимости использования опытных станочников. С другой стороны, степень ответственности может быть фактором, влияющим на выбор квалификации станочника для станка с ЧПУ. Хотя на таком станке объем работы, требующей физической силы, сокращается, стоимость станка с ЧПУ и обрабатываемой детали на порядок выше, чем для обычного станка.

5.2. МАШИННЫЕ КОДЫ

Коды, воспринимаемые станками, более эффективно выражаются в системах, основанных на двоичном счислении. На рис. 5.6 для сравнения приведены перфолента с информацией, закодированной в системе, основанной на двоичном счислении, и обычная перфокарта с информацией, закодированной в десятичном счислении. В обоих случаях закодировано 80 цифр. Оба формата могут быть считаны и машиной, и человеком. Человек, конечно, должен знать двоично-десятичный код, чтобы прочесть информацию на перфоленте, и код Холлерита, чтобы прочесть информацию на перфокарте. Тенденцией является, однако, хранение программ на более компактных программоносителях, где

5.1. Привязка дорожек перфоленты к десятичным числам *

Номер дорожки перфоленты	1	2	3	4
Число в двоичной системе	2^0	2^1	2^2	2^3
Десятичный эквивалент	1	2	4	8

* Например, для представления десятичного числа 9 на дорожках с первой по четвертую будет пробито двоичное число: 1001.

5.2. Соотношение между числами в десятичной, двоично-десятичной и двоичной системах счисления

Десятичная система	Двоично-десятичная система	Двоичная система
0	0	0
1	1	1
2	10	10
3	11	11
4	100	100
5	101	101
6	110	110
7	111	111
8	1 000	1 000
9	1 001	1 001
10	10 000	1 010
11	10 001	1 011
12	10 010	1 100
13	10 011	1 101
14	10 100	1 110
15	10 101	1 111
16	10 110	10 000
17	10 111	10 001
18	11 000	10 010
19	11 001	10 011
20	100 000	10 100

ного числа. В табл. 5.2 приведены для сравнения десятичные цифры и их написание в двоично-десятичной и двоичной системах счисления. Обратите внимание, что написание десятичных цифр от 0 до 9 включительно в двоичном и двоично-десятичном счислениях совпадают. При написании десятичного числа 10 в двоично-десятичном коде добавляется еще один двоичный знак (как и в десятичном), и оно начинает считаться снова, начиная с нижнего порядка. Чисто двоичные числа продолжают иметь четыре знака, пока все возможные комбинации не будут исчерпаны. Таким образом, двоичная система более эффективна, но труднее для прочтения. Если читатель усвоил описываемый метод, он уже может написать двоично-десятичный эквивалент любого десятичного числа,

Десятичное число 1439 =
 = двоично-десятичному числу 1010000111001 (3 нуля в начале опущены)
 Написание:

Двоично-десятичное число	0001	0100	0011	1001
Десятичное число	1	4	3	9

Рис. 5.8
 Написание числа 1439 в двоично-десятичной системе

а чисто двоичный — нет. Попробуйте проделать это для десятичного числа 1439. Написание числа в двоично-десятичной системе проиллюстрировано на рис. 5.8.

Двоично-десятичные числа распределены по знакам десятичного числа. Чисто двоичное написание десятичного числа 1439 будет короче, чем в двоично-десятичной системе, и в то же время сложнее для прочтения. Перевод больших десятичных чисел в чисто двоичную систему требует вычислений, изложение которых выходит за рамки этой книги и в действительности не нужно большинству инженеров по автоматизации и робототехнике. Для сравнения, однако, укажем, что чисто двоичное написание числа 1439 будет 1011001111. В этом можно удостовериться, найдя десятичный эквивалент каждой из двоичных единиц в двоичном числе и сложив эти эквиваленты, чтобы получить суммарное десятичное значение всего двоичного числа. Десятичный эквивалент знака в двоичном числе равен 2^{n-1} , где n — номер разряда в двоичном числе, считая справа. При счете справа в нашем примере двоичное число 1011001111 имеет двоичную единицу в 1, 2, 3, 4, 5, 8, 9 и 11 разрядах. Десятичный эквивалент затем можно подсчитать, как это сделано в табл. 5.3.

5.3. Вычисление десятичного эквивалента двоичного числа 1011001111

Разряд цифры	Двоичное число	Десятичный эквивалент	Вычисленный десятичный эквивалент
1	1	2^0	1
2	1	2^1	2
3	1	2^2	4
4	1	2^3	8
5	1	2^4	16
6	0	0	0
7	0	0	0
8	1	2^7	128
9	1	2^8	256
10	0	0	0
11	1	2^{10}	1024
Итого			1439

Буквы. Шестая и седьмая дорожки совместно с дорожками, предназначенными для кодирования цифр, используются для кодирования букв и специальных знаков. Рис. 5.7 показывает, что существует определенная система кодирования букв (английского) алфавита, хотя код ASCII несколько отличается от кода EIA244A. В коде ASCII алфавит кодируется пробиванием отверстий в шестой и седьмой дорожках с добавлением чисто двоичного кода номера буквы от 1 до 26 на дорожках с первой по пятую, поскольку в алфавите 26 букв. Код EIA244A следует двоично-десятичной системе счисления за счет разделения алфавита на три группы по десять букв. Группы кодируются следующим образом.

Буквы алфавита от A до I: отверстия в шестой и седьмой дорожках; буквы от J до R: отверстия в седьмой дорожке; буквы от S до Z: отверстия в шестой дорожке. Внутри группы цифры нумеруются с 1 до 9 в двоичной системе.

Внимательный читатель уже вероятно заметил некоторое несоответствие в описании процедуры кодирования букв в коде EIA244A. Ведь алфавит состоит из 26 букв, а не из 27, поэтому «три группы по девять букв» оставляют одну комбинацию неиспользованной. Сможете ли вы определить по рис. 5.7, какая комбинация пропущена и какому месту в алфавите это соответствует?

Проверка четности. Пятая дорожка в коде EIA244A и восьмая дорожка в коде ASCII зарезервированы для проверки надежности перфоратора и устройства считывания программы, установленного на станке. По установленному жесткому правилу число отверстий в каждом горизонтальном ряду всегда должно быть четным (в случае кода EIA244A) или нечетным (в случае кода ASCII). Это правило называется проверкой четности (или нечетности в зависимости от кода). Поскольку некоторым знакам двоично-десятичного кода соответствует четное количество отверстий, а некоторым — нечетное, дорожка четности используется для добавления в случае необходимости отверстия, обеспечивающего четность (или нечетность) каждого горизонтального ряда. Цель этой операции сейчас будет объяснена.

При набивке и считывании программ могут возникнуть четыре основных механических ошибки:

1. Перфоратор может пробить ненужное дополнительное отверстие.
2. Перфоратор может не пробить отверстие в нужном месте.
3. Считывающее устройство может ошибочно считать отверстие в том месте, где его в действительности нет.
4. Считывающее устройство может не считать имеющееся отверстие.

Каждая из ошибок возникает крайне редко, но на практике может произойти. Если одна из перечисленных ошибок произойдет, число отверстий в горизонтальном ряду не будет соответствовать установленному правилу четности. Таким образом, если

устройство одновременно подсчитывает количество пробитых или считанных отверстий в каждом ряду, нарушение четности будет немедленно выявлено как ошибка устройства, и оно остановится. Конечно, при двух ошибках в одном горизонтальном ряду перфоленты четность не будет нарушена и ошибка пройдет незамеченной. Две ошибки, конечно, не означают, что ошибок вообще нет, но устройство может воспринять это именно таким образом. Правило четности базируется на том, что вероятность появления двух ошибок в одном ряду настолько мала, что ею можно пренебречь.

Осталось объяснить только назначение восьмого ряда в коде ЕІА244А. Этот ряд предназначен для разделения блоков информации. Отверстие в восьмом ряду легко распознается, потому что имеет только одно значение — концевая строка. Это особенно удобно для визуальной проверки перфолент.

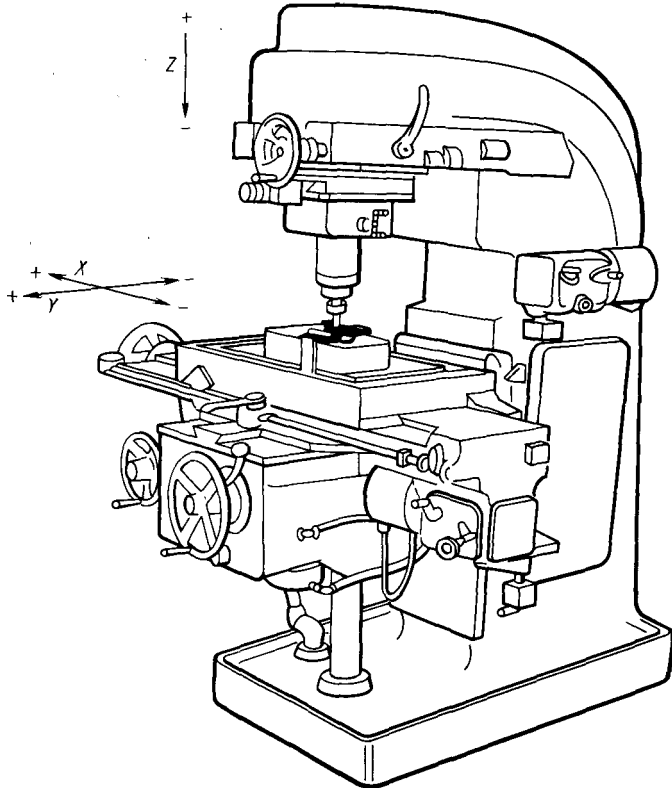


Рис. 5.9

Обозначение осей координат для вертикально-фрезерного станка с ЧПУ. Ось Z — вертикальное перемещение шпинделя. Оси X и Y — перемещение стола с закрепленной заготовкой. Например, перемещение стола влево заставляет фрезу обрабатывать деталь с правой стороны, производя резание по оси X в положительном направлении

5.3. ОСИ КООРДИНАТ

Основной функцией программного управления является выдача станку команд на перемещение инструмента или задание относительного перемещения инструмента и детали. Эта ключевая концепция прослеживается как в станках, описанных в данной главе, так и в роботах, о которых рассказано в гл. 6.

Движения в станках обычно описываются в декартовой системе координат, что делает перемещения по осям x , y и z основой программирования. Чтобы показать эти оси, мы обратимся к образу всех станков с ЧПУ — вертикально-фрезерному станку. На рис. 5.9 изображен такой станок с указанием осей x , y и z . Подобная схема используется и для других станков (рис. 5.10). Из этих рисунков видно, что назначение направления осей производится в соответствии с некоторыми принципами, правда, не без исключений. Обычно оси x и y определяют плоскость, перпендикулярную направлению внедрения инструмента в заготовку. Внедрение инструмента обычно совпадает с осью z , причем внедрению соответствует отрицательное направление оси, а отводу — положительное. В плоскости xz за большую ось (т. е. ось, по которой происходит наибольшее перемещение) обычно принимают ось x , а за малую — ось y . Все эти правила справедливы для вертикально-фрезерного станка, изображенного на рис. 5.9, но в некоторых примерах, показанных на рис. 5.10, нарушаются.

5.4. ПРОГРАММИРОВАНИЕ В ПРИРАЩЕНИЯХ И В АБСОЛЮТНОЙ СИСТЕМЕ КООРДИНАТ

Существуют два основных способа программирования перемещений для станков с ЧПУ и подобного оборудования. Первым способом является задание перемещения инструмента в новое положение, отличающееся на x , y и z от исходного положения в пространстве. Этот вид программирования обычно называют «программированием в приращениях». Другим способом является программирование в абсолютной системе координат, при котором имеется некоторое положение, являющееся исходным для задания всех команд на перемещение.

Большинство станков с ЧПУ могут работать и «в приращениях», и в абсолютной системе координат по желанию программиста. Такая возможность существовала не всегда, но она появилась благодаря постоянной работе по повышению гибкости автоматизации. Таким образом, можно выбрать тот или иной способ задания перемещений в зависимости от области применения обрабатываемого изделия, системы простановки размеров, примененной конструктором, или просто удобства программиста.

Для иллюстрации сказанного можно показать, как задается положение осей отверстий, которые должны быть просверлены в детали, изображенной на рис. 5.11.

Задание координат «в приращениях»: отверстие 1: $x = 2,000$, $y = 2,500$; отверстие 2: $x = 0$; $y = -0,500$; отверстие 3: $x = 1,250$; $y = 0$.

Задание в абсолютной системе координат: отверстие 1: $x = 2,000$; $y = 2,500$; отверстие 2: $x = 2,000$; $y = 2,000$; отверстие 3: $x = 3,250$; $y = 2,000$.

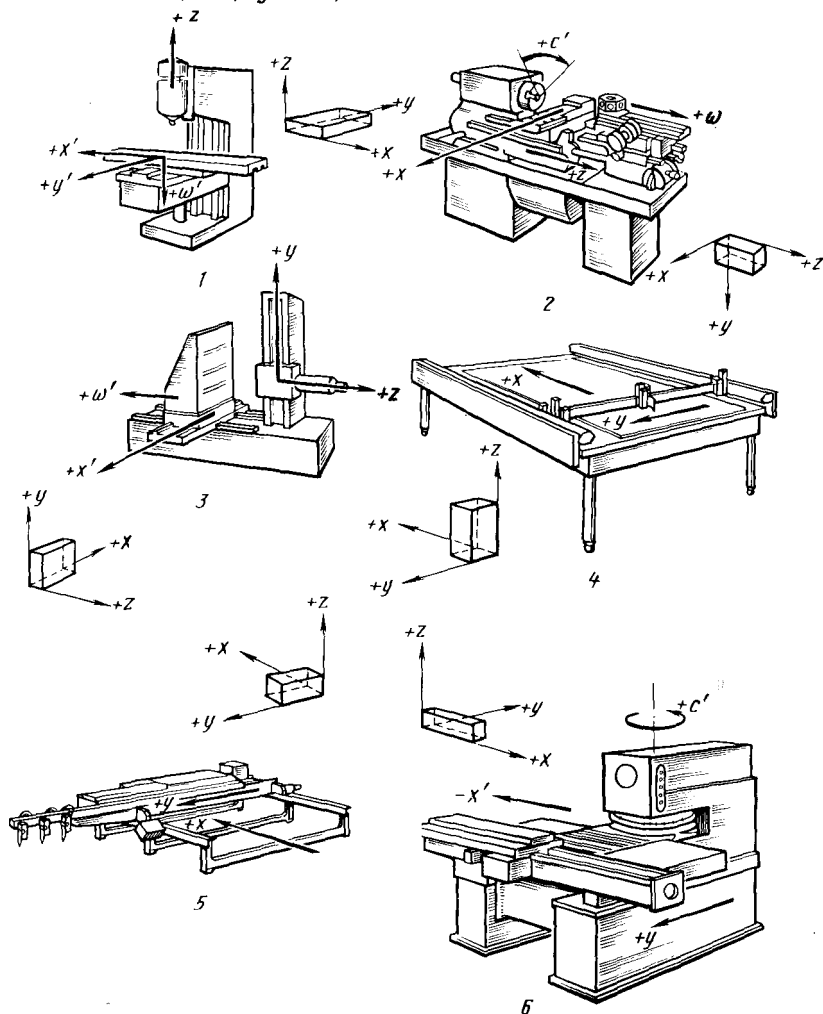
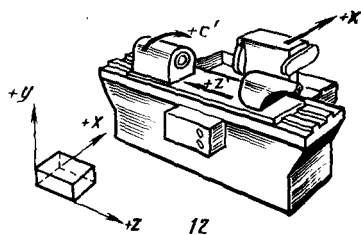
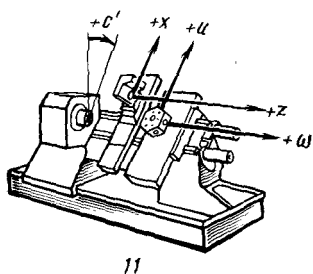
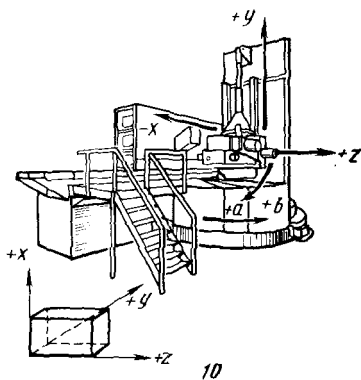
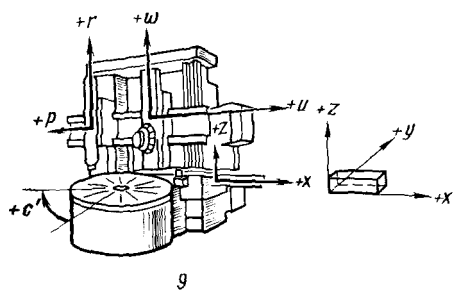
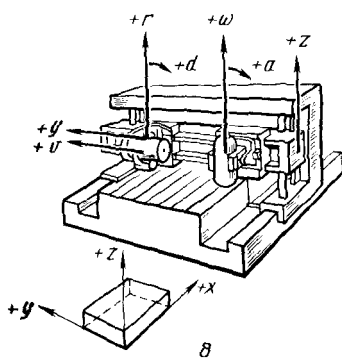
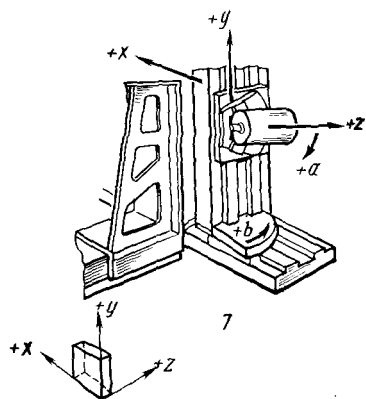


Рис. 5.10

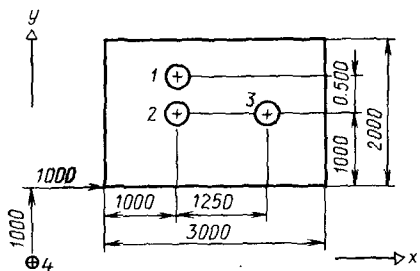
Оси координат для различных станков с ЧПУ:

1 — контрольно-фрезерный станок; сверляльный станок, координатно-расточной станок; 2 — токарно-револьверный станок; 3 — горизонтально-расточной станок; 4 — графопостроитель; 5 — газорезательная машина; 6 — револьверный дыропробный пресс; 7 — горизонтально-фрезерный станок для объемной обработки, пять осей координат; 8 — портальный копировально-фрезерный станок; 9 — вертикальные токарные станки, карусельные станки; 10 — фрезерный станок для объемной обработки с вращающимся столом, пять осей координат; 11 — токарный станок с наклонной станиной, четыре оси координат; 12 — круглошлифовальный станок



Продолж. рис. 5.10

Рис. 5.11
Чертеж обрабатываемой детали. Размеры даны в дюймах (1 дюйм = 25,4 мм):
1—3 — обрабатываемые отверстия; 4 — исходное положение



5.5. КОНТРОЛЬ ПЕРЕМЕЩЕНИЙ

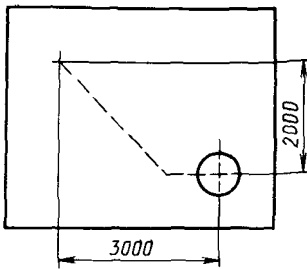


Рис. 5.12

Траектория перемещения инструмента при управлении «от точки к точке»

Степень контроля по пути перемещения инструмента является одним из важных показателей, определяющих стоимость станка с ЧПУ, причем многие операции обработки требуют значительно меньше контроля, чем другие. Инженер по автоматизации должен понимать эту разницу, чтобы принять рациональное решение по капитальным вложениям. Даже экономическая целесообразность автоматизации целиком может зависеть от необходимой

степени контроля перемещений в станке с ЧПУ.

Управление «от точки к точке». Сверление отверстий является важной областью применения станков с ЧПУ. Пока инструмент перемещается из предыдущей позиции для сверления в следующую, никакой обработки не происходит. После окончания позиционирования необходимо контролировать перемещение только по одной оси, обычно по оси z . Эта степень управления перемещениями называется управлением «от точки к точке». Тем же способом может быть осуществлено управление при фрезеровании пазов, параллельных осям x и y и других операциях обработки. На рис. 5.12 показана траектория перемещения инструмента для позиционирования над местом, где должно быть просверлено отверстие при управлении «от точки к точке». Перемещение по осям x и y происходит одновременно и с одинаковой скоростью, пока расстояние по оси u не пройдено. Затем движение продолжается по оси x до конечного положения.

Контурное управление. Обработка криволинейных поверхностей требует управления относительными скоростями перемещения инструмента по двум и более осям. Это осуществляется при помощи изменения напряжения, подаваемого на двигатели переменного тока, которые обеспечивают перемещения по осям во время резания. Если кто-нибудь возьмет на себя труд запрограммировать тысячи перемещений от точки к точке, станок с позиционной системой управления сможет выполнить контурную обработку при помощи прямолинейной аппроксимации. Станок же с контурной системой управления автоматически выполняет сложные вычисления, необходимые для образования криволинейных поверхностей.

5.6. ПРОГРАММИРОВАНИЕ ДЛЯ ЧПУ

Целью программирования является разработка набора команд для выполнения станком на основе одного из двоично-десятичных кодов. Однако в рамках стандартных кодов были разработаны различные схемы для того, чтобы передавать информацию станку.

Некоторые из этих схем называются форматами, потому что они предписывают построение команд в соответствии с определенной формой. Схемы передачи информации высокого уровня похожи на языки программирования для ЭВМ и называются языками программирования для ЧПУ. Прежде чем перейти к дальнейшему изложению, следует отметить, что от формата или языка, выбранного для передачи информации на станок с ЧПУ, выполнение команд не зависит. В целом невозможно определить, какой язык программирования применен, просто наблюдая за работой станка.

Структура кадра. Определенные важные элементы должны присутствовать в каждой команде, другие элементы необязательны, а некоторые вообще редко встречаются. Когда все элементы команды скомпонованы, команду обычно называют «кадром». О главном элементе команды — положении инструмента — уже рассказывалось в разделе «Оси координат». Другим важным элементом является порядковый номер, необходимый для соблюдения последовательности и обеспечения возможности обращения к нужной команде в программе. Может потребоваться задать подачу в дюймах в минуту, миллиметрах в минуту или в дюймах на оборот шпинделя. Если надо, чтобы шпиндель вращался, может оказаться необходимым задать еще один элемент: частоту вращения в оборотах в минуту. Могут быть заданы и другие разнообразные функции, такие, как «Включить охлаждение» или «Выключить охлаждение». Каждый из элементов или составных частей кадра, описанных выше, называется словом. Таким образом, кадры программы являются группами слов, разделенными неким знаком конца кадра.

Форматы кадра. Основным способом обеспечения связи со станком с ЧПУ является запись слов кадра в соответствии с одним из нескольких заданных обусловленных форматов. Некоторые задачи могут быть решены с помощью любого формата, но для каждого формата по-разному. Один формат может быть компактным и эффективным, другой лучше понимается. Некоторые различия относятся к стилистическим. Как и разговорные языки, форматы эволюционируют, и некоторые из них отмирают вместе со станками, для которых они предназначены. Ниже приведено несколько наиболее важных форматов.

1. Жесткая последовательность. Это был один из самых ранних форматов для ЧПУ. Для слов в кадре задавалась определенная последовательность, причем требовалось указывать все слова, вне зависимости от того, означали ли они изменения в состоянии станка или нет. Например, слово, определяющее положение по оси x , должно быть включено в кадр, даже если изменения в положении по этой оси не требуется.

2. Адрес кадра. В этом формате в начале кадра указывается его адрес, чтобы определить, какие слова будут употребляться в данной конкретной команде. Те элементы, которые не перечи-

5.4. Код адреса кадра

Изменение параметра			<i>x</i>	<i>y</i>	<i>x и y</i>	<i>z</i>	Изме- нение поло- жения		
Только положение			00	10	20	30			
Подача			01	11	21	31	41		
Частота вращения			02	12	22	32	42		
Охлаждение			03	13	23	33	43		
Подача и частота вращения			04	14	24	34	44		
Подача и охлаждение			05	15	25	35	45		
Частота вращения и охла- ждение			06	16	26	36	46		
Подача, частота вращения и охлаждение			07	17	27	37	47		
Код частоты вращения									
Частота враще- ния, об/мин	0	77	154	308	424	583	847	1165	2330
Код	0	1	2	3	4	5	6	7	8
Код подачи									
Подача, дюйм/об.	0	0,0025	0,0050	0,0075	0,0100	0,0150	0,0200		
Код	0	1	2	3	4	5	6		
Код охлаждения									
Охлаждение			Код						
Выключено			0						
Включено			1						

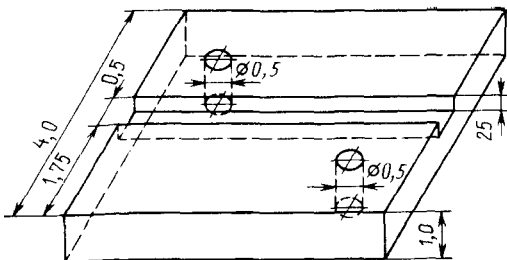
слены, остаются без изменений. В табл. 5.4 приведены некоторые типичные коды адресов кадра.

3. Последовательность с табуляцией. Этот формат похож на первый описанный формат, за исключением того, что опускаемые слова заменяются меткой табуляции. Отчетливое разрядное написание метки (6—5—4—3—2) легко распознается визуально. Такой формат довольно компактен, но уступает в этом формату с адресом кадра.

4. Адрес слова. В этом формате в начале каждого слова ставится буква, являющаяся кодом слова, который указывает, какое слово будет определено. Аналогично табуляции и адресу кадра ненужные слова могут быть опущены, обеспечивая компактность. Адрес слова является очень важным форматом, потому что он позволяет перейти к программированию современного оборудования на базе ЭВМ. Некоторые из наиболее распространенных кодов адресов слов: *N* — номер кадра; *G* — подготовительная функция; *X* — перемещение вдоль оси *x*; *Y* — перемещение вдоль оси *y*; *Z* — перемещение вдоль оси *z*; *F* — подача; *T* — выбор инструмента; *S* — частота вращения шпинделя; *M* — вспомогательные функции; *EOB* — конец кадра.

Рис. 5.13

Чертежи детали для примера 5.13. Размеры даны в дюймах (1 дюйм = 25,4 мм)



В большинстве языков с адресами слов требуется соблюдать некоторую последовательность слов, хотя ненужные слова можно опустить.

В следующем примере иллюстрируются и сравниваются все четыре формата кадров.

Пример 5.1. Программирование фрезеровального станка с ЧПУ. В детали, изображенной на рис. 5.13, необходимо профрезеровать шпоночную канавку и обработать два отверстия. Обработка производится концевой фрезой на вертикально-фрезерном станке. Концевая фреза является универсальным инструментом для фрезерования, позволяющим производить обработку периферией, торцом или периферией и торцом одновременно. Разработайте программы для выполнения обработки, используя форматы с жесткой последовательностью, адресами кадра, табуляцией и адресами слов.

Решение. Принимаем, что нулевая плоскость для оси z проходит на расстоянии 0,25 дюйма (6,35 мм) над деталью.

Определения слов: G0 — быстрый ход; G1 — линейная интерполяция; G7 — возврат в исходное положение; G17 — выбор осей $x-y$; G20 — выбор оси z ; G90 — программирование в абсолютной системе координат; G91 — программирование в приращениях; M0 — отсутствие функций по M; M2 — конец программы; M3 — включение шпинделя; M5 — останов шпинделя.

Подача F определяется умножением подачи на оборот на частоту вращения шпинделя, например:

$$F = 0,0050 \text{ дюйм} \cdot 573 \text{ об/мин} = 2,87 \text{ дюйм/мин} (72,9 \text{ мм/мин}).$$

Частота вращения определяется по формуле

$$S = \frac{\text{Рекомендуемая скорость резания}}{\pi \times (\text{Диаметр инструмента})}.$$

Например:

$$S = \frac{75 \text{ фут/мин} \times 12 \text{ дюйм/фут}}{\pi \times 0,5 \text{ дюйм}} = 573 \text{ об/мин.}$$

Программирование с адресами слов

№ 01	G0 G17 X1.00 Y3.00 EOB
№ 02	G20 Z—0.75 F2.87 S573 M3EOB
№ 03	G20 Z0.00 EOB
№ 04	G0 G17 X—0.50 Y2.00 EOB
№ 05	G20 Z—0.50 EOB
№ 06	G17 X5.25 F2.67 S382 EOB
№ 07	G20 Z0.000 EOB
№ 08	G0 G17 X3.00 Y1.00 EOB
№ 09	G20 Z—0.75 F2.87 S573 EOB
№ 10	G20 Z0.00 EOB
№ 11	M5 EOB
№ 12	M2 EOB

Программирование с жесткой последовательностью

№	X	Y	Z	F	S	M
001	1,00	3,00	0,00	2,87	000	0
002	1,00	3,00	—0,75	2,87	573	3
003	1,00	3,00	0,00	2,87	573	3
004	—0,50	2,00	0,00	2,87	573	3
005	—0,50	2,00	—0,50	2,87	573	3
006	5,25	2,00	—0,50	2,67	382	3
007	5,25	2,00	0,00	2,67	382	3
008	3,00	1,00	0,00	2,67	382	3
009	3,00	1,00	—0,75	2,87	573	3
010	3,00	1,00	0,00	2,87	573	3
011	3,00	1,00	0,00	2,87	000	5
012	3,00	1,00	0,00	2,87	000	2

Программирование с адресом кадра

№	Код адреса кадра	Данные			
01	21	1,00	3,00	2 *	
02	32	—0,75	5 **		
03	30	0,0			
04	20	—0,50	2,00		
05	30	—0,50		4	
06	0,4	5,25	2		
07	30	0,00			
08	20	3,00	1,00	5	
09	34	—0,75	2		
10	30	0,00			
11	42	0			

Конец программы

* Код подачи из табл. 5.4.

** Код частоты вращения из табл. 5.4.

Программирование с таблицей

01	TAB	1,0	TAB	3,0	TAB	TAB	2,87	EOB		
02	TAB	TAB	TAB	—0,75	TAB	TAB	573	TAB	3	EOB
03	TAB	TAB	TAB	0,00	EOB					
04	TAB	—0,50	TAB	2,0	EOB					
05	TAB	TAB	TAB	—0,50	EOB					
06	TAB	5,25	TAB	TAB	TAB	2,67	TAB	382	EOB	
07	TAB	TAB	TAB	0,00	EOB					
08	TAB	3,00	TAB	1,00	EOB					
09	TAB	TAB	TAB	—0,75	TAB	2,87	TAB	573	EOB	
10	TAB	TAB	TAB	0,00	EOB					
11	TAB	TAB	TAB	TAB	TAB	000	TAB	5	EOB	
12	TAB	TAB	TAB	TAB	TAB	TAB	2	EOB		

Хотя форматы кадров отличаются, они согласуются по последовательности и процедуре представления данных для станков с ЧПУ. Все они требуют, чтобы любое изменение положения по осям координат в декартовой системе включалось в кадр. Это удобно для станкостроителей, но персоналу, занимающемуся программированием обработки на станке, предпочтительно иметь систему программирования, более ориентированную на пользователя. Для этой цели служат разработанные языки программирования, о которых рассказано в следующем разделе.

Языки программирования. Для многих программистов станков с ЧПУ удобнее писать программы с использованием формата, включающего ограниченный набор английских слов, таких, как GOTO, RUN, EXECUTE и др. Этот способ используется в языках программирования для ЭВМ общего применения, например в языках Бейсик, Фортран и Кобол. Те, кто занимается программированием на одном или нескольких языках, найдет сходство между такими языками, как АПТ и Компакт, в которых тоже используется такой формат.

Есть и другое, более важное преимущество использования языков программирования для программирования обработки на станках с ЧПУ — символическое адресование. В таких языках, как АПТ, программист может задать точку в декартовых координатах и затем, позже, отменить или возвращаться к этой точке в программе. Это позволяет экономить на подробном задании координат по x , y и z в тысячных, десятичных долях дюйма или в сотых долях миллиметра.

Программирование контурных перемещений на станках, позволяющих выполнять операции контурной обработки, нецелесообразно без использования языка программирования. Контурные поверхности описываются математически как части стандартных геометрических поверхностей, обрабатываемых инструментом, а программное обеспечение ЭВМ выполняет интерполяцию, необходимую для управления станком. Вообразите попытку решить эту задачу, используя формат кадра, в котором координаты x , y и z каждой интерполированной точки должны быть рассчитаны и детально определены программистом. Справедливости ради здесь необходимо указать, что для удобства программирования в форматах кадра также предусмотрена возможность задавать обработку дуг и простых геометрических форм, но программирование контуров не является сильной стороной этого способа программирования.

Ранее говорилось о том, что некоторые форматы компактны и эффективны, в то время как другие легче воспринимаются. То же самое можно сказать и об языках программирования. Как и форматы, языки программирования эволюционируют. Языки программирования делятся на семейства, что позволило применить термин «диалект» в отношении к языку, родственному другому, более известному.

В задачу этой книги не входит глубокое ознакомление с каждым из множества языков программирования и даже с одним конкретным языком, используемым в станкостроении. Эта задача должна решаться в отдельном курсе, вероятно применительно к конкретному имеющемуся оборудованию или к оборудованию, которое предполагается закупать. Здесь же мы хотим дать инженеру по автоматизации общее понятие о языках, их сильных и слабых сторонах, а также об их месте в решении проблемы роботизации и автоматизации промышленности.

С целью сравнения с форматами, использованными при решении примера 5.1, ниже приведена аннотированная программа на языке АПТ. Конечно, не предполагается, что читатель овладеет языком АПТ, просто изучив эту программу, но каждая команда и структура этого языка станет ему ясна. Для ускорения процесса обучения программа составлена для той же обработки, что и в примере 5.1.

Программа на языке АПТ

Графы 1—6	Графа 10	Примечания (не являются частью программы)
PARTNO	THE 2 HOLES AND KEYWAY PGM MACHIN/MILL,2	Обозначение программы Обозначение используемого инструмента Диаметр фрезы
RT0	CUTTER/.500	
RT1	POINT/0,0,0	
RT2	POINT/1.0,3.0,0	
RT3	POINT/— .25,2.0,0	
RT4	POINT/5.25,2.0,0	
	POINT/3.0,1.0,0	Положение точек
	FEEDRAT/2.87	
	SPINDL/573	Подача
	FROM/PTO	Частота вращения шпинделя
	GOTO/PT1	Исходное положение
	GODLTA/0,0,—0.75	Над отверстием 1
	GODLTA/0,0,+0.75	Сверление отверстия 1
	GOTO/PT2	
	GODLTA/0,0,—0.5	Положение для обработки паза
	FEEDRAT/2.67	
	SPINDL/382	Режимы резания
	GODLTA/5.75,0,0	Фрезерование паза
	GODLTA/0,0,+0.5	В точке 3
	GOTO/PT4	Над отверстием 2
	FEEDRAT/2.87	Режимы резания при сверлении
	SPINDL/573	
	GODLTA/0,0,—0.75	
	GODLTA/0,0,+0.75	Сверление отверстия 2
	GOTO/PTO	Возврат в исходное положение
	SPINDL/O	Останов шпинделя
	FINI	Конец (программы)

Из показанного примера должны быть очевидны преимущества языка программирования, подобного языку программирования для ЭВМ, хотя данный пример в действительности слишком прост, чтобы в полном объеме показать возможность такого языка, как АПТ, при контурной обработке. В середине 70-х годов стало казаться, что преимущества программирования с помощью ЭВМ при использовании соответствующих языков приведут к преобладанию языков, а затем и к фактическому вытеснению ими неудобных форматов кадра. Однако в начале 70-х годов произошло нечто, что опрокинуло эту тенденцию. Это описано в следующем разделе.

5.7. ПРЯМОЕ ЧИСЛОВОЕ ПРОГРАММНОЕ УПРАВЛЕНИЕ

Достижением конца 60-х годов стало дистанционное управление станками с ЧПУ от универсальных ЭВМ в режиме разделения времени. Этот вид управления стал называться прямым программным управлением. Лучшее всего понять значение прямого программного управления можно, изучив особенности обычного ЧПУ, которое оно заменяет.

Обычное числовое программное управление. Обычное ЧПУ, в отличие от прямого, осуществляется с помощью форматов кадра, которые служат непосредственно для подготовки перфоленты. Программирование также может быть выполнено на одном из языков с последующим переводом исходной программы с помощью ЭВМ в формат кадра, воспринимаемый станком. Подготовленная перфолента извлекается затем из перфоратора и соединяется в кольцо. Это кольцо устанавливается в считывающее устройство станка, и информация с него считывается во время выполнения программы. Важно уяснить себе, что перфолента физически проходит через считывающее устройство каждый раз, когда изготавливается новая деталь или программа выполняется снова. В режиме реального времени станок фактически выполняет программу по мере того, как она команда за командой считывается считывающим устройством.

Не удивительно, что «бумажные» перфоленты на самом деле изготавливаются из майлара, а не из бумаги, что позволяет многократно использовать их в течение долгого времени.

Режим разделения времени. В цехах заводов 60-х годов громадные фрезерные станки с ЧПУ выполняли обработку по программе, в то время как кольца перфолент постоянно прокручивались через их считывающие устройства. Наверху же, в вычислительном центре завода, дорогая высокоскоростная универсальная ЭВМ, олицетворявшая гордость конструкторского отдела и отдела обработки данных, за секунды или даже миллисекунды выполняла сложные программы. Было хорошо известно, что объем памяти этих больших ЭВМ позволяет хранить программы обработки для станков с ЧПУ, работающих в цеху, но непреодолимый барьер высокой стоимости не позволял использовать ЭВМ

таким образом. Кроме того, ЭВМ работает слишком быстро, чтобы ожидать, пока фрезерный станок отработает выдаваемые ею команды. Пришлось бы оснащать дорогой ЭВМ каждый станок, что, конечно, экономически нецелесообразно.

В конце 60-х годов разработка для ЭВМ режима разделения времени обеспечила управление целыми группами станков с помощью одной ЭВМ. Все станки с ЧПУ присоединялись непосредственно к ЭВМ, выдавшей программы в реальном времени без использования перфолент. Высокая скорость работы ЭВМ позволяла обслуживать несколько станков, выдавая команды в нужной последовательности.

Дополнительным преимуществом, создаваемым прямым программным управлением, является возможность хранения информации на магнитной ленте, дисках или на других мощных программоносителях, обеспечивающих ее немедленное извлечение и передачу в активную память ЭВМ для управления тем или иным станком. Если программу необходимо изменить, ЭВМ проделывает это мгновенно и перезаписывает ее на диск. Сравните это со старым методом подготовки новой перфоленты и ручного склеивания ее в кольцо. Даже просто хранение старых колец перфолент создавало неудобства, особенно из-за их различной длины. Неудивительно поэтому, что новые системы прямого программного управления, которые заменили старые перфоленты, казались вершиной автоматизации в свое время. Но у этих систем были свои недостатки, поэтому их применение было недолгим. Технология, породившая промышленные роботы, уже была на подходе.

Закат прямого программного управления. Могучие универсальные ЭВМ 60-х годов, разбросавшие сети кабелей прямого управления станками по всему заводу, были чудом века, когда они работали. Любой сложной механической или электронной системе приходится простаивать в результате отказа, переделок или обслуживания, и большим универсальным ЭВМ присущи эти черты. Но когда ЭВМ непосредственно управляет всеми станками с ЧПУ на заводе, работая в режиме разделения времени, ее отказ означает прекращение всех производственных процессов, связанных с ЧПУ. Этот непреложный факт означал приговор всей концепции прямого программного управления.

Наряду с неприятностями, создаваемыми отказами ЭВМ, существуют проблемы электрических наводок в длинных кабелях управления, обеспечивающих необходимую информационную связь между ЭВМ и станком. В условиях производства, где работают эти системы, большая нагрузка силовых кабелей высокого напряжения, питающих мощные станки, вызывает помехи в информационных кабелях низкого напряжения систем прямого программного управления. Количество ошибок по четности резко возрастает, увеличивая время простоев.

Жизнеспособной формой прямого программного управления, существующей и сегодня, оказалась система, называемая иерар-

хической системой ЧПУ. В ней прямое управление заменяется распределенным. Иерархическую систему лучше описать после рассмотрения следующего этапа развития программного управления: CNC.

5.8. ЧИСЛОВОЕ ПРОГРАММНОЕ УПРАВЛЕНИЕ ОТ ЭВМ

В 70-х годах технология изготовления ЭВМ сделала шаг вперед благодаря налаживанию производства все более дешевых электронных схем все меньшего размера. Появился термин микро-ЭВМ, и вдруг обнаружилось, что ЭВМ стали достаточно дешевы, чтобы оснастить ими все станки с ЧПУ на заводе. Эти станки получили нечто подобное прямому программному управлению, только лучше: ЭВМ не должна управлять сразу несколькими станками. Эта новая техника позволила обходиться без кабелей управления, передача информации по которым в производственных условиях искажается электрическими наводками.

Такое управление, при котором каждым станком управляет своя ЭВМ, называется ЧПУ от ЭВМ или CNC. Практически все современные станки с ЧПУ имеют такой тип управления. Многие старые станки с программным управлением или предназначенные для прямого программного управления были модернизированы для управления типа CNC. Этой модернизации были подвергнуты даже некоторые обычные станки с ручным управлением, изготовленные десятилетия назад, до изобретения ЧПУ. Несколько компаний производят наборы для модернизации вертикальных фрезерных станков.

На рис. 5.14 показан такой станок фирмы «Бриджпорт».

Направление развития станков с ЧПУ неожиданно претерпело резкое изменение, вызванное тем, что новые системы типа CNC возродили интерес к форматам блоков информации, особенно к адресам слов. Причиной этого очевидного шага назад послужило то, что введение CNC позволило операторам самостоятельно составлять управляющие программы, пользуясь клавиатурой системы управления. На рис. 5.14 видна панель управления системы ЧПУ вертикального фрезероувального станка. По виду она чем-то напоминает панели цифровой индикации, с которыми имеют дело операторы обычных станков с ручным управлением.

Познательность программирования и характеристики форматов блочного типа обеспечивают методику работы, аналогичную применяемой операторами обычных станков. Наличие ЭВМ в составе системы управления позволяет производить корректировку программ прямо на станке, не обращаясь в некий удаленный вычислительный центр или на терминал инженера. Объединение станка и ЭВМ породило новую специальность оператора-программиста. Оператор учится программировать, а программист получает навыки управления станком. Системы ЧПУ типа CNC сближают эти профессии.



Рис. 5.14

Станок фирмы «Бриджпорт», оснащенный устройством ЧПУ типа CNC модели «Смарт» фирмы «Аэротех»

Хотя на рис. 5.14 показана конкретная система ЧПУ, панели систем управления типа CNC имеют много общего. Типовая система ЧПУ имеет следующие группы кнопок управления.

1. Выбор режима работы. Этот сектор панели дает возможность пользователю выбрать следующие режимы: режим ручного управления, автоматическое выполнение программы команда за командой, полное выполнение всего цикла, автоматическое повторение циклов, ввод новой программы, редактирование введенных команд.

2. Управление. Этот сектор панели позволяет осуществлять выбранный режим. Например, пользователь может вручную задавать перемещения по осям, запустить программу или приостановить ее выполнение в автоматическом режиме, ввести новую команду в режиме ввода программы, вывести на дисплей или отыскать нужную команду в режиме редактирования.

3. Клавиатура. Панель управления системы ЧПУ типа CNC должна иметь цифровую клавиатуру для ввода значений координат и дополнительных функций, например подачи. Если система управления предназначена для программирования на языке с адресами слов (а большинство систем именно такого типа), на клавиатуре должны иметься и некоторые буквы алфавита.

4. Дисплей. Система управления типа CNC должна иметь устройство для отображения текущих команд управления, текущего положения инструмента, кодов ошибок и других данных цифрового или буквенно-цифрового вида. Обычно это осуществляется с помощью светодиодного индикатора или экранного дисплея.

Некоторые модели систем управления оснащаются считывающими устройствами для кассет с информацией на магнитной ленте, устройствами ввода-вывода для обмена программами с внешней ЭВМ. Другим полезным свойством, которым обладает большинство систем, является возможность изменения частоты вращения шпинделя или подачи в долях (в процентах) заданного программой значения. Это особенно важно при отладке программ.

5.9. ИЕРАРХИЧЕСКОЕ ПРОГРАММНОЕ УПРАВЛЕНИЕ

Прямое программное управление сохраняется в форме, названной иерархическим программным управлением. Такая форма

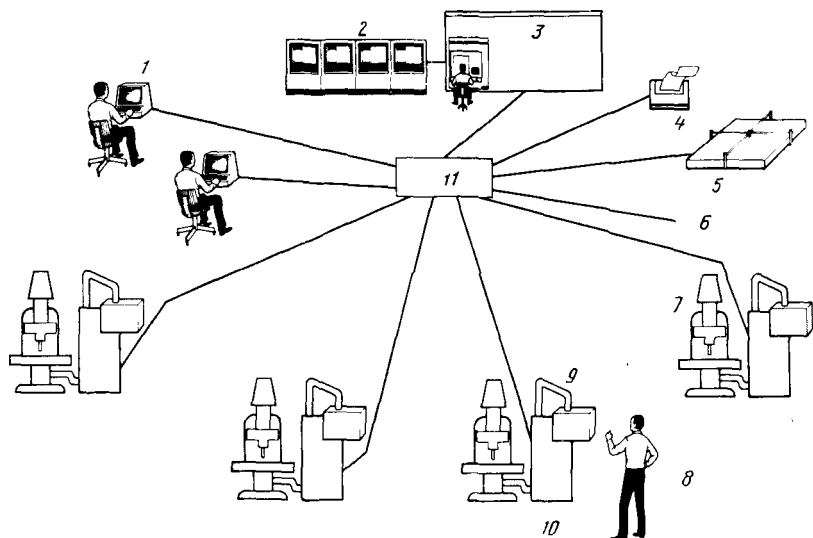


Рис. 5.15

Структура иерархической системы управления:

1 — программирование с удаленного терминала; 2 — дисководы памяти на магнитных дисках для хранения программ обработки и математического обеспечения комплексной системы автоматизированного проектирования и изготовления с помощью ЭВМ; 3 — центральная ЭВМ; 4 — печатающее устройство; 5 — графопостроитель; 6 — оборудование с ЧПУ и технологические процессы; 7 — металлорежущие станки с ЧПУ; 8 — оператор-программист станков с ЧПУ; 9 — панель управления устройства ЧПУ; 10 — управляющая ЭВМ; 11 — интерфейс

управления сочетает лучшие черты программного управления с помощью ЭВМ и прямого программного управления. При иерархическом программном управлении каждый станок имеет свою собственную систему ЧПУ на базе ЭВМ, соединенную с более крупной центральной ЭВМ, но связанную с ней только периодически в режиме разделения времени. В отличие от прямого программного управления при отказе центральной ЭВМ иерархической системы управления подчиненные системы ЧПУ на станках имеют ограниченную возможность функционировать самостоятельно. Центральная ЭВМ обычно управляет процессом производства на базе выборочного контроля, содержит библиотеку управляющих программ, доступную для подчиненных систем ЧПУ, а также воспринимает сигналы о неисправностях и сигнализирует о них. Возможно даже организовать промежуточный иерархический уровень управления среди ЭВМ, как это показано в гл. 13. На рис. 5.15 изображена структура типичной иерархической системы управления.

5.10. КОМПЛЕКСНАЯ СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ И ИЗГОТОВЛЕНИЯ С ПОМОЩЬЮ ЭВМ

Возможности ЭВМ для системы управления значительно шире, чем просто хранение подготовленных управляющих программ. Накопитель на дисках предназначен не только для хранения программ, но и для хранения программного обеспечения комплексной системы автоматизированного проектирования и изготовления изделий с помощью ЭВМ. Применение ЭВМ в автоматизированном производстве многообразно, и в гл. 13 этот вопрос рассмотрен более подробно. Следует отметить особую роль, которую играет ЭВМ в процессе конструирования.

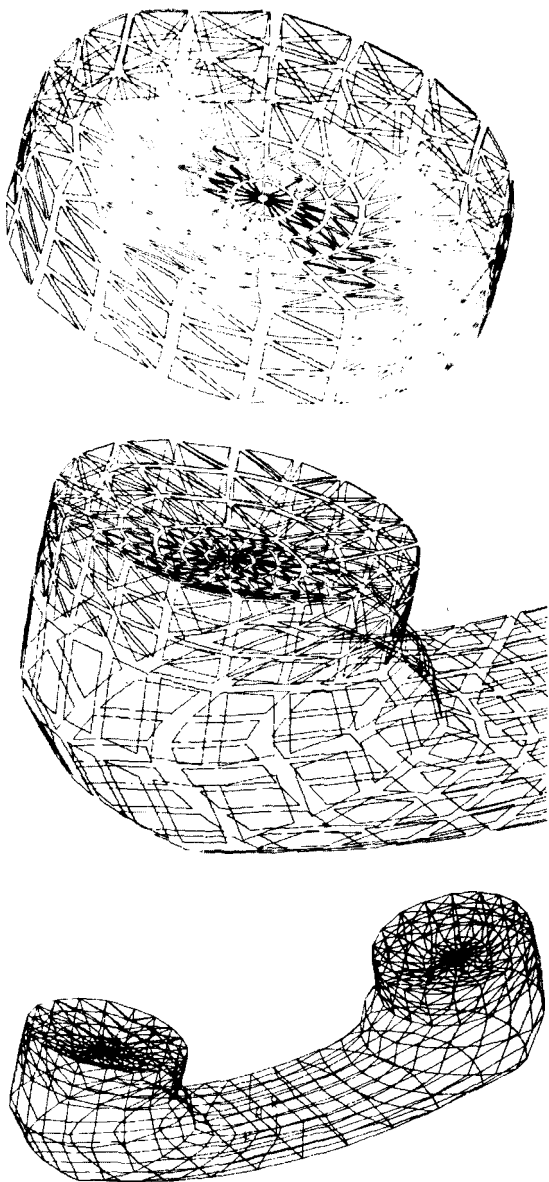
На рис. 5.16 приведены изображения изделия, возникающие на дисплее при конструировании с помощью ЭВМ. Соответствующая программа обработки может быть автоматически подготовлена с помощью программного обеспечения, заложенного в ЭВМ и использующего исходные данные, полученные при проектировании. Это позволяет избежать обычной утомительной рутины при подготовке управляющих программ. Еще более популярными являются менее совершенные системы, в которых программист пишет программу обработки на языке, подобном АПТ, и затем использует программное обеспечение системы автоматизированного проектирования и изготовления для ее проверки, моделируя процесс обработки с помощью ЭВМ. ЭВМ выявит синтаксические ошибки и уведомит об их наличии до выполнения программы, а также проведет диагностирование программы. По откорректированной с синтаксической точки зрения программе ЭВМ моделирует ее выполнение и создает чертеж «обработанной» детали либо на экране дисплея, либо на графопостроителе по выбору програм-

Рис. 5.16

Графическое изображение изделий при конструировании с помощью ЭВМ

миста. После получения удовлетворительной программы ЭВМ может передать ее в систему ЧПУ станка и записать на магнитном диске для будущего использования. Может быть также создано объемное изображение детали, получены ее размеры и подготовлена программа для робота на передачу этой детали со станка на станок во время ее обработки.

Одним из принципиальных преимуществ автоматизированного проектирования является возможность получения полного комплекта документации на изделие и процесс его изготовления в дополнение к информационной базе данных. База данных может быть использована для разработки графиков производства, определения потребностей в материалах, разработки руководств по эксплуатации, карт контроля качества и другой документации, которая, с одной стороны, полезна для производства а с другой — сама является продуктом производства.



В ы в о д ы

Промышленные роботы имели в качестве предшественников автоматизированные токарные и фрезерные станки. Первые попытки автоматизации станков осуществлялись за счет кулачков

и механических связей, что является стратегией «жесткой» автоматизации. Но развитие цифровых ЭВМ позволило сделать автоматизацию станков перепрограммируемой и автоматизировать мелкосерийное производство. Это же технологическое достижение позже привело к возникновению промышленных роботов. По сути, робот можно рассматривать как разновидность станка с ЧПУ. Некоторые из них, например сварочные роботы, выполняют аналогичные функции. Очевидно, что ЧПУ обеспечивает не только гибкость управления автоматическим станками, но и многие другие функции. Большие и сложные детали, характерные для аэрокосмической промышленности, не пригодны для производства с использованием жесткой автоматизации. Возможность выполнения объемных сложных операций на одном станке и в одном крепежном приспособлении позволяет экономить на стоимости наладки, приспособлении и инструментальной оснастки, одновременно повышая качество.

Кодирование информации для ЧПУ выполняется в двоично-десятичной системе, причем ошибки, могущие возникнуть при набивке или считывании программ, контролируются с помощью дорожки четности. Для каждого станка с ЧПУ определено соответствие между направлением каждого перемещения рабочего органа и определенной геометрической осью. Задание перемещения инструменту производится в приращениях или в абсолютной системе координат. Перемещение инструмента может контролироваться в конечных точках (программирование от точки к точке) или непрерывно (контурное управление).

Программирование станков с ЧПУ может осуществляться двумя способами: форматами кадра или на языках программирования. Типичным языком такого рода является язык АРТ.

Прямое программное управление предполагает использование центральной ЭВМ для управления несколькими станками с ЧПУ в режиме реального времени. Прямое программное управление в целом уступило свое место системам ЧПУ типа CNC, где каждому станку придается своя собственная ЭВМ. Лучшие черты прямого программного управления и ЧПУ типа CNC собраны в системе управления, называемой иерархической. Такие системы позволяют программировать одновременно и с помощью центральной ЭВМ, и прямо на рабочем месте, с помощью ЧПУ. Разработано всеобъемлющее программное обеспечение для автоматизированного проектирования и изготовления изделий с помощью ЭВМ. Пакеты программ записаны на магнитных дисках, доступных центральной ЭВМ в режиме реального времени. Наряду со станками, центральная ЭВМ может управлять промышленными роботами, что описано в гл. 6.

Упражнения и вопросы

5.1. Программа на перфоленте (рис. 5.17) имеет ошибку по четности. Можете ли вы ее определить? В этом примере имеется ошибка лишь одного типа из двух возможных. Какие типы ошибок вам известны?

5.2. Программа на перфоленте с рис. 5.17 составлена с применением формата кадра. Какой это формат? Как вы это определили? Какой номер у единственного полного кадра программы?

5.3. Программа на перфоленте, показанной на рис. 5.18, составлена с применением формата кадра. Какой это формат? Как вы это определили? Сколько кадров показано?

5.4. Станок, работающий по программе, приведенной ниже, фрезерует букву на пластинке, прикрепленной к столу. Координата поверхности стола $z = -3800$. Координаты положения инструмента задаются в тысячных долях дюйма. Глубина обработки составляет 0,2 дюйма (5,08 мм).

№ 1	G91 EOB
№ 2	G7 EOB
№ 3	G1 G17 X4000 Y-3000 F100 M3 EOB
№ 4	X-500 Y500 F40 EOB
№ 5	G20 Z-3200 F100 EOB
№ 6	Z-300 F3 EOB
№ 7	G17 X100 F10 EOB
№ 8	G17 X-1000 Y-1000 F10 EOB
№ 9	G17 X1000 F10 EOB
№ 10	G20 Z1700 F100 EOB
№ 11	G7 M5 EOB
№ 12	M2 EOB

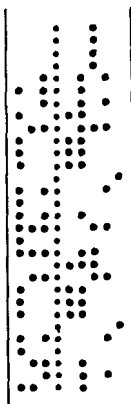


Рис. 5.17
Программа на перфоленте с ошибкой по четности

Какой формат был использован в этом примере? Программирование было выполнено в приращениях или в абсолютной системе координат? Какая буква фрезеруется при помощи этой программы?

5.5. Какова толщина материала, обрабатываемого в примере 5.4? Как близко к обрабатываемой поверхности подходит инструмент непосредственно перед резанием? Как высоко над столом поднимается инструмент после обработки? На каком расстоянии от обрабатываемой поверхности находится инструмент в этот момент?

5.6. Следующий фрагмент программы написан в абсолютной системе координат. В алюминиевой заготовке фрезеруется паз шириной $\frac{3}{8}$ дюйма (9,525 мм) и глубиной $\frac{1}{4}$ дюйма (6,35 мм). Координаты задаются в тысячных долях дюйма.

```
00119+03000+04000-0245010065003 EOB
00217+05500+04000-0245001565003 EOB
```

Какой формат использован в этом фрагменте программы?

5.7. Перепишите фрагмент программы из упражнения 5.6, используя систему в приращениях.

5.8. Используя систему в приращениях, перепишите программу из упражнения 5.6 при помощи табуляции.

5.9. На рис. 5.19 указаны размеры паза, который необходимо обработать на фрезервальном станке с ЧПУ. Используется концевая фреза диаметром 0,200 дюйма (5,08 мм), а глубина фрезерования составляет 0,150 дюйма (3,81 мм).

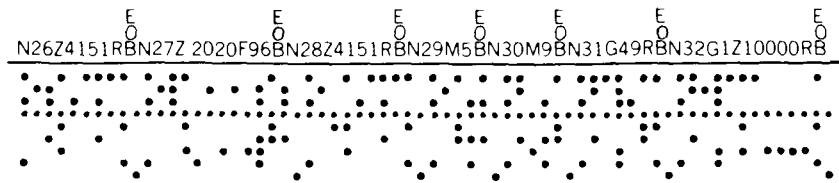


Рис. 5.18
Программа на перфоленте с применением формата кадра

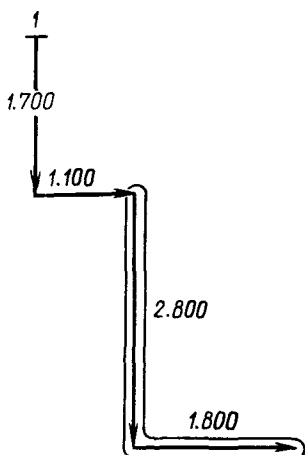


Рис. 5.19

Чертежи паза, обрабатываемого на станке с ЧПУ. Размеры даны в дюймах (1 дюйм = 25,4 мм):

1 — исходное положение инструмента

Размеры характеризуют перемещения центра диаметра инструмента. Каковы габариты изображенного паза? Почему наблюдается расхождение между длинами пути инструмента и размерами паза?

5.10. Не используя функции N, G, F, S и M, составьте фрагмент программы на обработку паза, изображенного на рис. 5.19, в абсолютном формате слов, приняв, что исходная точка инструмента находится на расстоянии 5,500 дюйма (63,5 мм) от поверхности материала. Записывайте только слова координат в этом примере.

5.11. Перепишите программу по примеру 5.10, используя систему вращения и формат адреса слов.

5.12. Почему в мелкосерийном производстве автоматизация с использованием ЧПУ предпочтительнее «жесткой» автоматизации?

5.13. При условиях обработки, аналогичных приведенным в упражнении 5.4, определите, какая буква будет обработана по следующей программе:

```

№ 1   G91 EOB
№ 2   G7 EOB
№ 3   G1 G17 X4000 Y—3000 F100 M3 EOB
№ 4   X—500 Y500 F40 EOB
№ 5   G20 Z—3200 F100 EOB
№ 6   Z—300 F3 EOB
№ 7   G17 X1000 Y—1000 F10 EOB
№ 8   G20 Z300 F100 EOB
№ 9   G17 Y1000 F40 EOB
№ 10  G20 Z—300 F3 EOB
№ 11  G17 X—1000 Y—100 F10 EOB
№ 12  G20 Z1700 F100 EOB
№ 13  G7 M5 EOB
№ 14  M2 EOB
  
```

Каково будет расстояние от рабочего органа станка до поверхности детали после обработки кадра № 5?

5.14. Подготовьте управляющие программы по типу программ, приведенных в примерах 5.4 и 5.13 для обработки: а) буквы L, б) буквы Y и в) буквы M. Обратите внимание, что программы для букв L и M больше похожи на программу для буквы Z, а программа для буквы Y подобна программе для буквы X.

5.15. На рис. 5.11 диаметры отверстий не указаны, поскольку это не влияет на составление программы обработки. Как может диаметр отверстия повлиять на программу?

6. ПРОМЫШЛЕННЫЕ РОБОТЫ

Понятие «робот» в общественном мнении совпадает с понятием «промышленная автоматизация». Это лишь частично правильно, потому что, во-первых, робот является лишь элементом общей системы автоматизации, а, во-вторых, самое представление о роботе весьма преувеличено. Однако, несмотря на все неувязки, роботы сами по себе и публичный ажиотаж вокруг них помогают энергично продвигать промышленную автоматизацию на заводы. Эта книга, наверное, не была бы написана, если бы не интерес, высказываемый экономистами, инженерами, студентами и общественностью по отношению к промышленным роботам.

Большинство людей, услышав слово «робот», вспоминает о R2D2 * из «Звездных войн». Такая ассоциация имеет некоторый смысл, поскольку слово «робот» в действительности родилось на сцене, а не в цеху. Роботы впервые появились в Нью-Йорке 9 октября 1922 г. в пьесе «Р. У. Р». Их создателем был чехословацкий драматург Карел Чапек, а слово «робот» является сокращением чешского слова *robota* — «работа» [69].

В 1956 г., более чем через 30 лет после выхода в свет пьесы Чапека, но до появления «Звездных войн», была образована фирма «Юнимейшн». Она целиком занималась робототехникой. Через 16 лет, в 1972 г., затратив 12 млн. долл., «Юнимейшн» выпустила первый робот, предназначенный для промышленности.

В отличие от робота R2D2 или роботов Чапека, большинство настоящих промышленных роботов мало напоминают людей. В действительности более подходящим для большинства промышленных роботов был бы термин «механическая рука». На рис. 6.1 изображена популярная модель робота с механической рукой, далекого от R2D2, но значительно более полезного с точки зрения производства.

Робот с рис. 6.1 отвечает определению понятия «робот», разработанному Американским институтом роботов:

«Роботом называется перепрограммируемый многофункциональный манипулятор, предназначенный для перемещения мате-

* R2D2 — имя фантастического робота из популярного в 70-х годах фильма «Звездные войны».



Рис. 6.1
Популярная модель робота типа «механическая рука»

риалов, деталей, инструментов или специальных приспособлений посредством различных программируемых движений с целью выполнения разнообразных задач» [70].

Более общее определение робота предложено Микелом Грувером:

«Промышленным роботом называется универсальное программируемое устройство, обладающее определенными антропоморфными характеристиками» [25].

Определение Грувера не сводит концепцию робота к манипуляторам, а наоборот, открывает возможности к расширению различных дополнительных антропоморфных (человекоподобных) характеристик, таких, как мышление, способность принимать решение и зрение.

Не все промышленные роботы походят на механические руки. Рабочая зона некоторых из них представляет собой объем, ограниченный параллелепипедом, ребра которого оформлены в виде направляющих движения робота. Этот тип роботов иногда называется порталным.

Ключевым словом в определении промышленного робота является слово «программируемый». Эта характеристика робота, реализованная с появлением недорогих микросхем в 70-х годах, более чем другие способствовала приходу роботов на рабочие места. В гл. 14 эти микросхемы рассмотрены более подробно.

Задолго до появления программируемых роботов существовали манипуляторы с механически обусловленными движениями, задаваемыми кулачками. В японской промышленности такие манипуляторы также включаются в разряд роботов, но эта интерпретация является спорной. Расхождение в определениях робота может исказить статистические данные, используемые для сравнения количества роботов в различных странах.

Манипуляторы с механическим кулачковым управлением, причисляются они к роботам или нет, играют важную роль в заводской автоматизации. Эти манипуляторы являются иллюстрацией к термину «жесткая автоматизация», приведенному в гл. 1 и являющемуся противоположностью «гибкой автоматизации», представителем которой является программируемый промышленный робот.

6.1. ГЕОМЕТРИЯ РОБОТА

Поскольку конструктивно роботы могут сильно отличаться друг от друга, целесообразно дать некоторую классификацию роботов по их геометрии. В промышленности в этом случае для описания количества возможных движений используют понятие степеней подвижности. Вид этих движений и способ их обеспечения определяют тип робота.

Степени подвижности. Каждое место сочленения частей робота, в котором какой-либо вид привода обеспечивает движение этих частей, за исключением захватов и инструментов, называется степенью подвижности. Движение может быть вращательным или возвратно-поступательным, как, например, под воздействием пневматического или гидравлического цилиндра. На рис. 6.2 показан робот с шестью степенями подвижности: 1) вращение основания, 2) поворот плеча, 3) поворот локтевого сустава, 4) сгибание запястья, 5) поворот запястья, 6) вращение запястья.

За небольшим исключением, у роботов степени подвижности образуют некоторую последовательность. Так, первая степень подвижности у робота на рис. 6.2 — вращение основания — сообщает движение всем частям робота, имеющим последующие степени подвижности. Напротив, третья степень подвижности (поворот локтя) не влияет на поворот основания. Из этого следует, что наиболее сложными движениями являются движения тех частей

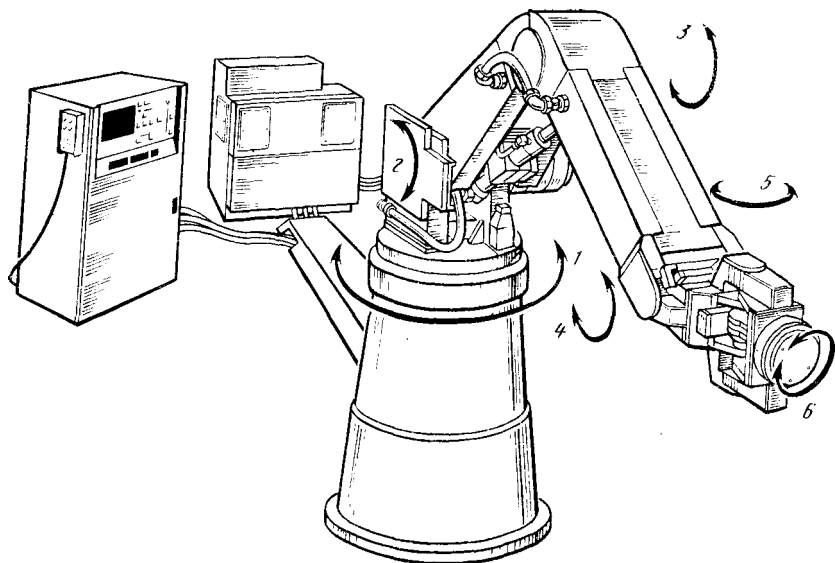


Рис. 6.2

Промышленный робот с шестью степенями подвижности (свободы):

1 — вращение основания; 2 — поворот плеча; 3 — поворот локтевого сустава; 4 — сгибание запястья; 5 — поворот запястья; 6 — вращение запястья

робота, которые имеют более высокую степень подвижности. В общем робот с большим количеством степеней подвижности может выполнять более сложные движения, однако следует учитывать и другие важные факторы, такие, как диапазон и качество движения при данном количестве степеней подвижности. Этот вопрос далее рассмотрен более подробно.

Сочетание различных степеней подвижности и соответствующие виды движения определяют то, как робот будет выглядеть в действительности. Теоретически может быть много разновидностей роботов с шестью степенями подвижности. На практике, однако, почти все роботы можно отнести к одному из нескольких распространенных типов.

Роботы с многозвенной рукой. Некоторые роботы, например показанный на рис. 6.1, фактически работают как человеческая рука. Такие роботы называют роботами с многозвенной рукой. Основание робота поворачивается наподобие человеческого торса. Плечевой и локтевой суставы большинства таких роботов имеют по одной оси поворота, которые перпендикулярны к руке и параллельны плоскости, на которой устанавливается основание. Запястье робота с многозвенной рукой почти всегда может сгибаться, а поворот и вращение могут и отсутствовать. Читатель может заметить, что человеческая рука также очень хорошо сгибается, а поворот и вращение у нее ограничены. Поворот обычно не превышает 60° . Фактически сама кисть человеческой руки почти не

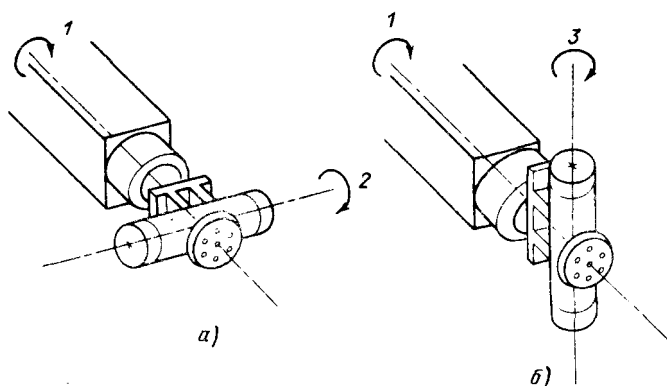


Рис. 6.3

Осуществление движения типа поворота запястья у некоторых типов роботов: развернуть запястье на 90° вокруг оси поворота и затем согнуть:

1 — ось вращения запястья; 2 — ось сгибания запястья; 3 — ось поворота запястья

поворачивается, но используя локтевой и даже плечевой суставы, можно получить поворот около 270° . Благодаря этому человек может вернуть лампочку или закрутить гайку-барашек, но только путем ряда поворотов с перехватом. У большинства роботов с многозвенной рукой угол поворота кисти больше, чем у человеческой руки.

Существует прием, который помогает осуществить движение типа поворота у пятиосевого робота, не имеющего такого специального поворота. Для некоторых случаев можно обеспечить поворот на 90° , а затем и сгибание по оси сгибания. Этот метод показан на рис. 6.3. Обратите внимание, что больше всего прием подходит для одноточечного инструмента и не подходит для роботов с захватами. В последнем случае может обнаружиться, что заготовка или инструмент в захвате робота неправильно ориентированы.

Вариантом робота с многозвенной рукой является робот с горизонтальным сочленением, что просто означает, что оси поворота по второй, третьей и четвертой степеням подвижности направлены вертикально, что заставляет звенья робота сгибаться в горизонтальной плоскости. Такой робот показан на рис. 6.4. Этот тип робота не так напоминает человеческую руку, как предыдущий, но он также завоевал популярность благодаря удобству использования в простых операциях. Еще более странным вариантом многозвеного робота является гибкий робот-хобот, показанный на рис. 6.5. «Хобот» является удачным названием для этого робота, поскольку он действительно напоминает хобот слона, изгибающийся практически в любом направлении. Преимуществом этого типа движения является то, что робот может проникнуть своим «хоботом» в труднодоступные места и затем направить рабочий орган (обычно краскопульт) почти в любом направлении.

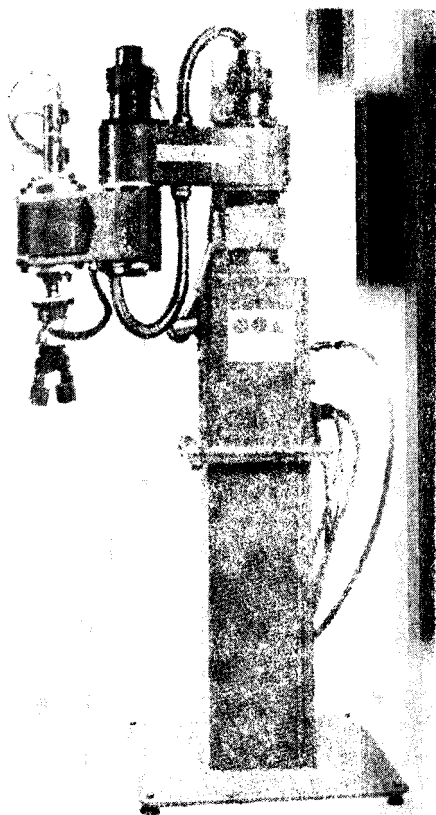


Рис. 6.4

Робот со звеньями, сгибающимися в горизонтальной плоскости

Промышленные роботы, работающие в полярных координатах. Роботы, работающие в полярных или цилиндрических координатах выглядят совершенно иначе, чем роботы с многозвенной рукой, но отличие фактически имеется только в третьей оси (3-й степени подвижности). Вместо локтевого сустава такой робот имеет пневматический или гидравлический цилиндр, который обеспечивает удлинение руки. Робот, работающий в полярных координатах (рис. 6.6), очень распространен в автомобилестроении.

Типичным движением робота является подъем детали по вертикали с сохранением ее ориентации. Далее показано, что для

робота в полярных координатах такое движение требует одновременных скоординированных движений по трем осям: в плечевом суставе, в руке и в запястье. Для роботов с многозвенной рукой это также представляет проблему, так как требует одновременных скоординированных движений плечевого и локтевого суставов и поворота запястья.

Промышленные роботы, работающие в цилиндрических координатах. Робот, работающий в цилиндрических координатах, имеет вертикальное возвратно-поступательное движение для второй степени подвижности или подвижное основание. Обычно это выполняется с помощью пневматического или гидравлического цилиндра, но для роботов больших габаритов может быть использована передача типа колесо—рейка или цепная передача (рис. 6.7). Оси *A*, *B* и *C* робота, работающего в цилиндрических координатах, образуют рабочее пространство в форме цилиндра. Роботы такого типа обычно имеют возвратно-поступательное движение по третьей степени подвижности аналогично роботам, работающим в полярных координатах. У роботов, работающих в ци-



Рис. 6.5
Многозвечный робот типа «хобот»

линдрических координатах, можно получить движение основания и поворот плечевого сустава, придавая им дополнительную степень подвижности между основанием и запястьем. Такие роботы имеют восемь степеней подвижности, но такие компоновки считаются редкими.

Проблема подъема детали или инструмента по вертикали легко решается у роботов, работающих в цилиндрических координатах. Для этого требуется только одна степень подвижности вместо трех осей движения, необходимых роботам, работающим в полярных координатах, или роботам с многозвечной рукой. Недостатком роботов, работающих в цилиндрических координатах, является то, что они не могут огибать препятствия.

Роботы, работающие в декартовой системе координат. Работчим, имеющим дело со станками, удобнее использовать роботы, работающие в декартовой системе координат. Чаще всего такие роботы устанавливаются на порталах, но могут быть смонтированы на рельсах, установленных на полу. Первые три оси роботов, работающих в декартовой системе координат — это оси X ; Y и Z металлорежущих станков. Преимуществом описываемого типа роботов является их жесткость, обеспечиваемая за счет коробчатой конструкции рамы портала. Небольшое количество узлов

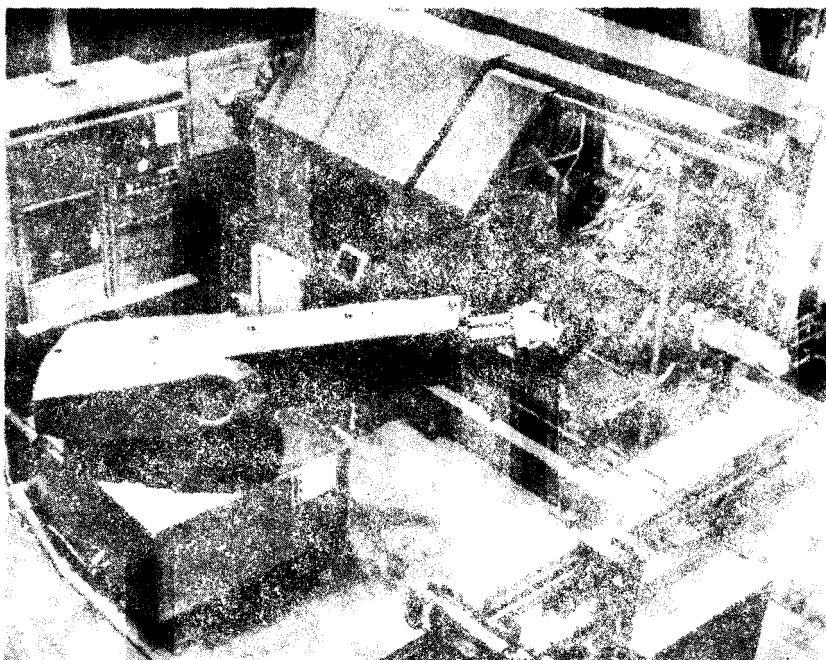


Рис. 6.6
Робот, работающий в полярных координатах

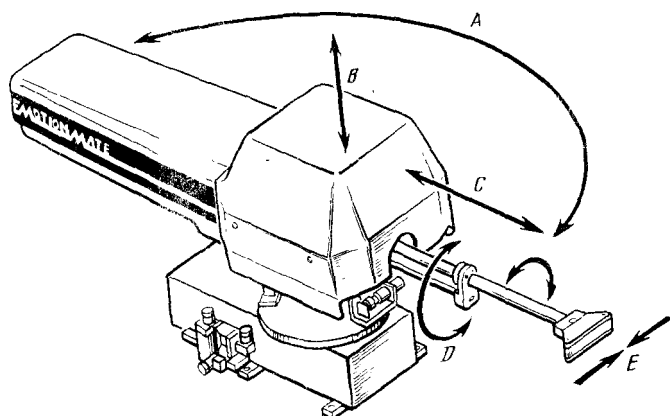


Рис. 6.7
Робот, работающий в цилиндрических координатах:
A — вращение основания; B — подъем робота; C — выдвижение руки; D — вращение;
E — захват

Рис. 6.8

Схема для расчета положения центральной точки инструмента

робота перемещается на консоли, это самые легкие узлы, близкие к рабочему органу. Маленькие роботы, работающие в декартовых координатах, могут обеспечить более жесткие допуски, а для очень больших роботов такая конструкция становится обязательной.

Рабочее пространство.

Зная степени подвижности робота и его конструктивные особенности, связанные с осуществлением этих степеней подвижности, пользователь, применяя геометрию и тригонометрию, может определить положение центральной точки инструмента в зависимости от различного осевого положения частей робота. На рис. 6.8 приведен расчет положения центральной точки инструмента как функции положения звеньев робота. Обратите внимание на то, что углы A , B и C измеряются от прямых, параллельных оси X . Углы A и B имеют направление против часовой стрелки (положительные углы), а угол C — по часовой стрелке и поэтому считается отрицательным. Синус отрицательного угла отрицателен, поэтому выражение $c \sin C$ в формуле для определения Y отрицательно. Изучение диаграммы движения звеньев робота, показанной на рис. 6.8, подтверждает справедливость этого положения.

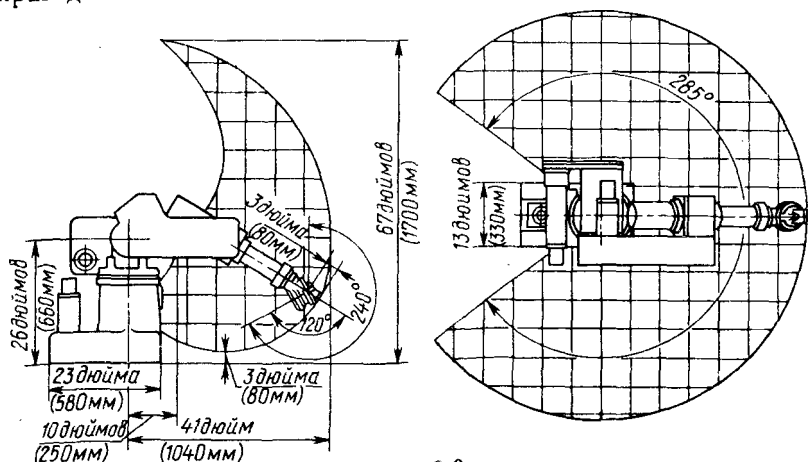
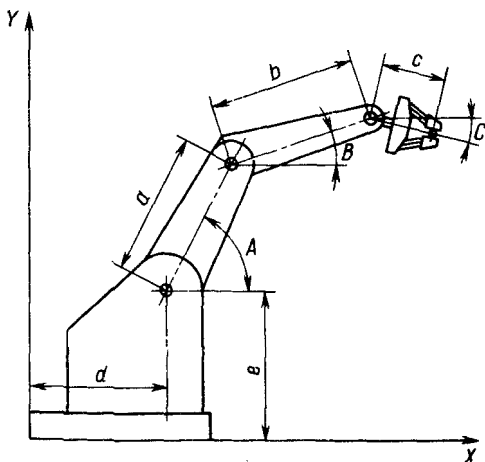


Рис. 6.9

Пример рабочего объема промышленного робота

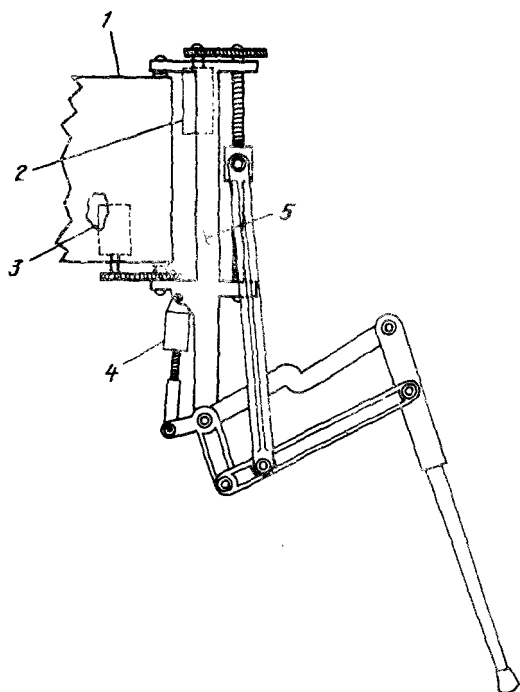


Рис. 6.10
Конструкция ноги шагающего робота:

1 — несущая конструкция; 2 — двигатель вертикального перемещения; 3 — двигатель поворота ноги; 4 — двигатель выдвижения ноги; 5 — каркас ноги

Крайние положения осей робота определяют границу зоны, в которой робот может функционировать. Эти границы очерчивают пространство, что в промышленности называется рабочим объемом. На рис. 6.9 показан пример рабочего пространства. Рабочее пространство является весьма важной характеристикой, которую необходимо учитывать при каждой закупке робота. Размер рабочего пространства опреде-

ляет пределы досягаемости при работе робота, что важно с точки зрения эксплуатации. Однако он также важен и с точки зрения обеспечения безопасности. Мощный робот, помещенный в пространство, габариты которого меньше габаритов рабочего пространства, скоро произведет разрушения, несмотря на горделивые заверения и профессиональную предусмотрительность программистов. При этом несоответствии авария неизбежна, даже в гипотетическом случае идеального безотказного робота, поскольку при программировании следует учитывать слишком много переменных, а при пользовании — слишком много дополнительных условий.

Подвижные роботы. Выше дано понятие рабочего пространства, применяемое для большинства промышленных роботов, поскольку они устанавливаются либо на неподвижном основании, либо на каретке, имеющей ограниченную возможность перемещаться по основанию. Однако некоторые роботы могут перемещаться на ногах или на колесах. На рис. 6.10 изображен робот, который может перемещаться со скоростью спешащего человека. Он может поднять груз массой 1 т и ходить с грузом массой 0,5 т. Этот робот, демонстрируя свои возможности, не только залез в кузов пикапа, но и, приподняв его за задок и шагая, повернул на 90°.

Шагающие роботы являются не просто курьезами или игрушками для развлечения. Необходимость в их использовании имеется во многих отраслях промышленности. Примерами могут служить подземные разработки, космические исследования, аварийные работы на объектах, связанных с атомной энергетикой, взрывные работы, обследование мест пожаров, исследование морского дна и служба охраны. Шагающие роботы сложнее, чем колесные модели, но они более приспособлены к перемещению по труднопроходимым неровным поверхностям. Колесные варианты имеют свои преимущества в случае использования их в качестве автоматических транспортных систем или устройств для доставки почты в учреждениях. Чтобы подчеркнуть полезность робота, изображенного на рис. 6.12, его создатели (фирма «Одетикс») в противовес фантастической внешности окрестили его «функционоид» [60].

6.2. ПРИВОДЫ РОБОТОВ

Наиболее яркой чертой, характеризующей промышленный робот, является привод робота. Он обычно определяет диапазон рабочих характеристик робота, а тем самым и возможности применения роботов, хотя различные типы роботов могут иногда применяться по одному назначению. Ниже рассмотрены четыре основных типа роботов.

Роботы с гидроприводом. По физическим показателям наиболее мощными являются роботы с гидроприводами. В таких роботах значительные усилия могут быть приложены непосредственно к местам сочленений и к захвату или к центральной точке инструмента. Вместе с тем стоимость роботов с гидроприводом выше, чем стоимость роботов с электро- или пневмоприводом, обладающих теми же эксплуатационными характеристиками. Кроме того, для роботов с гидроприводом требуется насос, сосуд для рабочей жидкости, клапаны и гидроаппаратура, рассчитанная на высокое давление. На рис. 6.11 показан робот с гидроприводом грузоподъемностью 225 фунтов (190 кг).

Важной областью применения роботов с гидроприводом являются окрасочные операции. В связи с повышенной пожароопасностью участков окраски распылением для них могут потребоваться роботы, отвечающие требованиям стандартов национальной ассоциации по защите от пожаров (NEPA), класс I, разд. 1, «Воспламеняющиеся среды». Этим требованиям соответствуют, за малым исключением, только роботы с гидроприводом.

Роботы с пневмоприводом. Роботы с пневмоприводом являются одними из наименее дорогих и наиболее практичных для операций типа «взять—установить» или для загрузки-разгрузки станков. Очевидным преимуществом является простота обеспечения сжатым воздухом (давлением около 90 фунтов на кв. дюйм (620 кПа)). На большинстве предприятий имеются разведенные по всем производственным помещениям магистрали сжатого воз-

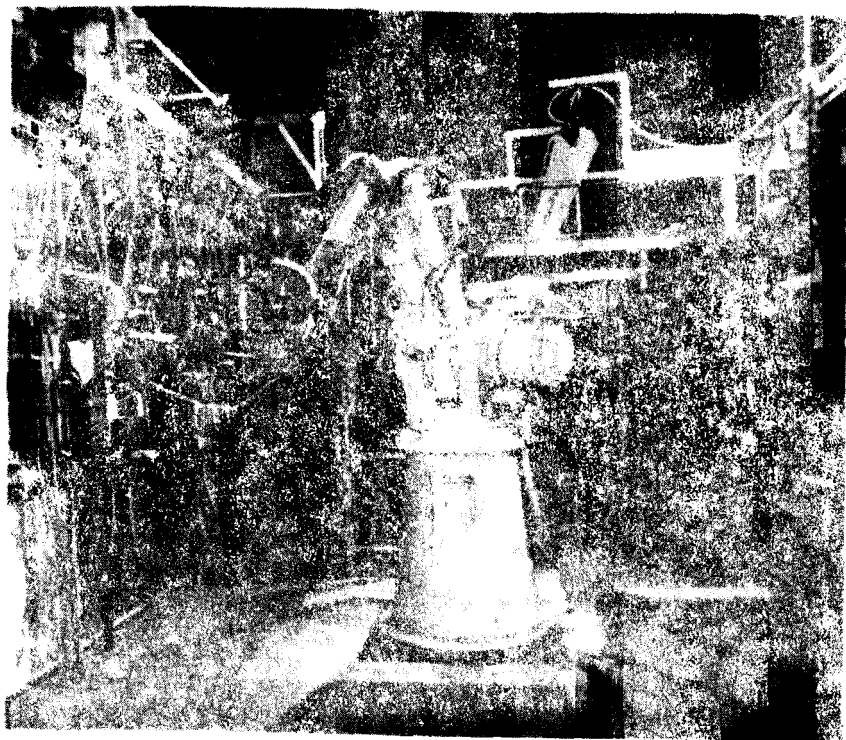


Рис. 6.11
Гидравлический робот

духа, к которым может быть присоединен робот с пневмоприводом. На рис. 6.12 изображен такой робот, работающий в цилиндрических координатах, грузоподъемностью до 5 фунтов (22 кг).

Роботы с пневмоприводом обычно работают по упорам на каждой из осей. Эти упоры ничем не отличаются от тех, что использовались в пневмоприводах до появления роботов. В действительности, робот с пневмоприводом является конструкцией, состоящей из нескольких пневмоцилиндров, каждый из которых осуществляет движение по одной из осей.

(При управлении движением по осям только в конечных точках у читателя может возникнуть вопрос, что же программируется у робота с пневмоприводом. Вспомните, однако, что время цикла и последовательность движения являются важными характеристиками, изменение которых создает бесконечное количество вариантов программ для управления роботом с пневмоприводом, даже без применения гаечного ключа. Если прибавить сюда наладку упоров, количество вариантов еще возрастет. Следовательно, непрерывно управляемого, постоянно изменяемого движения обычный робот с пневмоприводом обеспечить не может.

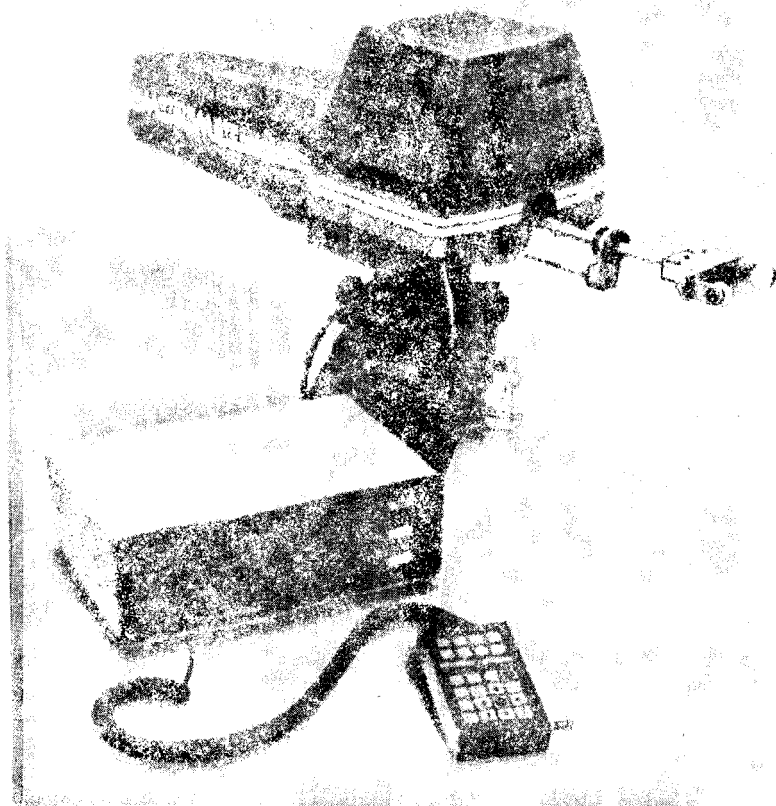


Рис. 6.12
Пневматический робот

Есть, все же, один тип робота с пневмоприводом, конечно не типичный по конструкции, позволяющий получить непрерывное управляемое движение посредством приема, известного как дифференциальная осцилляция. При этом подается ряд коротких импульсов сжатого воздуха, заставляющих рабочий орган робота перемещаться в заданном направлении, проходя весь путь под непрерывным контролем. На рис. 6.13 показан образец выпускаемого промышленностью недорогого робота с пневмоприводом, использующего прием дифференциальной осцилляции для получения контролируемого движения.

Одним из принципиальных преимуществ роботов с пневмоприводом является модульность их конструкции наряду с возможностью использования стандартных покупных компонентов. Это относится и к другим роботам, но особенно ярко проявляется у роботов с пневмоприводами. Эта особенность открывает воз-

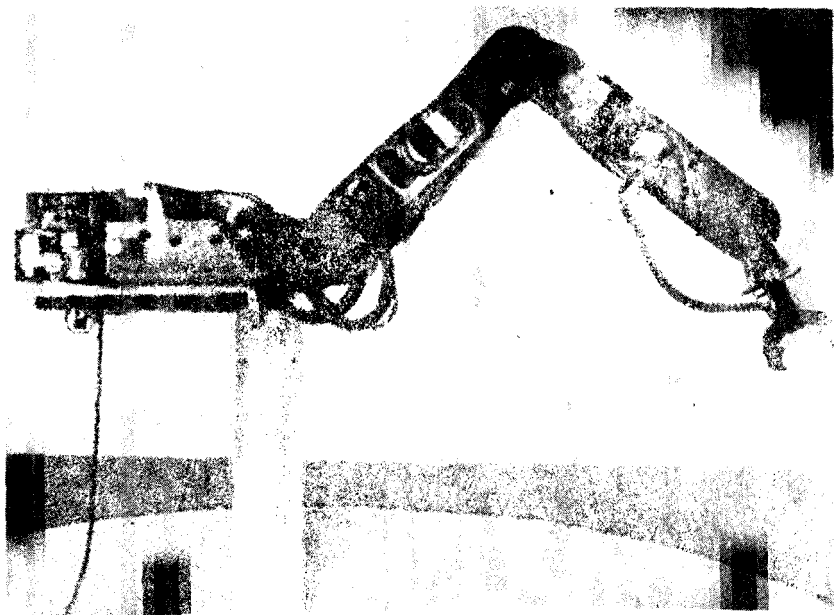


Рис. 6.13

Пневматический робот с непрерывным управлением по пути за счет дифференциальной осцилляции

возможность для деятельности тех фирм, которые решили создать свой собственный робот, подчас со значительной экономией средств. Некоторые поставщики компонентов для роботов рекламируют концепцию «сделай сам» при продаже своей продукции. Та фирма, которая решила действовать в направлении «сделай сам», чтобы сэкономить на стоимости оборудования, должна также учитывать дополнительную стоимость проектирования и затраты на приобретение компонентов.

Роботы с электрическим приводом. Роботы с электроприводом широко применяются для точных работ потому, что они позволяют обеспечить точное перемещение и способны к обучению сложным операциям. Можно возразить, что многие роботы с гидроприводом имеют подобные характеристики, однако наиболее сложные движения являются типичными именно для полностью электрических моделей роботов.

Роботы с электроприводом можно разделить на две группы соответственно типам электрических двигателей, обеспечивающих перемещение по осям. В первой группе используются шаговые двигатели (см. гл. 2), которые осуществляют точное угловое перемещение под воздействием единичных импульсов напряжения, генерируемых системой управления приводами. Перемещение ротора шагового двигателя может быть очень точным, если момент



Рис. 6.14

Робот, оснащенный электродвигателями постоянного тока

нагрузки не превышает предельный момент, установленный для данного двигателя. Из-за этой точности роботы с шаговым приводом иногда имеют разомкнутую систему управления, т. е. устройство ЧПУ вычисляет количество импульсов, необходимое для осуществления заданного движения, и передает команду роботу, не проверяя, совершил ли он это движение. К сожалению, из-за каких-либо препятствий или по другим причинам в механической передаче робота может произойти проскальзывание, поэтому робот не всегда полностью выполняет заданное движение. Когда это происходит, робот с разомкнутой обратной связью теряет управление, и устройство ЧПУ не может определить действительного положения его подвижных частей. При этом робот продолжает следующие циклы с постоянной ошибкой по положению, что может сделать операцию полностью бесполезной или даже разрушительной. Выход из неприятного положения будет описан позже.

Другой разновидностью роботов с электроприводом являются роботы с серводвигателями постоянного тока. Они обязательно имеют обратную связь между приводом и подвижной частью. При этом система управления постоянно отслеживает положение управляемых частей робота, сравнивает положение с заданным и замечает рассогласование или создавшуюся ошибочную ситуацию. На двигатели подается постоянный ток, который вызывает коррекцию рассогласования, пока оно не сведется к нулю.

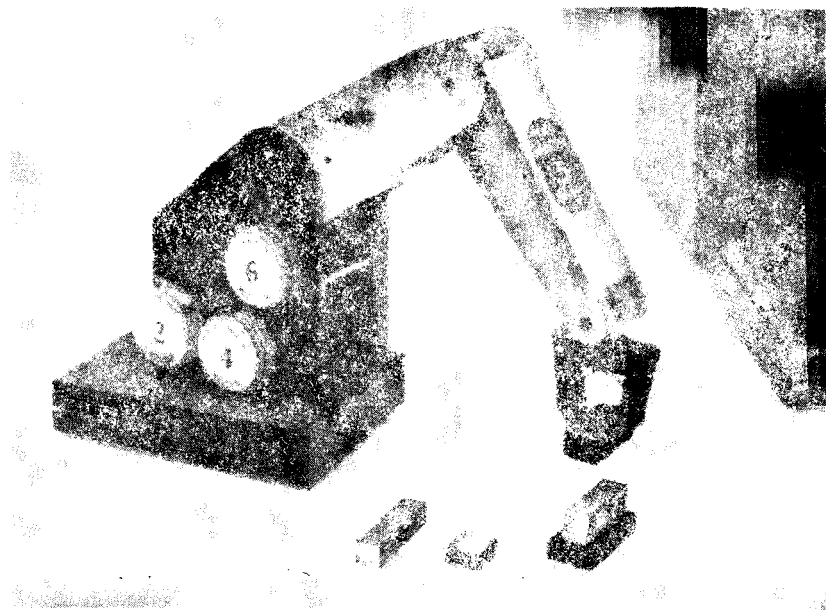


Рис. 6.15
Робот, оснащенный шаговыми двигателями

Цепи обратной связи могут быть реализованы также и на роботах с двигателями постоянного тока. Для отслеживания действительного углового положения подвижных частей могут быть использованы оптические датчики. Эта информация поступает в устройство ЧПУ, которое запрограммировано на принятие мер по корректировке любых рассогласований.

Поскольку этот прием может быть использован как при разомкнутой, так и при замкнутой обратной связи, можно ожидать, что роботы с шаговым приводом будут доминировать над роботами с двигателями постоянного тока. Следует, однако, учитывать, что последние имеют преимущества аналогового устройства, позволяющие им обеспечивать более плавное и постоянно контролируемое движение.

Среди двух основных типов электрических приводов роботов наиболее популярны роботы с электродвигателями постоянного тока. На рис. 6.14 и 6.15 показаны соответственно робот с серводвигателями постоянного тока и робот с шаговыми двигателями.

Приводы с кулачками и зубчатыми передачами. Для полноты изложения опишем манипуляторы, приводимые зубчатыми передачами и кулачками. Этот тип манипуляторов имеет жесткую программу, заложенную при изготовлении, и не отвечает понятию «программируемый», фигурирующему в большинстве определений понятия «промышленный робот». Если проследить всю кинематическую цепь до источника движения, то манипуляторы

следует считать устройствами с электрическими приводами. Однако энергия передается к подвижному узлам через сложную кинематическую цепь. У манипуляторов, приводимых зубчатыми передачами и кулачками, есть два основных преимущества: низкая стоимость и высокая скорость.

6.3. КОНТРОЛЬ ПЕРЕМЕЩЕНИЙ

Наиболее очевидной отличительной чертой робота является тип используемого привода, однако таким же важным, хотя и более тонким показателем, является степень контролируемости его перемещений. Вид контроля зависит от выбора привода робота, как было показано ранее, но не в полной мере определяется этим обстоятельством. Пользователям роботов необходимо знать, в какой степени надо контролировать движение в зависимости от области применения робота, поскольку от этого зависит его стоимость. Ниже будет рассказано о трех категориях контроля движения в порядке возрастания сложности контроля.

Контроль конечных положений. Самым упрощенным и поэтому самым дешевым видом управления движениями робота является контроль конечных положений. Этот тип управления иногда называют двухпозиционным контролем, поскольку робот обычно имеет по два крайних положения на каждой оси. При описании роботов с пневматическим приводом было отмечено, что эти крайние положения обычно определяются механически переставляемыми упорами.

Пользователи роботов с концевыми упорами практически не могут выбирать скорость перемещения объектов, поскольку они не могут повлиять на эту скорость. В некоторой степени изменять скорость, все же, возможно за счет изменения параметров источника питания. Для обеспечения требуемой длительности цикла можно программировать паузы между движениями робота. Типичной областью использования роботов с концевыми выключателями являются загрузка и разгрузка станков. Роботы с концевыми упорами всегда имеют пневматический или гидравлический привод.

Управление от точки к точке. Несколько лучшим по отношению к контролю конечных положений является управление от точки к точке. В этом режиме пользователь может выбрать любую точку в пространстве рабочей зоны робота и получить перемещение непосредственно к этой точке. Траектория движения и его скорость на пути к месту назначения чаще всего не поддаются регулированию. Даже если скорость регулируется, управление считается выполняемым «от точки к точке», если траектория перемещений не контролируется. Такой вид контроля хорош для установки изделий, сверления отверстий, точечной сварки и грубых сборочных операций. Загрузка и разгрузка станков с использованием поддонов или палет также требует управления от точки к точке.

Перемещение от точки к точке не следует путать с прямолинейным перемещением. Даже простое прямолинейное движение между двумя точками не может быть выполнено роботом с управлением от точки к точке. Единственным исключением из этого правила является прямолинейный вертикальный подъем робота, работающего в цилиндрических координатах. Однако прямолинейные движения — непростая задача для роботов, особенно с многозвенной рукой. Для осуществления прямолинейного движения роботам с многозвенной рукой всегда требуется выполнять одновременные контролируемые движения более чем по одной оси. В противовес этому недостатку изготовители роботов разработали программное обеспечение, которое берет на себя распределение напряжений, импульсов или движений клапанов по осям движений для получения прямолинейных движений.

Контурное управление. Как и для станков с ЧПУ, наиболее сложной категорией управления роботами является контурное управление. При контурном управлении движение контролируется постоянно. Когда приводом служит шаговый двигатель, управление является не совсем контурным, но может рассматриваться как приближение к нему при наличии цепи обратной связи и способности системы управления варьировать частоту импульсов, подаваемых на шаговый двигатель.

Разницу между управлением от точки к точке и контурным трудно уловить, особенно для того, кто никогда в реальности не программировал робот. Благодаря детальному и скрупулезному программированию программист может заставить робот с управлением от точки к точке двигаться по непрерывной, на глаз, траектории. Иногда программное обеспечение позволяет освободить программиста от описания мириадом точек, требующихся для аппроксимации криволинейной траектории множеством небольших прямолинейных перемещений. Однако программирование не является здесь единственной проблемой. Время выполнения программ для робота с управлением от точки к точке зависит от количества указанных точек и поэтому может оказаться слишком большим, что отрицательно скажется на эффективности робота. При контурном управлении контролируется не только положение органов робота, но также скорость перемещения инструмента по каждой управляемой оси. Контурное управление роботами требуется при большинстве операций окраски распылением, полировании, склеивании и дуговой сварке.

Режим слежения за конвейером. Одной из наиболее сложных разновидностей контурного управления является так называемый «режим слежения за конвейером», который заключается в выполнении операций на непрерывно движущемся конвейере. Этот режим является, конечно, только развитием контурного управления, а не отдельной категорией контроля. Однако сложность задачи слежения за конвейером определяется сложностью программирования системы ЧПУ робота, особенно для роботов на неподвиж-

ном основании, составляющих большинство. Некоторые роботы, спроектированные специально для работы с конвейером, имеют для первой степени подвижности горизонтальную траверсу вдоль направления движения конвейера. Движение по траверсе может быть согласовано со скоростью конвейера, что дает такому роботу явное преимущество в отношении программирования перед роботами с неподвижным основанием. В то же время роботы с неподвижным основанием могут оснащаться базовым программным обеспечением, позволяющим в случае необходимости без труда модифицировать программу пользователя для условий слежения за конвейером. Следовательно, критерием выбора между двумя типами роботов будет сравнение стоимости оборудования и программного обеспечения.

Режим слежения за конвейером имеет очевидные преимущества. При обработке продукции можно использовать непрерывный конвейер вместо прерывистого. Механическая часть непрерывного конвейера значительно проще и поэтому дешевле и надежнее. При более надежном конвейере и меньшем времени восстановления непрерывный конвейер имеет меньшее время простоев и большую загрузку станка и робота.

Особенно подходит для режима слежения за конвейером робот, выполняющий окраску распылением. Обычно краска накладывается на изделие, которое при этом перемещается непрерывным подвесным цепным конвейером. Робот должен окрасить изделие со всех сторон; эта задача проще всего может быть выполнена при непрерывном движении изделия через рабочую позицию и возможности работы в режиме слежения за конвейером. Этот режим удобен, если на каждом изделии робот должен выполнить много операций.

6.4. ОСНАЩЕНИЕ РОБОТА

Новичок в области робототехники может изумиться, обнаружив, что он приобрел робот, который ничего не может делать! Хотя в этой главе часто упоминались захваты робота, большинство роботов при поставке не оснащаются такими устройствами. Памятуя о том, что программируемость и универсальность являются отличительными чертами современного промышленного робота, читатель сообразит, что стратегией изготовителей роботов является предоставление пользователю права выбора окончательного оснащения. Действительно, многие роботы запрограммированы на использование различных инструментов или захватов в одной наладке, автоматически выбирая и меняя инструменты в соответствии с заданной последовательностью.

Захваты. Номенклатура захватов очень широка, и они часто конструируются заказчиком применительно к конкретной задаче. Большинство захватов зажимаются, чтобы захватить деталь, но немало и таких, которые требуют введения их пальцев в деталь,

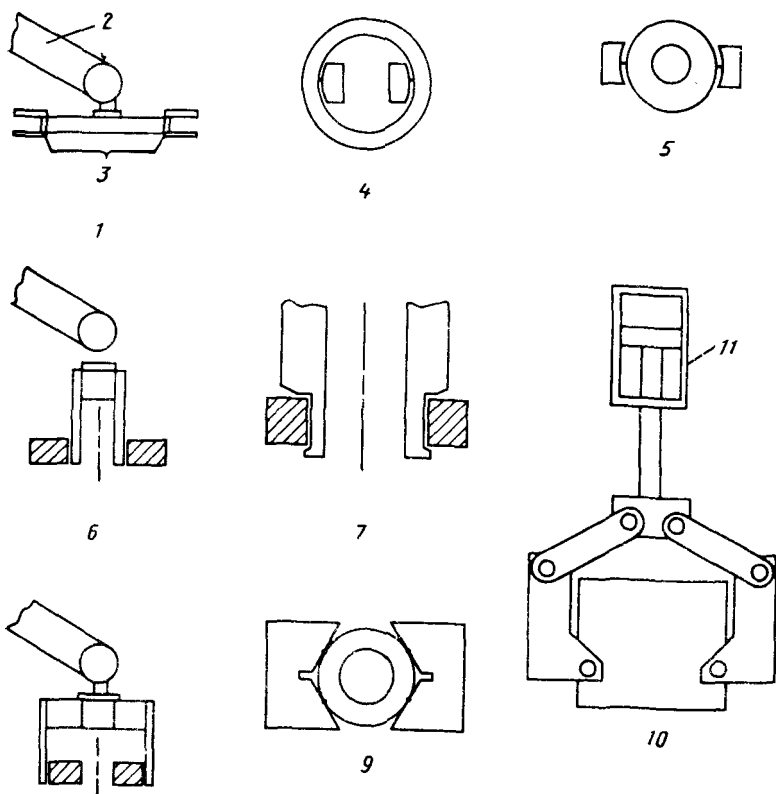


Рис. 6.16

Виды захватов промышленных роботов:

1 — двойной захват; 2 — захваты; 3 — рука робота; 4 — двухточечный захват изнутри; 5 — двухточечный захват снаружи; 6 — захват по внутреннему диаметру; 7 — захват с блокировкой; 8 — захват по наружному диаметру; 9 — четырехточечный захват с помощью призматических блоков; 10 — коленчатый рычажный механизм; 11 — пневмоцилиндр

чтобы захватить ее, разжимаясь. Многие захваты могут эффективно работать в обе стороны, так что выбор остается за программистом. На рис. 6.16 показан ряд захватов.

Во многих случаях двойные захваты более эффективны, чем одинарные. Это утверждение чаще всего справедливо для операции загрузки-разгрузки станков, когда благодаря двойному захвату робот может снять заготовку и установить новую, не перемещаясь между позициями. Это экономит много времени, которое в противном случае было бы потрачено на повторное движение по основной оси (обычно вращения основания). Вращение основания, как правило, является самым медленным движением робота.

Принадлежности. Кроме обычных захватов для деталей, рабочий орган робота может быть оснащен инструментальными го-

Рис. 6.17

Функционирование компенсирующей муфты в случае несоосности вала и отверстия:

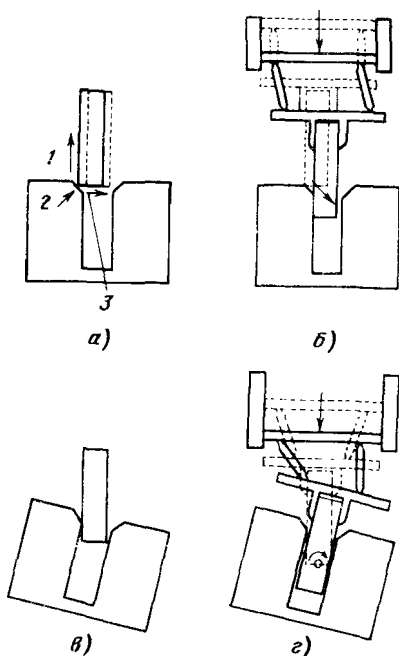
1 — осевое усилие; 2 — реакция контакта; 3 — радиальное усилие.

ловками различного типа. Широкое разнообразие инструментального оснащения выделяет роботы из группы обычных устройств для манипулирования материалами.

После захватов наиболее распространенным оснащением роботов являются сварочные головки. Обычно это головки для точечной сварки, но необходимость в применении роботов для дуговой сварки постоянно возрастает, что будет показано при описании областей применения. Краскопульты для окрашивания распылением являются важной разновидностью оснащения роботов. Подобными являются устройства для нанесения клея. Использование этих устройств стало возможным благодаря высокой точности и повторяемости движений робота. Необычной оснасткой является устройство для разводки электрических проводов. Оно используется при программируемой сборке электрических щитов самолетов и другого крупного оборудования.

Самоустанавливающаяся оснастка. Малейшее отклонение от правильного относительного положения детали или инструмента—робота может привести к полной остановке процесса и, вероятно, к повреждению изделия или руки робота. Чтобы избежать или по крайней мере смягчить последствия неправильного относительного положения можно применить следующий прием: снабдить захват гибким элементом, который позволяет захвату робота слегка смещаться, когда он наталкивается на захватываемый объект. В промышленности такой подход называется самоустанавливающейся оснасткой.

Одним из сложных видов самоустанавливающейся оснастки является компенсирующая муфта. Ее работа показана на рис. 6. 17. На рис. 6.17, а робот пытается вставить ось в отверстие, но ему мешает несоосность. Несколько помогает наличие фаски, но ось, вставляемая с помощью жесткой оснастки, может не попасть в отверстие. Даже использование более податливой оснастки может не дать положительных результатов, потому что радиальная составляющая силы контакта с фаской приводит к повороту встав-



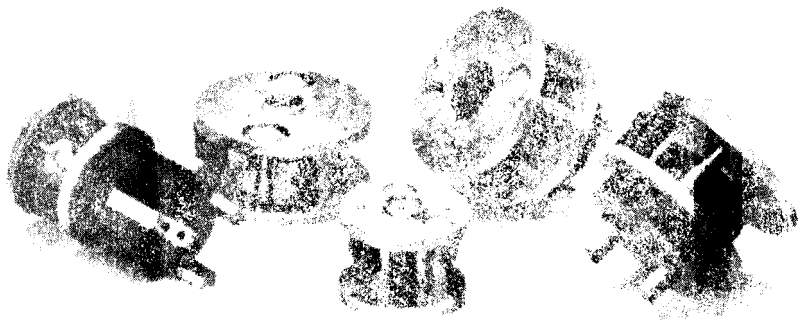


Рис. 6.18

Упругие муфты, облегчающие использование роботов для точных сборочных операций

ляемой оси вокруг центра жесткости. В конструкции, показанной на рис. 6.17, б, центр жесткости проектируется на входящий в паз конец оси, который является как бы вынесенным центром жесткости. Важным следствием этого является то, что ось сдвигается в радиальном направлении вместо поворота вокруг точки контакта.

На рис. 6.17, в представлено положение, когда рассогласования между осью и отверстием в радиальном направлении нет, зато есть рассогласование по углу. Обратите внимание, что в этом случае контакт оси с отверстием происходит в двух точках. Радиальные составляющие контактных сил параллельны и направлены навстречу, но эти силы действуют не по одной прямой: сила слева воздействует на точку оси, находящуюся выше, чем сила справа. Создается момент сил, который действует на ось и на упругую муфту, показанную на рис. 6.17, г, что позволяет оси войти в отверстие. Внешний вид упругих муфт показан на рис. 6.18.

Более простой прием, срабатывающий в некоторых случаях, заключается в использовании резины или нейлона в конструкциях пальцев захвата, что позволяет обеспечить мягкое, податливое удержание объекта, напоминающее действие человеческих пальцев. Любая оснастка, обеспечивающая гибкость захвата, может рассматриваться как самоустанавливающаяся. Изложенный принцип очень прост и кажется совершенно очевидным для решения проблемы обеспечения захвата, однако удивительно часто применение роботов оказывается неудачным из-за того, что пользователю или инженеру не приходит в голову этот принцип применить.

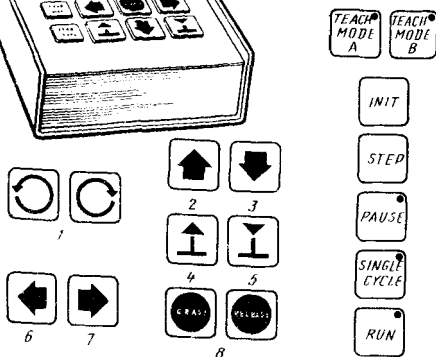
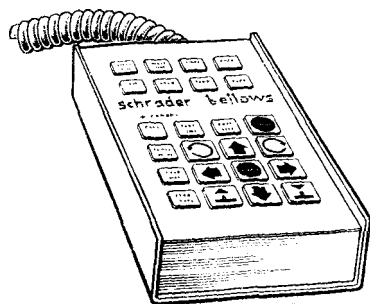
6.5. ПРОГРАММИРОВАНИЕ

Главной особенностью робота является его способность к перепрограммированию для выполнения различных задач. Эта черта отсутствовала у механических манипуляторов, которые можно

Рис. 6.19

Панель обучения промышленного робота. Функциональное назначение клавиши:

1 — поворот запястья (на 90° или 180°); 2 — выдвинуть руку; 3 — убрать руку; 4 — подъем; 5 — опускание; 6 — поворот налево; 7 — поворот направо; 8 — управление захватом (зажать, разжать)



было видеть и до появления роботов в 70-х годах. В гл. 7 программирование роботов рассмотрено более детально, но основная классификация приведена здесь.

Для программирования некоторых роботов необходимо вручную вставлять в барабан штыри, которые нажимают на электрические выключатели в заданной последовательности. Другие роботы программируются на диалекте какого-либо распространенного языка, например языка Бейсик. Одним из таких диалектов является АРМБейсик — язык для роботов, названный так фирмой «Микробот», его создателем и владельцем авторских прав [42]. Другим языком программирования для роботов является ВАЛ, разработанный фирмой «Юнимейшн» для сложных роботов модели ПУМА [73].

Подавляющее большинство роботов оснащаются какой-либо разновидностью портативного устройства управления, называемого панелью обучения. Оператор заставляет робота совершать нужные движения, нажимая на кнопки управления, расположенные на панели. Темп перемещения обычно выбирается значительно ниже, чем в производственных условиях, что позволяет оператору тщательно подготовить программу, которая автоматически записывается и воспроизводится по команде с рабочей скоростью. Панель обучения может также применяться для ознакомления с действиями робота и для установки робота в требуемое положение без запоминания программы.

На рис. 6.19 показаны панель обучения промышленного робота и в увеличенном масштабе имеющиеся на ней клавиши. Работа возможна в двух режимах. Режим обучения А предполагает автоматическое задание времени отработки команд и порядка поступления внешних команд во время обучения робота. Режим обу-

чения *Б* предполагает две стадии обучения. В первой стадии программируется только последовательность выполнения действий. Во второй стадии в полученную последовательность включаются время отработки и внешние команды. В режиме *Б* можно также изменить время выполнения любой команды, запрограммированной в режимах *А* и *Б*.

Запираемый на ключ выключатель, имеющийся на всех панелях, блокирует верхний ряд функциональных клавиш. Это исключает возможность случайного перехода в режимы обучения.

Кроме клавиш выбора режимов обучения, на панели имеются клавиши:

INIT — возврат в исходное положение; STEP — отсчет времени движения, повторное нажатие останавливает движение и начинает отсчет времени следующего движения; PAUSE — остановка движения робота в режиме отсчета времени; SINGLE CYCLE — выполнение одного полного запрограммированного цикла; RUN — непрерывное выполнение запрограммированного цикла.

Панель обучения робота очень похожа на панель управления мостовым краном. Однако панель обучения значительно сложнее, потому что она имеет дополнительные элементы управления, приказывающие роботу запомнить точки по пути движения. Кроме того, система управления роботом может задействовать таймеры для синхронизации операций, воспринимать команды извне и выдавать выходные команды на периферийное оборудование.

На высшей ступени развития находятся роботы, обучаемые при помощи непосредственного воздействия на их рабочие органы, когда обучающий берет инструмент в руку и вхолостую выполняет всю последовательность производственных процессов. Робот получает команду запомнить всю траекторию в процессе обучения, а затем точно ее воспроизводить. Этот тип обучения особенно хорош для окраски распылением и сварочных операций. Оператор, имеющий опыт в обычной окраске распылением или сварке, может поделиться с роботом своим умением, выполняя работу, как обычно, вручную. Роботу затем остается только воспроизвести действие учителя. Конечно, опыт переданный роботу, приложим только в данном конкретном случае, а не в окраске или сварке вообще.

Имеется вариант описанного метода обучения, при котором применяется механизм, имитирующий робот: рука для обучения, иногда называемая макетом робота. По сравнению с настоящим роботом рука для обучения (рис. 6.20) в целом легче, и ею проще манипулировать опытному оператору, отвечающему за обучение робота. Окраска распылением является идеальным полем деятельности для таких механизмов, потому что опытный оператор должен во время обучения иметь ощущение, как будто он действительно держит в руках краскопульт. Сравнительно легкая рука для обучения может обеспечить это требование.



Рис. 6.20

Механическая рука для обучения робота непосредственной имитацией движений в режиме обучения

В режиме обучения рука передает траекторию своего движения на управляющую ЭВМ. В свою очередь управляющая ЭВМ задает настоящему роботу ту же траекторию в режиме воспроизведения. Роботы, предназначенные для самых сложных операций, обычно программируют, используя комбинации режимов обучения, описанных здесь. Так, координаты фиксированных точек можно получить, используя панель обучения, а выполнение сложных траекторий, таких, как дуги или кривые, может быть запрограммировано при помощи программного обеспечения ЭВМ, например на языке ВАЛ фирмы «Юнимейшн». Необходимость в этом станет более очевидной после изучения гл. 7, в которой детально рассмотрены вопросы программирования робота.

6.6. ВОЗМОЖНОСТИ ОЧУВСТВЛЕНИЯ

Механическое манипулирование имеет много областей применения, однако его успешность сильно зависит от позиционирования и ориентации деталей. К тому же, слепое повторение механических операций может оказаться катастрофическим, если что-то произошло не так, как положено. Робот, способный «видеть» или «ощущать» приложенную нагрузку и некоторые условия окру-

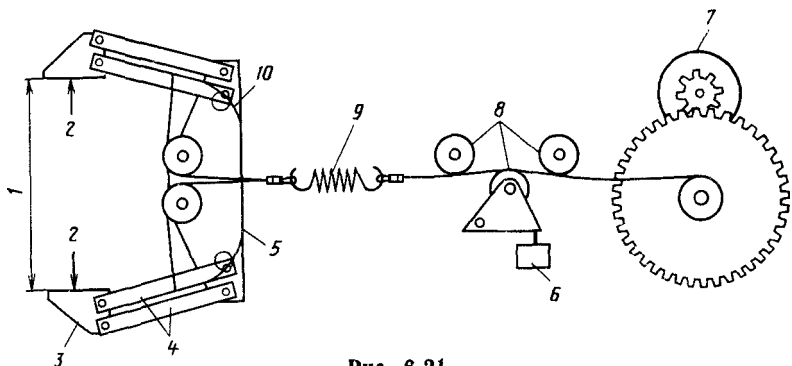


Рис. 6.21

Устройство, использующее «принцип сухожилия». Трос, проходящий в руке робота, позволяет одновременно управлять захватом и регулировать усилие зажима. Обратите внимание на установку конечного выключателя, контролирующего натяжение троса:

1 — раскрытие захвата; 2 — направление усилий; 3 — захват; 4 — рычаги; 5 — корпус; 6 — конечный выключатель для контроля натяжения; 7 — двигатель привода; 8 — натяжные ролики; 9 — пружина растяжения; 10 — пружина кручения

жающей обстановки, может применяться значительно шире, чем его более ограниченные «сородичи», действующие как «глухие», «немые» и «слепые» манипуляторы.

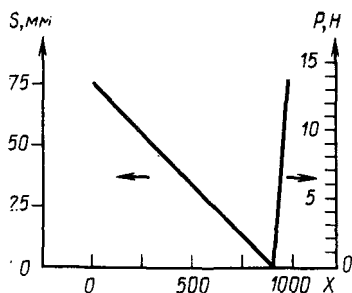
Способность ощущать может быть развита у роботов в различной степени. Разными могут быть и устройства очувствления. Например, к оптическим системам относятся как простые фотоэлементы, так и сложные системы пространственного видения. Ниже описаны различные категории устройств очувствления, начиная с простейших и наиболее практичных и кончая самыми сложными из имеющихся.

Контроль усилия зажима. Вероятно, самой элементарной формой очувствления роботов является придание их захватам способности контролировать усилие зажима между пальцами. В простейшем случае датчик усилия зажима представляет собой обычный конечный выключатель, который срабатывает при достижении установленного значения усилия. Этот выключатель практически является предохранительным устройством против чрезмерного закрытия захвата в случае ошибки в программе или изменения размеров зажимаемой детали. Однако это устройство может использоваться значительно шире, чем только для предохранения, поскольку конечный выключатель может служить также для обеспечения стандартного усилия зажима при измерении. Так, робот может измерять толщину объекта, просто зажимая его в захвате. Эта способность робота обходится недорого, при этом робот легко обгоняет по точности человеческую руку.

Иногда закрытие захвата и обеспечение определенного усилия сливаются в одну операцию. Этого можно достигнуть, например, используя «принцип сухожилия», т. е. приводя захваты в дви-

Рис. 6.22

Графики зависимостей раскрытия захвата S и усилий зажима P от количества импульсов X , обработанных шаговым двигателем механизма, использующего «принцип сухожилия»



жение при помощи тросов, соединенных с двигателем, установленным на основании робота. На рис. 6.21 показано, как одна из фирм-изготовительниц роботов (фирма «Микробот») устанавливает конечный выключатель для контроля натяжения троса, управляющего закрытием захвата. На рис. 6.22 представлены графики зависимостей между раскрытием захвата и количеством шаговых импульсов (левая часть), а также усилием зажима и числом шаговых импульсов (правая часть), иллюстрирующих двоякую роль, которую играет в механизме трос. Способность устройства измерять толщину характеризуется допуском $\pm 1/16$ дюйма (1,6 мм), указанным в документации, однако в действительности робот модели «Микробот Минимумер 5» продемонстрировал значительно лучшие результаты.

Рассмотрим пример расчета количества импульсов, подаваемых на шаговый двигатель для захвата объекта известного размера при определенном усилии сжатия.

Пример 6.1. Расчет количества импульсов. Захват робота приводится в действие шаговым двигателем с использованием «принципа сухожилия» при характеристиках устройства, соответствующих графику на рис. 6.22. Предположим, что захват раскрыт на 2,5 дюйма (63,5 мм), а требуется зажать объект с усилием 20 унций (5,566 Н). Толщина объекта 1,375 дюйма (34,9 мм). Сколько импульсов необходимо подать на двигатель?

Решение. Прямые на графике, изображенном на рис. 6.22, пересекают ось X приблизительно при $N = 895$ импульсов, а полный ход от полностью открытого до полностью закрытого состояния при усилии сжатия $F_1 = 3$ фунта (13,34 Н) составляет $N_2 \approx 976$ импульсов. Чтобы зажать объект толщиной $a = 1,375$ дюйма (34,9 мм) из положения, соответствующего $d_1 = 2,5$ дюйма (63,5 мм), на двигатель необходимо подать следующее количество импульсов: чтобы приблизиться вплотную к объекту

$$N_3 = (d_1 - a) N_1/d_2 = (2,5 - 1,375) 895/3 = 336 \text{ импульсов,}$$

чтобы обеспечить усилие зажатия $F_2 = 20$ унций (5,566 Н)

$$N_4 = F_2 \frac{N_2 - N_1}{16F_1} = 20 \frac{976 - 895}{3} = 34 \text{ импульса}$$

(1 фунт = 16 унциям).

Суммарное количество импульсов

$$N_5 = N_3 + N_4 = 336 + 34 = 370 \text{ импульсов.}$$

Контроль присутствия с помощью оптических средств. Поскольку робот уже получил способность чувствовать усилие зажима, следующим шагом будет снабжение его каким-либо устройством контроля присутствия объекта, обычно фотоэлементом.

Естественное место для установки датчика, определяющего наличие объекта, — захват, однако он ни в коей мере не является единственным местом установки. В действительности датчики обычно на захватах не устанавливаются. В таких случаях единственной физической связью между датчиком и роботом является электрический кабель.

Фотоэлектрические датчики могут быть расположены в различных местах вокруг рабочего объема робота с тем, чтобы они служили для обеспечения безопасности, останавливая робот в случае, если он неожиданно из этого объема выйдет. По меньшей мере один ведущий производитель роботов (фирма «Праб») практикует оснащение каждого нового робота складным ограждением, закрывающим рабочее пространство. Ограждение имеет дверь с электрической блокировкой: при открытой двери робот останавливается.

Фотоэлектрические датчики являются наиболее популярными устройствами контроля присутствия, однако существуют другие типы датчиков. Преимуществом инфракрасных датчиков является помехоустойчивость по отношению к обычному свету. Датчики, работающие в диапазоне радиочастот, должны быть настроены на срабатывание в нужное время и от нужного объекта. Работа этих датчиков зависит от размеров и проводимости объектов, присутствие которых необходимо установить, поэтому нестабильность свойств этих объектов может вызвать некоторые затруднения.

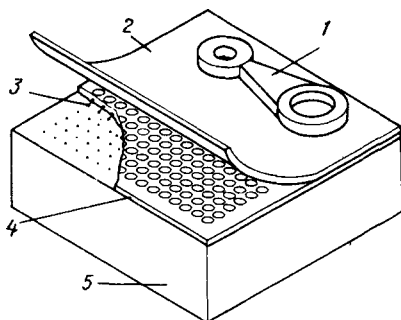
Техническое зрение робота. В области распознавания образов с помощью оптических средств ведутся интенсивные разработки, и некоторые роботы уже способны распознавать объекты на конвейере по их очертаниям. Благодаря этому нужный объект может быть захвачен роботом при любой его ориентации и установлен в требуемом положении на рабочей позиции. Техническое зрение — прекрасная и многообещающая вещь, однако в области робототехники, по крайней мере в двадцатом веке, будут преобладать более привычные фотоэлектрические устройства для определения присутствия объектов.

Тактильные датчики. Чувство осязания человека является удивительным феноменом, и перед учеными еще стоит проблема создания захватов робота, способных действительно «чувствовать» разницу между различными структурами поверхности и формами. Однако В. Д. Хиллис и Д. Холлербек из Массачусетского технологического института уже разработали захват робота, который может «почувствовать» разницу между винтами, шайбами и шпилками. В [24] П. Киннукан рассказывает, что Хиллис и Холлербек использовали «Л' эггсовские свертонки» колготки для изготовления матрицы, состоящей из 256 (16×16) чувствительных к давлению выключателей, которые поставляют информацию для обработки на ЭВМ. Материал колготок был заложен между листом проводящей силиконовой резины и гибкой печатной платой. В нормальном состоянии резина и плата разделены материа-

Рис. 6.23

Электроконтактная решетка, используемая в конструкциях тактильных датчиков:

1 — распознаваемый объект; 2 — пластина из поливинилиденфторида; 3 — электрические контакты; 4 — печатная плата; 5 — заземленный корпус



лом колготок, но в точках нагружения материалы контактируют через сетку материала, создавая матрицу чувствительных точек. Рисунок,

образуемый контактами, анализировался ЭВМ с целью распознавания объекта, взятого роботом. Эксперимент с колготками был предшественником современных осязательных систем, использующих электроконтактные решетки. Для этой цели оказались очень полезными пьезоэлектрические материалы, генерирующие при деформации электрические сигналы. На рис. 6.23 изображена осязающая решетка, изготовленная на основе поливинилиденфторида (PVF_2), имеющего пьезоэлектрические свойства.

Распознавание голоса. Ученые, изобретатели и фантазмечтатели предвидят день, когда люди смогут говорить с промышленными роботами и обучать их, отдавая приказания голосом. Однако это сложная техническая проблема будет решена только в самых дорогих промышленных роботах. Век настоящих роботов R2D2, без сомнения, наступит через несколько десятилетий.

6.7. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

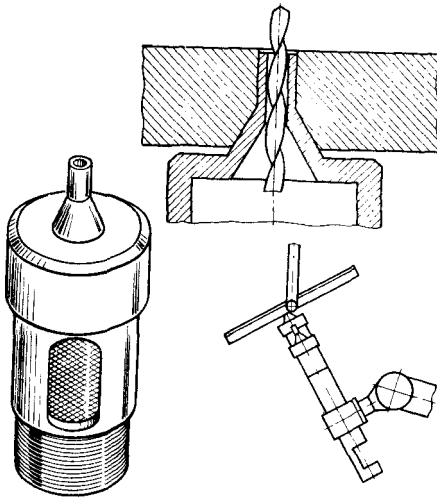
Основные особенности, отличающие дорогие многофункциональные роботы от более дешевых, уже описаны. Есть, однако, ряд общих технических характеристик, которые важно учитывать при выборе робота для данной области применения.

Грузоподъемность. Грузоподъемность является важной характеристикой, однако ее не так просто определить. Большинство роботов способны удерживать значительно большую массу, чем та, с которой они могут поворачиваться с максимальной скоростью. На способность робота эффективно манипулировать объектами влияет также форма объекта и состояние его поверхности. Грузоподъемность робота в случае, если груз находится близко к основанию, очевидно, выше, чем при полном вылете руки.

Некоторые изготовители роботов указывают две грузоподъемности: нормальную и максимальную, статическую и номинальную, или статическую и динамическую. Потенциальный пользователь робота должен точно удостовериться, для каких условий производитель указывает номинальную нагрузку, особенно если приведена только одна цифра.

Рис. 6.24

Зонд с коническим носиком, позволяющий улучшить повторяемость при выполнении операции сверления с помощью робота за счет позиционирования сверла по кондукторной плите



Повторяемость. Читатель должен понять разницу между терминами «точность» и «повторяемость». Точность определяется как способность переместиться в заданную точку пространства, определяемого системой координат X , Y , Z или какой-либо другой. Работа

металлорежущих станков тесно связана с понятием «точность». Повторяемость определяется как способность снова и снова возвращаться в одну и ту же точку, установленную при обучении. Для промышленных роботов повторяемость является более важным понятием, чем точность, потому что обычно первый раз робот обучают с помощью панели обучения. Важной проверкой является проверка способности робота повторять цикл, которому его обучили, не промахиваясь мимо цели. Инженеры по автоматизации иногда бывают несколько обескуражены, обнаружив, что точность позиционирования у промышленных роботов в целом значительно ниже, чем у станков с ЧПУ. Захваты и инструменты робота по большей части закреплены на значительно менее жесткой конструкции, чем это допустимо для металлорежущих станков.

Самым жестким требованиям по повторяемости отвечают маленькие пневматические роботы, позиционируемые механическими упорами. Некоторые из них имеют повторяемость $\pm 0,001$ дюйма ($\pm 0,025$ мм) и лучше. Большие стандартные гидравлические роботы, характерные для автомобилестроения, имеют повторяемость порядка $\pm 0,050$ дюйма ($\pm 1,27$ мм). Роботы для окраски распылением и для сварки могут иметь допуски на позиционирование до $\pm 0,125$ дюйма ($\pm 3,175$ мм) и даже более широкие.

Существуют несколько приемов, позволяющих справиться с проблемами, возникающими из-за недостаточно хорошей повторяемости. На рис. 6.24 показан зонд с коническим носиком, который вставляется роботом в направляющее отверстие для того, чтобы перед началом операции обеспечить точность позиционирования, указанную заказчиком. Позиционирование с помощью зонда осуществляется, если необходимо обеспечить его точность.

Другим методом, особенно полезным в случае применения роботов с шаговыми двигателями, без обратной связи, является периодический возврат в некоторое исходное положение для того, чтобы вставить зонд и обнулить показания регистров, указывающих положение рабочего органа в установленных координатах, восстановив тем самым правильность исходного положения. Пользователь имеет возможность выбрать программу для обнуления в каждом цикле или один раз за несколько циклов в зависимости от требуемой точности и смещений, происходящих за каждый цикл.

Интересно сравнить точность позиционирования, обеспечиваемую роботом и средним человеком. При достаточной внимательности человек способен позиционировать с большей точностью, чем большие промышленные роботы. Однако при постоянном повторении совокупности действий на сохранение внимательности человека при выполнении работы рассчитывать нельзя. Робот же, запрограммированный однажды, будет быстро и стабильно возвращаться в ту же точку в каждом цикле. Хотя люди могут обеспечивать большую точность, они не в состоянии работать с такой же повторяемостью.

Скорость. Скорость является еще одной характеристикой, которая может разочаровать потенциального пользователя робота. Для малых пневматических роботов с концевыми упорами время выполнения простейших циклов загрузки-разгрузки обычно составляет 2—3 с, для некоторых из этих роботов достаточно 1 с, а манипуляторы с механическим кулачковым управлением работают даже еще быстрее. Обычные скорости больших гидравлических роботов составляют 50 дюймов/с (1270 мм/с).

Во многих областях применения робот может оказаться более медлительным, чем его конкурент-человек. Тем не менее использование робота может быть оправдано благодаря большей его производительности, обеспечиваемой работой во время перерывов, в обеденное время и даже ночью.

6.8. ОБЛАСТИ ПРИМЕНЕНИЯ РОБОТОВ И ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЦЕЛЕСОБРАЗНОСТИ ИХ ИСПОЛЬЗОВАНИЯ

Читатель уже в состоянии перечислить некоторые преимущества применения роботов в промышленности. Однако вместо того, чтобы говорить о преимуществах робота перед рабочим, полезно считать, что преимущество имеет человек, достойный более осмысленной работы, чем та, которой обычно занимаются роботы. Эффективность роботов проявляется при выполнении монотонных повторяющихся операций, для которых совсем не требуется или требуется мало ума. Хороши роботы для очень утомительных работ, работ в горячих цехах или в ядовитой опасной атмосфере.

Кроме многочисленных областей применения роботов для монотонных и опасных работ, есть и другие работы, в которых роботы



Рис. 6.25

Робот, заменяя человека в трудновыполнимой для него операции, снимает заусенцы с часто посаженных ребер радиатора, выдерживая точное угловое положение инструмента, несмотря на необходимость перехода от ребра к ребру

имеют преимущества перед человеком. Роботы обеспечивают более высокую повторяемость при выполнении сложных повторяющихся операций. На рис. 6.25 показана сложная операция шлифования. Каждое тонкое ребро необходимо отшлифовать очень точно. Для эффективного выполнения работы должно быть выдержано точное угловое положение инструмента относительно ребра. Человеку очень трудно или даже невозможно сохранить требуемое угловое положение инструмента при переходе к обработке следующего ребра. Требуемую точность в то же время легко получить, запрограммировав робот, в данном случае робот модели «Юнимейт ПУМА».

Следует упомянуть и такие работы, которые требуют перемещения больших тяжестей, непосильных для человека, или настолько утомительны, что человек не выдерживает целую смену. В этом случае применение роботов более чем желательно — оно просто необходимо. Многие гидравлические модели роботов и некоторые электрические имеют грузоподъемность большую, чем человек.

В некоторых областях применения роботы имеют сразу несколько преимуществ. Хорошим примером может служить литье под давлением. Работа в горячих цехах опасна и утомительна, а отливки имеют большую массу. Это же можно сказать и о литейном производстве вообще. В сварочном производстве мы находим комбинацию требований по точности с проблемами обеспечения безопасности. Роботы являются отличными средствами для выполнения как точечной, так и дуговой сварки. Аналогично сварке окраска распылением и другие виды работ, связанные с распылением, представляют собой опасность и требуют точности. Может показаться, что окраска распылением не входит в разряд точных работ, однако при ручной окраске плотность покрытия может меняться от детали к детали, из-за чего они выглядят по-разному. Робот же может постоянно воспроизводить оптимальный процесс окраски.

Сопrotивление со стороны рабочих. Образ сборочных линий, обслуживаемых только роботами, вселяет ужас в умы рабочих, которые верят, что роботы могут заменить их в работе. Такие сборочные линии существуют на отдельных заводах, и роботы, конечно, могут заменить отдельных рабочих на отдельных работах. Однако рассматривать роботы в качестве причины общей безработицы — значит подходить к проблеме не с того конца. При правильном применении роботы повышают производительность, делая фирму более конкурентоспособной, что, наоборот, сохраняет работу. Правильность вышесказанного можно подтвердить на примере автомобилестроительной промышленности. Для автомобилестроителей США было бы безрассудным поступком отказаться от роботов и автоматизации с целью сохранения рабочих мест. Очевидно, что другие страны, выпускающие автомобили, особенно Япония, продолжали бы выпускать автомобили с максимальной производительностью, используя роботы и автоматизацию. Законы рынка скоро привели бы к закрытию менее эффективных неавтоматизированных фирм. Говоря словами Джеймса А. Бакера из фирмы «Дженерал Электрик»: «Генри Форд не изобрел автомобиль, он автоматизировал его производство, создав тем самым миллионы рабочих мест и спинной хребет американской экономики» [7].

Экономическое обоснование. На начало 80-х годов стоимость роботов варьировалась в широком диапазоне, от 3—5 тыс. долл. за самые дешевые до 150 тыс. долл. и выше. Их стоимость сравнима со стоимостью многих металлорежущих станков, за исключением самых дорогих с программным управлением, которые стоят еще дороже.

Легко обосновать применение станков и роботов, если оно жизненно необходимо для осуществления самого технологического процесса. Например, некоторые роботы совершенно необходимы для установки в изделия радиоактивных компонентов.

Экономическое обоснование применения роботов в тех случаях, когда они не необходимы, может вызвать затруднения и требует тщательного анализа стоимостных факторов. Когда робот призван заменить рабочего, имеется тенденция к сравнению суммарных годовых затрат на зарплату рабочего с первоначальной стоимостью робота. Этот подход является чрезмерно упрощенным. Прежде всего, рабочий обычно работает 40 ч и менее в неделю. Сверхурочные работы требуют дополнительных расходов и имеют свои границы. Никакой рабочий не может работать семь дней в неделю по три смены в сутки. Робот тоже не может работать совсем непрерывно, однако ограниченное обслуживание и ремонт позволяет обеспечить работу в этом режиме. Даже при простое робота в течение 8 ч в неделю он работает за четверых рабочих, которым пришлось бы трудиться на этом рабочем месте.

Точность робота и его способность непрерывно повторять заданную последовательность действий приводят к такому снижению себестоимости, которое может оказаться больше, чем снижение стоимости живого труда. Очевидно, что точность, присущая роботам, положительно влияет на качество выпускаемой продукции, что сопровождается сокращением брака и затрат на доработку. Менее очевидной является экономия лакокрасочных материалов, клеев и других материалов, расходуемых роботами при выполнении технологических операций. Рабочий может израсходовать слишком мало краски (проблема качества) или слишком много краски (одновременно проблемы качества и перерасход материалов). Человек имеет склонность расходовать слишком много краски, т. е. совершать более серьезную из двух ошибок.

Менее значимыми экономическими факторами являются дополнительные расходы, связанные с использованием рабочих, такие, как оплата отпусков и больничных, расходы на обеспечение безопасности, содержание комнат отдыха, столовых, парковок для автомобилей и даже оплата отопления и освещения. Имеются сообщения о том, что в Японии существует завод, где работают только одни роботы. Работа происходит в полной темноте, за исключением случаев возникновения отказов. В этом случае ЭВМ дает команду на освещение именно того сектора, который необходим ремонтным службам. Кроме того, он включает одну за другой лампы освещения по пути обслуживающего персонала к месту происшествия.

Вернувшись от фантастики к действительности, рассмотрим экономические недостатки роботов. Как и любое новое закупаемое оборудование, робот требует существенных первоначальных капитальных вложений в счет будущих ежегодных доходов. Большой расход наличных денег может осложнить вопрос о вложении капи-

тала в роботы, даже если уровень ожидаемых доходов высок. Устаревание тоже может быть фактором, хотя и не таким серьезным, поскольку роботы могут быть перепрограммированы. Как показано в гл. 1, оборудование для гибкой автоматизации, включая роботы, не так подвержено устареванию, как специализированное «жесткое» автоматическое оборудование.

Хотя экономика является общим знаменателем всех решений по применению роботов, некоторые критерии настолько трудно выразить количественно, что их необходимо рассмотреть отдельно. Эмерсон [20] рекомендует рассматривать три основных аспекта осуществимости робототехнического проекта: эксплуатационный, технический и экономический. Некоторые руководители настаивают на рассмотрении при обосновании целесообразности внедрения роботов не только экономических факторов. Эксплуатационный и технический аспекты могут помочь убедить руководство принять проект внедрения, даже если о них не спрашивают. Ни технические руководители, ни люди, принимающие окончательное решение, не хотят, чтобы проект внедрения роботов застрял в самом начале из-за недооценки следующих эксплуатационных и технических факторов.

Эксплуатационные факторы: сопротивление внедрению роботов со стороны рабочих; неосуществимость планирования производства; остановка производства во время установки и отладки; необходимость в параллельном ручном производстве во время установки и отладки робота; невозможность использовать ручную работу, чтобы покрыть простои роботов; задержка в сроках поставки заказанного робототехнического оборудования; необходимость обучения операторов; планирование обслуживания; поставка запасных частей; безотказность выполнения работ.

Технические факторы: несовместимость оборудования; несоответствие уровня надежности робота (выше или ниже) уровню надежности обслуживаемого оборудования; недостаточная повторяемость движений робота; проблемы ориентации заготовок; проблемы нестабильности размеров заготовок; излишняя хрупкость заготовок; проблемы взаимной блокировки робота и технологического оборудования; несоответствие рабочего объема робота выполняемой задаче.

Инженеры по автоматизации, которые обращаются к этим эксплуатационным и техническим факторам, представляя проект роботизации на рассмотрение начальству, вселяют в него уверенность, что они свое «домашнее задание» выполнили и осознали возможные последствия принимаемого решения о закупке и установке робота.

В ы в о д ы

Облик настоящего промышленного робота имеет мало общего с чарующим фантастическим образом, созданным средствами массовой информации, в основном кино. Однако реальный робот

имеет свою привлекательность в основном благодаря технологическим преимуществам микроэлектронных схем. Ключевой особенностью промышленных роботов является их программируемость, говоря точнее — перепрограммируемость, ставшая возможной благодаря системам управления на базе микроЭВМ.

В зависимости от конструкции роботы разнообразно реализуют данные им степени свободы. Оси движения робота могут быть вращательными, цилиндрическими или их различными сочетаниями. Наибольшее возможное удаление по осям движения образует границы рабочего объема робота — важной характеристики системы. Основными типами приводов для роботов являются гидро-, пневмо- и электроприводы. Уровень контроля движений роботов является основной характеристикой, определяющей способности робота и одновременно его стоимость.

Обычно роботы не поставляются оснащенными захватами и другими рабочими органами. Большой выбор оснастки, которая может быть закреплена на конце руки робота, является одним из ключевых факторов, обеспечивающих широкую универсальность роботов. Другим подобным фактором является простота, с которой робот может быть запрограммирован при помощи панели обучения, языков программирования и даже повторения движений опытного оператора.

Полезность промышленного робота может быть расширена с применением датчиков, передающих информацию в его систему управления. Наиболее практичными сейчас являются оптические и тактильные датчики, однако в будущем промышленные роботы смогут распознавать и голос.

При закупке робота инженер должен учитывать грузоподъемность, рабочий объем, повторяемость, точность и скорость. Роботы могут заменить человека на непристижной, утомительной или небезопасной работе, сохраняя в то же время другие рабочие места за счет повышения общей производительности. Проводя экономическое сравнение вариантов, анализирующий должен учитывать способность робота работать в три смены, если объем выпуска этого требует, не упуская из виду стоимость простоев и обслуживания. Экономическая целесообразность не является единственным фактором, необходимо учитывать также технические и эксплуатационные факторы.

У п р а ж н е н и я и в о п р о с ы

6.1. В чем состоит основное отличие промышленных роботов от их предшественников — механических манипуляторов?

6.2. Предположим, что ваша фирма приобрела многозвенную руку со следующими степенями подвижности: сгибание запястья; поворот запястья; вращение запястья; сжатие-разжимание захвата.

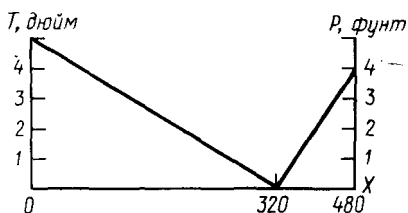
Объясните, каким оборудованием необходимо дополнить эту руку, чтобы превратить ее в робот с восемью степенями подвижности.

6.3. Если робот установить на обычный стол, имеющий возможность двигаться по осям X и Y , сколько дополнительных степеней подвижности он получает?

6.4. Что такое «режим слежения за конвейером»? Объясните его преимущества и недостатки.

Рис. 6.26

Графики зависимостей раскрытия захвата T (в дюймах) и усилия зажима P (фунт) от количества импульсов X , отработанных шаговым двигателем (1 дюйм = 25,4 мм, 1 фунт = 4,45 Н)



6.5. Назовите не менее двух языков программирования роботов и опишите некоторые их особенности.

6.6. Что такое «панель обучения»? Для какого режима обучения необходимо это устройство?

6.7. Что такое принцип сухожилия?

6.8. Сравните термины контроль присутствия с помощью оптических средств и зрение робота.

6.9. Какое отношение имеют колготки к осязанию роботов?

6.10. Какие факторы необходимо учитывать при грамотном сравнении роботов по грузоподъемности?

6.11. Вообще говоря, кто точнее: человек или робот? Кто обладает лучшей повторяемостью?

6.12. Какие три основных аспекта осуществимости робототехнического проекта нужно рассматривать при его обосновании? Назовите несколько факторов для каждого аспекта.

6.13. Сколько рабочих требуется, чтобы обслуживать одно рабочее место 24 ч в сутки 365 дней в год? Подтвердите ответ расчетами.

6.14. Робот с захватом, работающий по принципу сухожилия, обеспечивает зажим изделия в соответствии с графиком, приведенным на рис. 6.26. При условии, что захват начинает работать с полностью открытого положения, вычислите количество импульсов, которое необходимо подать на него системе управления, чтобы захватить заготовку толщиной 3 дюйма (76,2 мм) и зажать ее с усилием 24 унции (6,6 Н) перед тем, как поднять ее

6.15. На крупном заводе по производству авиационных конструкций имеются установки для обработки листового материала. Некоторые из них обслуживаются рабочими, а некоторые — роботами. На установках, обслуживаемых рабочими, брак составляет 10 %, в то время как при обслуживании роботами брак практически отсутствует. По умеренным оценкам, учитывая простои, роботизированная установка обрабатывает 50 изделий в 1 ч при себестоимости процесса 5 долл. за 1 шт. Какова годовая экономия за счет повышения качества, если робот работает одну стандартную 8-часовую смену в сутки? Какова экономия при круглосуточной работе? Укажите, какие допущения приняты вами при расчете.

6.16. Двухточечный внутренний захват (см. рис. 6.16) оснащен губками толщиной 0,5 дюйма (12,7 мм), что при полном закрытии руки дает суммарную толщину 1 дюйм (25,4 мм). Захват может использоваться и как наружный с расстоянием между полностью открытыми губками 3 дюйма (76,2 мм). Захват приводится в движение шаговым приводом по принципу сухожилия, причем подача одного импульса закрывает или открывает захват на $1/64$ дюйма (0,397 мм). Поддача дополнительных импульсов после соприкосновения с деталью увеличивает усилие зажима на 0,5 унции (0,14 Н) до максимального усилия 5 фунтов (2,2 Н). Начертите график зависимостей между раскрытием захвата, количеством шаговых импульсов и усилием зажима аналогично графику, приведенному на рис. 6.22.

6.17. Какими из цилиндрических деталей может манипулировать захват, описанный в упражнении 6.16 (D — наружный диаметр цилиндра, d — диаметр отверстия): а) $D = 3,5$ дюйма (88,9 мм), $d = 0,75$ дюйма (19,05 мм); б) $D = 2$ дюйма (50,8 мм), $d = 1,5$ дюйма (38,1 мм); в) $D = 6$ дюймов (152,4 мм), $d = 3,5$ дюйма (88,9 мм); г) $D = 1,5$ дюйма (38,1 мм), $d = 0,5$ дюйма (12,7 мм).

6.18. Сколько импульсов необходимо подать на привод полностью закрытого захвата, описанного в примере 6.17, чтобы зажать деталь, имеющую отверстие диаметром 2 дюйма (50,8 мм) с усилием 1 фунт (4,5 Н).

7. ОБУЧЕНИЕ РОБОТА ВЫПОЛНЕНИЮ РАБОТ

В гл. 6 показано, что промышленный робот является значительно более универсальным устройством, чем заменяемый им манипулятор. Однако эта универсальность оборачивается для пользователя обязанностью обучать робот выполнению данной операции или группы операций. Каждому уровню универсальности робота соответствует определенный уровень сложности его обучения, вернее его обучаемости. Однако технология, которая обеспечила универсальность робота, способна сделать обучение легким, продуктивным и удобным для потребителя. Наиболее процветающими среди производителей роботов оказываются те, которые концентрируют свое внимание на этой стороне взаимоотношений между роботом и человеком.

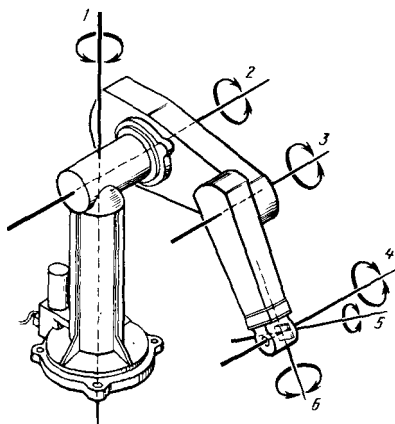
7.1. МИР РОБОТОВ И МИР РЕАЛЬНОСТИ

Чтобы осознать всю сложность проблемы обучения робота, проведем следующий эксперимент. Представьте себя стоящим между двумя урнами для голосования. Большим и указательным пальцами левой руки сделайте движение, как будто берете прямой стержень, вставленный в урну, поставленную на 1 фут (30,5 см) правее правого плеча и немного впереди него. Можете повернуть голову, чтобы видеть, как левая рука справляется с заданием, но корпус при этом держите как можно прямее. Теперь перемещайте воображаемый стержень по прямой линии влево, пока можно дотянуться, и имитируйте установку стержня во вторую урну.

Если перемещение стержня происходило действительно по прямой, рука по дороге проходила очень близко от лица и шеи. Повторите движение, обращая пристальное внимание на положение и действия локтевого и плечевого суставов и запястья. Движения весьма сложны, суставы принимают своеобразные положения, а гибкость, с которой они меняют положение, ускоряя и замедляя движение, даже когда скорость воображаемого стержня поддерживается постоянной, кажется сверхъестественной. Такова сложность обычного движения руки, такова наша способность принимать как должное тот метод, которым еще в детском саду учили рисовать прямые линии.

Рис. 7.1

Диапазоны перемещений в «координатах суставов» для робота «Пума-560»: 1 — корпус (320°); 2 — плечевой сустав (250°); 3 — локтевой сустав (270°); 4 — сгибание запястья (200°); 5 — поворот фланца (532°); 6 — поворот запястья (300°)



В гл. 5 при изучении программирования станков с ЧПУ сложные движения суставов не рассматривались. Большинство станков спроектированы для работы в реальном трехмерном пространстве, изображаемом осями X , Y , Z . О многозвенных роботах или им подобных, работающих в полярных или цилиндрических координатах, этого сказать нельзя. Имеющие дело с роботами говорят, что движения роботов описываются «координатами суставов» (рис. 7.1), в то время как реальные движения описываются декартовыми координатами X , Y , Z , которые иногда называют «мировыми координатами» (рис. 7.2). Таким образом, любая точка в декартовом пространстве рабочей зоны работы может быть описана набором положений по осям движения робота, например угловыми положениями в случае многозвенового робота. Существует, однако, еще одна область — область инструмента робота. Поскольку запястье робота может иметь различные исполнения, захват инструмента или другой рабочий орган может быть наклонен

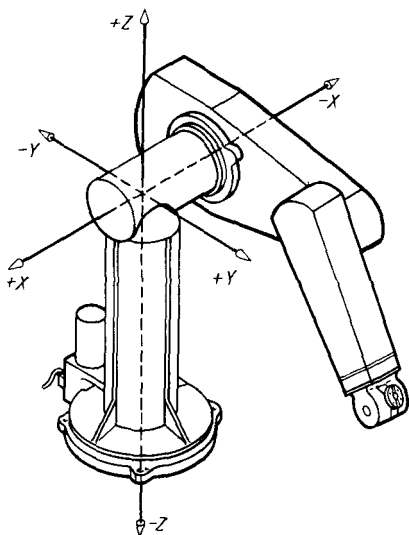


Рис. 7.2

Система «мировых координат» для робота «Пума-560»

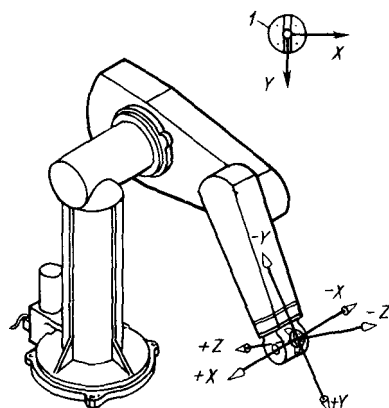


Рис. 7.3

Система координат инструмента для робота «Пума-560»;
1 — монтажный фланец (вид с торца)

по разным направлениям, даже когда центральная точка инструмента остается в одном и том же положении. Понятно, что одно положение в реальном мире (в мировых координатах) может быть представлено целой группой положений в мире робота (в координатах звеньев). Иногда может потребоваться запрограммировать движение робота, принимая за базу инструмент, а не робот или мировые координаты. При извлечении пневматического винтовёрта из глубокой цековки резьбового отверстия важно не положение звеньев робота или ориентация окружающих предметов (обстановки) в реальном мире, а положение инструмента. «Обратное» движение винтовёрта должно быть параллельно его оси центров. Обычно это интерпретируется как движение в отрицательном направлении по оси системы координат, именуемой «система координат инструмента» (рис. 7.3). Другие движения инструмента могут потребовать перемещения в плоскости XZ системы вместо перемещения только по оси Z .

Возможно рассчитать движение каждого звена робота и обеспечить любое перемещение в окружающем пространстве или в пространстве координат инструмента, задавая перемещения только в координатах звеньев робота. Теоретически это возможно, а на практике — нет. На рис. 6.10 было показано, какие сложные тригонометрические преобразования необходимо выполнить, чтобы перевести угловые положения звеньев с учетом их длин в координаты по одной только оси X декартовых координат. Подобные сложные вычисления требуются и для перехода к координатам по осям Y и Z , не говоря уже о комбинированных преобразованиях, необходимых для связи системы координат инструмента с реальным миром. Сложность проблемы возрастает, если робот способен осуществлять непрерывное движение по заданной траектории, что ставит задачу рассчитать не только переход от точки к точке, но и сопутствующее изменение скорости движения.

Подводя итоги, можно сказать, что первой отличительной чертой современного ориентированного на заказчика робота является приданная ему возможность перехода из одной системы координат в другую, осуществляемая ЭВМ, предпочтительно в режиме реального времени. Если быстроедействие ЭВМ недостаточно для выполнения расчета в режиме реального времени, потребуется провести расчет по двум программам: сначала установить положения рабочего органа в декартовых координатах, а затем перевести эти движения в координаты звеньев робота. Способность работать в режиме реального времени особенно важна для сложных контурных роботов. Удобным с точки зрения программирования является наличие на панелях управления этих роботов (или на панели обучения, клавиатуре и т. п.) переходителя, позволяющего программисту без затруднения переходить от координат звеньев к мировым координатам или к координатам рабочего органа.

7.2. МЕТОДЫ ПРОГРАММИРОВАНИЯ

Рассматривая основные методы программирования, следует отметить, что большинство распространенных роботов частично программируется механическим способом — при помощи гаечного ключа. Кроме программирования установкой упоров, многие роботы для облегчения процесса программирования обучаются с помощью имеющихся панелей обучения. Некоторые роботы с контурным программным управлением, особенно модели, предназначенные для окраски, и некоторые разновидности роботов для сварки, обучаются физическим перемещением рабочего органа или его макета по требуемому пути. Существует программирование с помощью клавиатуры системы управления на языке программирования для роботов. Последний способ является одновременно и самым утомительным, и самым производительным из всех описанных.

В этой книге не имеет смысла говорить об установке конечных упоров или об обучении роботов ручной проводкой их рабочих органов по требуемой траектории. Поэтому двум этим методам здесь уделяется мало внимания. В то же время стоит ознакомиться с использованием панели обучения, а тем более с подготовкой управляющих программ на языках программирования.

Программирование с помощью панели обучения. Программирование даже простых роботов с концевыми упорами может быть облегчено с помощью панели обучения. Для таких роботов панель обучения используется двояко: во-первых, чтобы обеспечить последовательность действий в цикле, во-вторых, чтобы задать время выполнения операций. Панель робота с концевыми упорами показана на рис. 7.4.

Пожалуй, самым примечательным в этой панели является наличие двух режимов обучения: *А* и *В*. В режиме обучения *А* задаются сразу последовательность действий и их продолжительность. Если оператор способен думать достаточно быстро, чтобы запрограммировать в режиме реального времени, т. е. может предвосхитить каждое движение робота и отреагировать на него в необходимом темпе, для программирования нужен только этот режим. Однако большинству операторов требуется время, чтобы обдумать каждый шаг программы, поэтому нужен дополнительный режим обучения для задания времени установленной последовательности действий — режим *В*. Необходимость режима *В* становится очевидной любому начинающему программисту, как только у него появляется возможность наблюдать за роботом, отрабатывающим его первую программу и неизбежно останавливающимся в разных местах, чего оператор не мог предусмотреть. Задав последовательность в режиме *А*, оператор может затем сосредоточиться на задании времени в режиме *В*, используя клавишу КАДР, чтобы отработать каждый запрограммированный кадр в реальном времени. Кроме обычных режимов обучения,

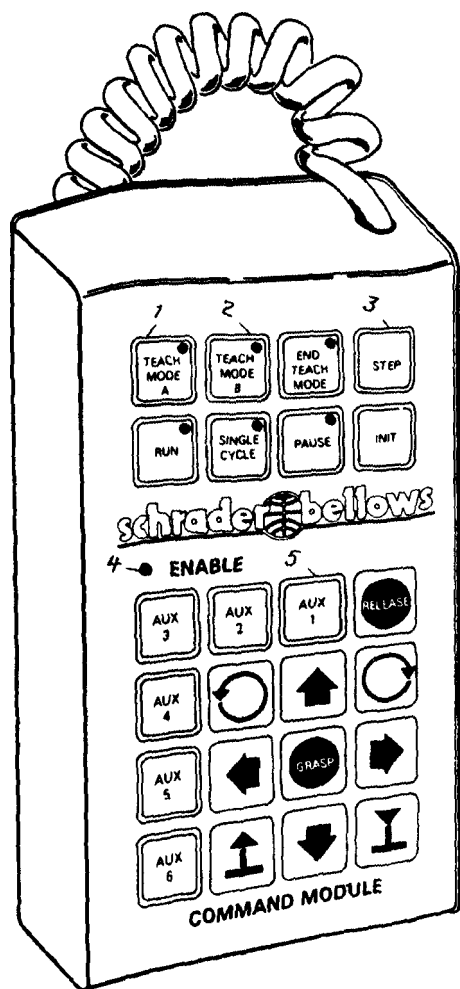


Рис. 7.4
Панель обучения робота с концевыми упорами:

1 — «Режим обучения А»; 2 — «Режим обучения В»; 3 — «Кадр»; 4 — «Готов»

существует тренировочный или экспериментальный режим, в котором оператор может провести робот через различные последовательности действий, чтобы рассмотреть их протекание без запоминания этой программы. В этом режиме можно узнать режим ручного управления, описанный в гл. 5, где рассмотрены станки с ЧПУ. В устройстве, изображенном на рис. 7.4, о работе в этом режиме свидетельствует горящая лампочка индикатора ГОТОВ.

Само программирование осуществляется с помощью клавиш, расположенных в правой нижней части клавиатуры панели обучения, изображенной на рис. 7.4. Эти клавиши обычно вместо слова обозначаются символами, что облегчает их использование. Несмотря на все внимание к «человеческому фактору», уделяемое

при конструировании клавиатуры, когда оператор только начинает программировать с помощью панели обучения, существует вероятность возникновения ошибок. Одной из проблем является определение правого и левого направлений, вне зависимости от того, обозначены ли они на клавиатуре словами налево или направо, или представлены стрелками. Требуется знать, для кого определяются эти направления: для робота или для оператора. В панели, изображенной на рис. 7.4, за основу принят робот, а для оператора все окажется наоборот, поскольку при программировании он обычно стоит к роботу лицом. Задача несколько упростится, если оператор при программировании встанет несколько сзади робота и будет «заглядывать ему через плечо». Клавиши вспомогательных функций AUX на панели предназначены для задания очередности внешних функций, реализу-

емых через систему управления роботом. Для введения в цикл необходимых внешних функций должно быть обеспечено подключение соответствующего оборудования к выходным зажимам системы управления роботом, после чего оператор может задать последовательность выполнения этих функций в нужных местах цикла, нажимая на клавиши AUX при программировании в режиме обучения А. Любая из дополнительных функций может быть введена в программу сколько угодно раз, и система управления роботом запомнит последовательность и время выполнения этих функций так же, как она запомнит движения робота. Таким образом, система управления роботом может использоваться в качестве управляющей ЭВМ или программируемого контроллера для всего комплекса, а не только для робота.

Подобно вспомогательным функциям к системе управления роботом могут быть подведены входные сигналы, позволяющие роботу распознавать и реагировать на изменение внешних условий, оказывающих влияние на его работу. На панели обучения, изображенной на рис. 7.4, клавиш для программирования ответных реакций нет, поскольку реакция однозначна: любой сигнал на входе понимается как универсальная команда на остановку робота. Такой подход полезен, когда сигнал на вход подается от следующих датчиков:

1. От концевого выключателя, установленного на питателе, подающем материалы, для сигнализации о том, что питатель пуст.

2. От фотозлектрического датчика, определяющего несоосность как неправильную ориентацию детали.

3. От концевого выключателя или датчика световой завесы, сигнализирующего о наличии оператора или постороннего предмета в рабочей зоне робота, что может вызвать аварию.

4. От концевого выключателя, сигнализирующего о заедании материала в подающем устройстве или на выходной дорожке питателя.

5. От электрического датчика, контролирующего физическое состояние руки робота или привязку к осям координат.

Этот список, как должно быть понятно читателю, можно продолжить.

Возможность останавливать робот по команде извне очень важна, но не менее полезна и возможность подавать команды извне, обеспечивая очередность действий робота. Классическим примером являются загрузка и разгрузка станков. Робот должен ожидать, пока дыропробивной пресс, машина для литья под давлением или другое технологическое оборудование выполнит свой цикл действий. Начало этого цикла может быть задано роботом (через вспомогательную функцию AUX) или датчиком через соответствующий вход в управляющее устройство станка, если таковое имеется. Однако после окончания цикла станка робот должен определить, что пора входить в рабочую зону станка

и извлекать обработанную деталь. Одним из возможных решений этой проблемы является изменение времени работы станка при наладке и программирование паузы соответствующей продолжительности в цикле робота. Этот подход вполне работоспособен, но не лишен некоторых недостатков. Время процесса обработки не может быть всегда совершенно одинаковым. Следовательно, необходимо, чтобы пауза в цикле робота была достаточно длинной по сравнению с самой большой продолжительностью цикла станка. В противном случае между роботом и станком, который он обслуживает, может произойти столкновение. Однако такая длинная пауза неизбежно приведет к бесполезному расходованию ценного общециклового времени, определяемого в этом случае с учетом самого длинного цикла обработки. Кроме того, наличие датчика, который сигнализирует об успешном окончании цикла обработки, вселяет некоторую уверенность в том, что не просто прошло достаточно времени для выполнения этого цикла, но что цикл закончен успешно и рабочая зона свободна для выгрузки детали. Вспомните закон Мэрфи и представьте себе, как пуансон прессы до упора входит в матрицу. В эту картину не вписывается безмозглый робот, слепо пытающийся ввести захват в рабочую зону.

Если мы обратимся к более совершенным роботам, легко представить себе другие внешние события, которые могли бы быть трансформированы во входные сигналы для систем управления. Даже некоторые «циклы остановки» робота, рассмотренные выше, у более совершенных роботов могут быть заменены запрограммированной последовательностью действий, направленной на устранение причин остановки. Если питающий бункер опустел, робот может быть заранее запрограммирован на его загрузку из ближайшей палеты. Несоосность детали может побудить робота перезахватить деталь и снова поместить ее под штамп или в другую точку рабочей зоны. Если оператор вторгается в рабочую зону, робот может быть запрограммирован выждать некоторое время, выдать предупреждающий сигнал и, после того как оператор выйдет из зоны, продолжить работу. Эти усовершенствования наиболее трудно ввести, используя только программирование с пульта обучения.

В целом метод программирования с использованием обучения является мощным, удобным способом обучения робота выполнению простых заданий. Особенно эффективен он при программировании операций типа «взять — установить», когда положения точек, откуда необходимо взять деталь и куда ее установить, не изменяются.

Вместе с тем, в цикл работы робота могут быть внесены дополнительные функции, причем запрограммированная реакция на сигналы извне становится с виду странной, потому что каждый из входных сигналов логически вызывает свою собственную последовательность действий робота. Кроме того, используя только панель обучения, трудно запрограммировать такие траек-

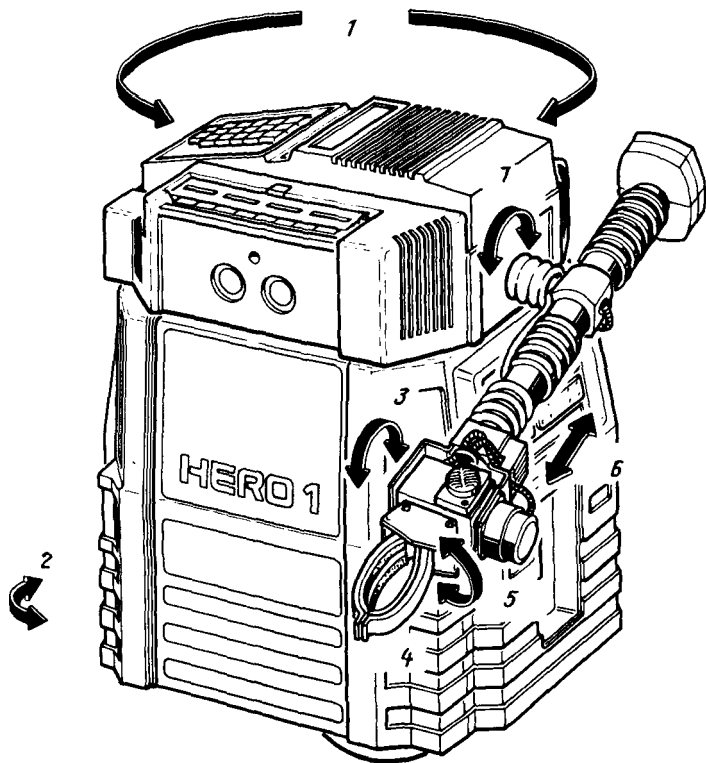


Рис. 7.5

Компактный самоходный любительский робот с вмонтированной в него клавиатурой управления:

1 — поворот башни; 2 — поворот переднего колеса; 3 — сгиб запястья; 4 — захват; 5 — поворот запястья; 6 — смещение руки; 7 — поворот руки

тории, как окружности или даже прямые линии. Наконец, этот вид программирования не позволяет использовать мощные методы итерационного программирования и применять подпрограммы.

Программирование посредством клавиатуры. Выше показаны преимущества программирования с помощью панели обучения. В то же время указаны и некоторые недостатки этого метода. Как читатель, наверное, уже догадался, программирование посредством клавиатуры свободно от этих недостатков. Однако следует усвоить, что управление с панели обучения или с клавиатуры аналогичного типа, позволяющее непосредственно приводить робот в движение, является основой любой практичной, эффективной системы программирования посредством клавиатуры.

Нельзя утверждать, что совсем нет роботов, управляемых исключительно по программе, подготовленной аналитически с клавиатуры. Имеется несколько интересных моделей, предназначенных для любителей робототехники и программирования.

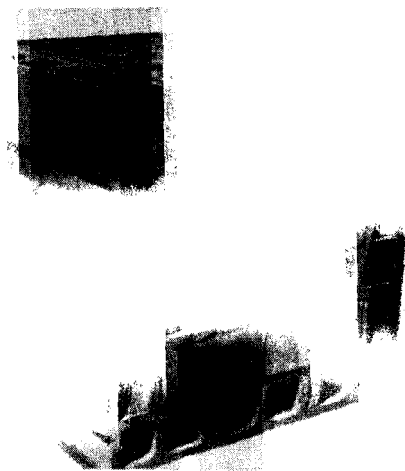


Рис. 7.6

Штабель строительных блоков, уложенный промышленным роботом

На рис. 7.5 представлен популярный робот для любителей, оснащенный клавиатурой для программирования, укрепленной с правой стороны, что объединяет робот, управляющую ЭВМ и средства программирования в единую компактную мобильную систему. Вообще мобильный робот нецелесообразно оснащать отдельной управляющей ЭВМ, связанной с роботом при помощи кабеля или бесконтактным способом.

Язык программирования робота, изображенного на рис. 7.5, представляет собой микропроцессорный шестнадцатеричный код, о котором рассказано в гл. 14. С точки зрения требований к памяти, техническим средствам и программному обеспечению ЭВМ, предъявляемых шестнадцатеричным языком ассемблера, этот язык является в высшей степени эффективным. В то же время он несколько утомителен своей детальностью. Любителю же, желающему посвятить свое время и усилия разработке сложных программ, упомянутый язык позволяет это сделать с наименьшими затратами и при максимальной компактности систем.

Чтобы обеспечить быстрое и легкое программирование роботов для выполнения различных задач, необходим легко усваиваемый мощный язык программирования, применяемый вместе с панелью обучения для управления роботом во время программирования и обучения. Чтобы проиллюстрировать необходимость иметь мощный язык программирования в дополнение к панели обучения, рассмотрим задание роботу сложить строительные блоки в штабель, как показано на рис. 7.6. В принципе возможно, пользуясь панелью обучения, научить робот брать по одному блоку и точно устанавливать его в требуемое место. Реализация этого метода будет зависеть от объема памяти системы управления роботом,

потому что ей придется запомнить каждое движение каждого блока. Для всего штабеля программа может состоять из нескольких тысяч таких движений робота. Представьте себе ужас программиста, если смена рабочего места потребует, чтобы штабель был немного передвинут!

При наличии панели обучения и языка для программирования с клавиатуры робот можно обучить нескольким основным положениям в удобной системе координат, например в мировой системе, манипулируя его рукой, управляемой с панели обучения. Затем, используя клавиатуру системы управления, можно подготовить такую программу, которая будет повторяться, базируясь на основных положениях, и осуществлять процесс итерации, укладывая каждый слой в установленном порядке. При этом учитываются размеры укладываемых блоков. Можно составить программу таким образом, чтобы она позволяла производить укладку блоков различного размера, изменяя стратегию укладки с целью оптимизации процесса для каждого размера. Наконец, могут быть разработаны многочисленные подпрограммы, предписывающие действия робота в случае возникновения тех или иных внешних обстоятельств.

Языки программирования, разработанные для выполнения таких задач, можно легко изучить за один-два дня. Имеющие опыт в программировании обычно осваивают языки программирования для роботов за считанные часы. Некоторые из этих языков даже написаны как диалекты некоторых существующих популярных универсальных языков программирования, таких, как Бейсик. Два широко используемых языка, таких, как ВАЛ фирмы «Юнимейшн» [73] и АРМ Бейсик фирмы «Микробот» [42], выбраны для того, чтобы проиллюстрировать формат и применение языков программирования для роботов вообще.

7.3. ЯЗЫК ВАЛ

Язык программирования роботов ВАЛ разработан на фирме «Юнимейшн», которая владеет авторскими правами на управляющую программу языка, программу редактирования и документацию [73]. Ниже приведено описание формата и особенностей языка ВАЛ. Пользователи промышленного робота ЮНИМЕЙТ обеспечиваются документацией в полном объеме и могут пройти курс обучения этому языку на заводе-изготовителе или у себя *.

Многие характеристики языка ВАЛ окажутся знакомы читателю, владеющему языком Бейсик. Программирование на языке ВАЛ, так же как и на языке Бейсик, производится с помощью

* Фирма «Юнимейшн—Вестингауз» является разработчиком и владельцем авторских прав на программное обеспечение языка ВАЛ, кратко описанного здесь с разрешения фирмы. За полным комплектом документации на язык следует обращаться по адресу: Unimation—Westinghouse, Inc. Shelter Rock Lane, Danbury, Conn. 06810.

клавиатуры терминала и поэтому требует наличия некоторых навыков обращения с печатными знаками, добавления и стирания строк, внесения изменений в программу, листинга операторов программы и на экране дисплея и т. д. Система этих средств подготовки и эксплуатации программ называется РЕДАКТОР и в языке ВАЛ, и в большинстве языков программирования, основывающихся на языке Бейсик. Во время редактирования система РЕДАКТОР создает дополнительные удобства для программиста, указывая ему, когда он совершает лингвистические (синтаксические) ошибки при составлении оператора программы. Конечно, в программе можно сделать ошибки, которые РЕДАКТОР не заметит: такие программы соответствуют всем правилам языка ВАЛ, но заставляют робот сделать что-нибудь нелогичное или по меньшей мере не совсем то, что хотел пользователь.

После того как редактирование закончено, программа может быть ИСПОЛНЕНА по команде с клавиатуры терминала, аналогично языку Бейсик. Однако с этого места характеристики языков ВАЛ и Бейсик начинают различаться. В языке Бейсик фазы редактирования и выполнения обычно выполняются раздельно. В языке ВАЛ абсолютно неограниченно можно видоизменять программу с помощью РЕДАКТОРА прямо во время ее выполнения. Как только программист завершает редактирование, осуществляемое после видоизменения существующей программы, которая исполняется в данное время, робот начинает действовать по измененной программе прямо посередине выполнения предыдущей версии программы. Как долго ни приходится заниматься старой программой, чтобы закончить один ее шаг, он начинает выполняться в тот же момент, как только с клавиатуры поступает команда о завершении редактирования.

Другим важным шагом при программировании на языке ВАЛ является физическое определение положений точек с помощью панели обучения. Этого, конечно, не делается в обычной программе, написанной на языке Бейсик. Фаза обучения с помощью панели обеспечивает связь между роботом, который должен работать в реальном мире, и программой на языке ВАЛ, написанной в таких символических терминах, как POINTA и POINTB. Панель обучения не используется для показа роботу всех точек, где он должен быть. Необходимо указать только несколько основных точек, таких, как расположение станка, конец конвейерной ленты или угол палеты. Тысячи других точек, в которых робот должен побывать, могут быть заданы от основных точек. В этом и проявляется сила языка ВАЛ.

Прежде чем перейти к программированию, изучим процесс определения местонахождения с помощью панели обучения. На рис. 7.7 представлена панель обучения робота «Пума» фирмы «Юнимейшн». Во втором ряду имеются клавиши «ИНСТРУМЕНТ, МИР, ЗВЕНО, (клавиши 9, 10 и 11). Эти три клавиши относятся к трем системам координат, объясненным в начале этой главы

Рис. 7.7

Панель обучения робота «Пума». Положения переключателя режимов работы:

А — замедленный; Б — нормальный; В — ускоренный

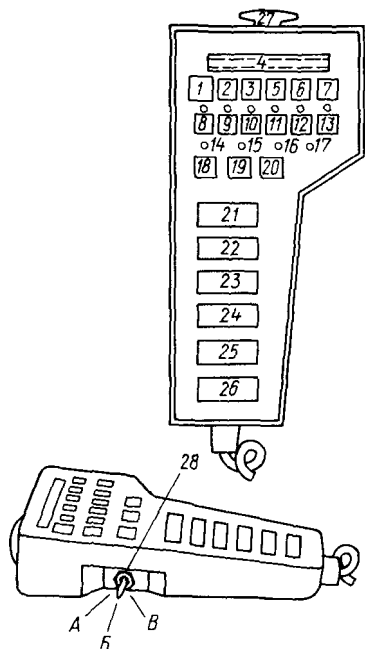
при сравнении систем координат робота и реального мира. Координатные системы являются взаимноисключающими, и светодиодные индикаторы, расположенные непосредственно над каждым из клавиш, указывают, в какой из состояний находится система. Шесть клавиш, расположенных одна над другой в нижней части панели (клавиши с 21 по 26), используются для физического перемещения органов робота в точки, которые необходимо запомнить. При этом программист может выбрать наиболее удобную координатную систему (обычно МИР или ИНСТРУМЕНТ). Робот, конечно, использует свою систему координат ЗВЕНО, но постоянно производит трансформацию одной системы в другую, что позволяет программисту использовать выбранный режим. Когда робот доходит до требуемой точки, оператор печатает с клавиатуры команду HERE (ЗДЕСЬ), а затем символическое наименование этой точки, которое затем будет использоваться в программе на языке ВАЛ. Например, запись HERE POINTA (здесь точка А) определяет текущее физическое положение робота, имеющее наименование POINTA.

Наименованием положения может быть любой непрерывный набор букв, цифр и других знаков, при условии, что первым знаком будет буква. Ниже приведены несколько употребительных наименований положений:

POINTA
POINT.A
A
MACHINE
PICK.POINT
PLACE.POINT
PALLET.CORNER
PROGRAM.1
PGM.2

Рассмотрим инструкции на языке ВАЛ, которые относятся к определению положений при обучении робота.

Каждая программа на языке ВАЛ должна иметь наименование, чтобы программу можно было вызвать для ИСПОЛНЕНИЯ,



занести в машинный файл или вызвать снова для анализа и внесения изменений в режиме редактирования (EDIT). Выбор наименования программы остается за пользователем, но, конечно, наименования не должны повторяться, чтобы не спутать программы, хранящиеся в памяти ЭВМ. Подготовка новой программы с прежним наименованием и внесение ее в память автоматически уничтожат старую версию программы. Правила составления наименований программ аналогичны правилам составления наименований положений, приведенным выше.

Инструкции на языке ВАЛ являются предписаниями роботу сделать что-нибудь и имеют знакомый формат, подобный формату языка Бейсик. На одной строке располагается одна инструкция, и основным полем является командное слово (или сокращение), отделенное от других командных слов пропуском. Переменные (параметры), используемые для того, чтобы специфицировать инструкции, отделяются запятыми. Если в дальнейшем потребуется перейти к данной инструкции из какого-либо места программы, этой инструкции присваивается произвольная метка. Следующий пример иллюстрирует форма команды на языке ВАЛ.

65 APPRO MACH, 50

65 — произвольная метка, указывается для того, чтобы позже программист вернул ЭВМ к этой инструкции;

APPRO — код инструкции, приказывающий роботу «приблизиться» к заданной точке;

MACH — наименование положения, показывающее роботу то положение, к которому он должен «приблизиться»;

50 — расстояние, мм, устанавливающее, как близко робот должен «приблизиться» к положению MACH.

Из приведенного примера видно, что инструкция длиной в одну строку является законченным предписанием роботу выполнить какое-либо действие. Следующая строка программы будет содержать другую инструкцию, и эти инструкции, выполняемые роботом, будут следовать строка за строкой, пока программа не скамандует вернуться или перейти к другой строке инструкций, имеющей метку, такую, как, например, число 65, произвольно присвоенное инструкции, приведенной выше.

Одним из самых эффективных способов изучения нового языка — любого языка — является попытка начать использовать его, не дожидаясь полного освоения. Мы используем здесь этот подход, написав программу на языке ВАЛ. В скобках справа от каждого шага программы помещено объяснение того, что инструкция предписывает делать роботу. Верхняя строка является наименованием программы, соответствующим ее назначению, заключающемуся в том, что робот просто должен забрать предметы из точки А и перенести их в точку В.

Программа A.T.O.B. (наименование программы)

OPEN (открыть захват робота)

MOVE A (переместить захват робота в определенную позицию А)

CLOSEI (закрыть захват робота немедленно)
MOVEB (переместить захват в определенную позицию B)

Это работоспособная программа на языке ВАЛ, хотя, как мы скоро обнаружим, и не очень хорошо составленная. Первым очевидным несоответствием кажется использование команды OPEN, чтобы открыть захват, и команды CLOSEI, чтобы закрыть его немедленно. Действительно, мы могли бы подать команду OPENI, чтобы немедленно открыть захват. Команда OPEN без добавленного I предписывает роботу начать открывать захват во время выполнения следующей команды. При условии, что до позиции A достаточно далеко, не будет большого вреда позволить роботу открыть захват во время перемещения к этой позиции. Важно, однако, закрыть захват на предмете до выполнения команды MOVEB, иначе он уйдет, не захватив предмет. Поэтому здесь и использована команда CLOSEI, предписывающая немедленно, т. е. до выполнения команды MOVEB, закрыть захват.

Однако, как уже говорилось, программа A.TO. B не из лучших. Получив команду MOVE, робот двигается непосредственно к пункту назначения, используя свои способности перемещения по осям координат. Прделанный путь не обязательно будет прямым, но он обязательно приведет непосредственно к захватываемому предмету. Представьте себе, как рука робота движется прямо к месту, где находится этот предмет. В подавляющем большинстве случаев рука робота соььет предмет вместо того, чтобы захватить его. Кроме того, во время перемещения руки к позиции B перемещение происходит прямолинейно, и, если позиции A и B находятся на поверхности стола, рука будет скрести по ней, вместо того, чтобы захватить что-нибудь! Наконец, программа заканчивается, когда зажатый в захват предмет находится над позицией B, куда робот должен был этот предмет поместить.

Усовершенствуем программу A.TO.B. Можно было изменить старую программу в режиме редактирования EDIT, сохранив наименование A. TO. B, однако лучше дать измененному варианту новое наименование, что облегчит распознавание вариантов.

BETTER.A.TO.B (наименование новой усовершенствованной программы)

OPEN

APPRO A, 25 (подойти к позиции A на расстояние 25 мм)

MOVE A

CLOSEI

DEPART 25 (отойти от A на 25 мм)

APPRO B, 25 (подойти к позиции B на расстояние 25 мм)

MOVE B

OPENI

DEPART 25 (отойти от B на 25 мм)

Эта программа решает проблемы, не решенные программой A.TO. B. Команда APPRO заставляет руку робота зависнуть непосредственно над заданной точкой, вместо того чтобы прямо прийти в нее. Выражение «зависнуть» является неточным, поскольку предмет может быть прикреплен к вертикальной стене

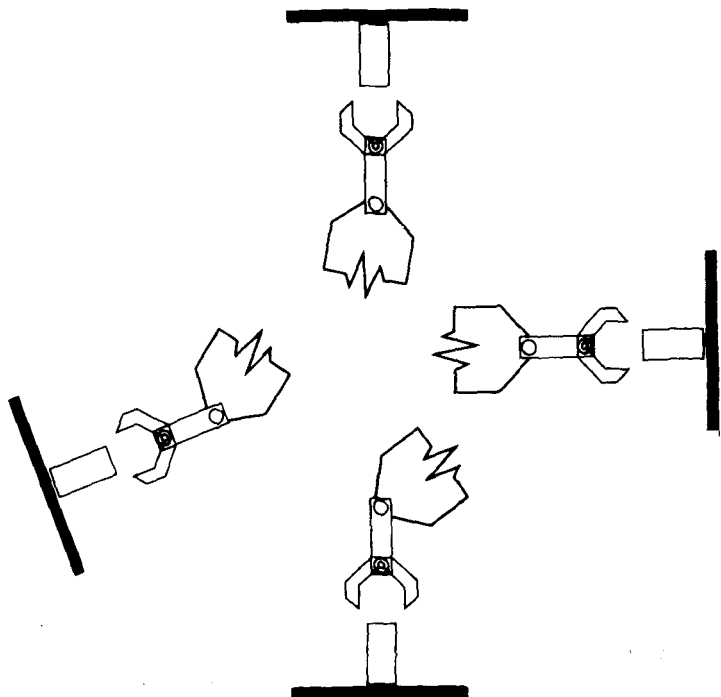


Рис. 7.8

Различные положения захвата в зависимости от описания позиции A

и даже к потолку. На рис. 7.8 показаны различные возможные положения, в которых робот может осуществить захват в зависимости от описания позиции A . Обратите внимание, что вне зависимости от расположения поверхности, с которой берется предмет, и угла, под которым движется рука робота, захват самостоятельно устанавливается так, чтобы подход к предмету осуществился правильно за счет непосредственного смещения по оси Z в системе координат инструмента. Читатель может снова обратиться к рис. 7.3, чтобы вспомнить, что такое ось Z в этой системе координат. Когда программист пользуется командой `APPRO`, программное обеспечение языка `ВАЛ` автоматически выполняет необходимый перерасчет осей, чтобы правильное направление подхода совпадало с осью Z захвата. Эта команда удобна не только для захвата предметов, которые необходимо перенести, но и для управления автоматическими винтовертами и другими инструментами, которые могут быть закреплены на руке робота.

Предположим, что требуется подходить к захватываемому предмету в направлении, отличном от перпендикулярного к поверхности стола, как это показано на примере на рис. 7.8. Эффективным направлением подхода для осуществления захвата некоторых предметов может быть направление сбоку, а не сверху.

По команде APPRO подход может быть осуществлен под любым требуемым углом. Методы языка ВАЛ для истолкования вопроса о желаемом угле подхода заключаются в точном согласовании ориентации инструмента, в котором он находился при определении положения на стадии обучения, с командой HERE, подаваемой с клавиатуры. Фактически угол подхода к позиции A можно изменить по сравнению с углом подхода к позиции B, просто изменив ориентацию инструмента во время позиционирования в режиме обучения по команде HERE B. После этого, сколько бы раз ни выполнялась программа BETTER.A.TO.B, требуемая ориентация захвата при подходе сохранится.

Программа BETTER.A.TO.B более совершенна, но ее использование в условиях реального производства не решает всех проблем. Успех применения программы зависит от выполнения довольно жестких требований, предъявляемых роботом к рабочему месту, которое он обслуживает. Предполагается, что каждый раз, когда робот возвращается в точку A, в точно установленном месте находится предмет, готовый к переноске. Кроме того, каждый раз, когда робот перемещается в позицию B, предполагается, что там имеется свободное место для помещения нового предмета. Таким образом, совместно с роботом, управляемым по программе BETTER.A.TO.B должен работать оператор или надежная автоматизированная система, в противном случае комплекс функционировать не будет.

Программы на языке ВАЛ будут выполняться столько раз, сколько указано в команде EXECUTE. Если таких указаний нет, программа будет выполнена один раз. На производстве чаще всего требуется, чтобы программа робота повторялась непрерывно до тех пор, пока ее не остановят командой ABORT. Чтобы обеспечить непрерывное выполнение программы, оператору следует всего лишь набрать вместо числа повторений любое отрицательное число. Поскольку отрицательное число повторений совершенно не имеет смысла, установлено, что оно воспринимается системой управления как команда на непрерывное повторение программы. После выполнения последней команды программное обеспечение просто возвращает систему управления к первой команде, не сбивая ритма.

В любом месте программы скорость ее отработки может быть изменена командой SPEED (сокращенно SP) с клавиатуры. Принято, что целое число, набранное после слова SPEED, представляет собой в процентах долю изменения стандартной скорости в диапазоне от 0 до 327 %. Это полезно в том случае, когда сначала надо снизить скорость для отладки программы, а затем, при исполнении ее, снова увеличить. Когда ABORT или просто A, набранная с клавиатуры терминала, прекращает выполнение программы (выполнение текущего шага), как только нажатие клавиши введет букву A. Кроме того, имеется кнопка экстренного отключения питания руки робота (EPO).

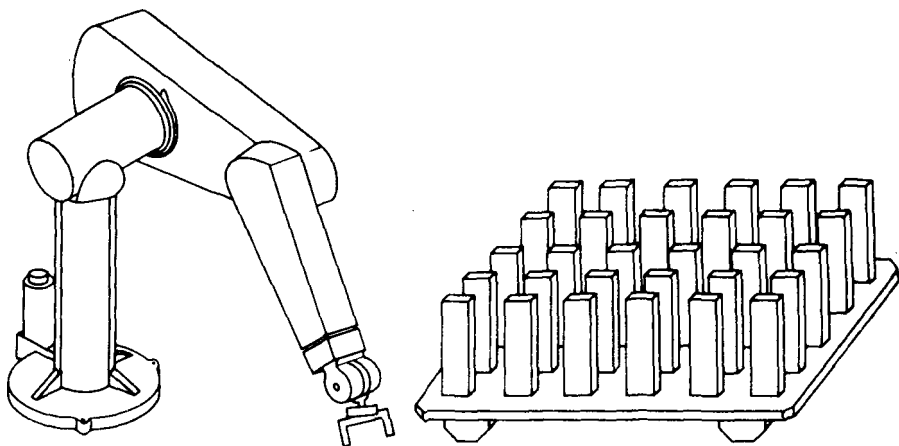


Рис. 7.9

Поддон может быть целиком загружен или разгружен роботом, использующим программные циклы

Используя возможности языка ВАЛ, можно применить программные циклы, заставляющие робот каждый раз слегка изменять положение позиции *A* с тем, чтобы постепенно разгрузить весь поддон изделий вместо того, чтобы каждый раз забирать изделие с одного и того же места. Предположим, что имеется поддон с 30 изделиями, установленными так, как показано на рис. 7.9. Обратите внимание, что изделия расставлены в пять рядов и шесть колонок. В зависимости от габаритов деталей и типа робота для манипулирования с ними вместо поддона может быть транспортная палета или лоток на конвейере.

Чтобы подготовить программу для робота на разгрузку поддона с 30 заготовками, необходимо знать расстояние между рядами и между колонками. Если эти расстояния одинаковы, программа будет значительно эффективнее. В этом примере ярко проявляется важность порядка в расстановке продукции и ее одинаковости. В планируемой упорядоченной автоматизированной системе действительные расстояния должны быть известны заранее или их можно измерить с помощью панели обучения. Примем расстояние между рядами 30 мм, а между колонками 40 мм, если измерять по серединам деталей. Тогда, если робот начинает с переднего левого угла и разгружает ряд за рядом, ему для захвата следующей детали необходимо каждый раз сдвигаться вправо на 40 мм. После окончания разгрузки каждого ряда робот должен снова вернуться налево, сдвинуться назад и начать новый ряд. Процесс продолжается до тех пор, пока поддон не опустеет. После этого робот может напечатать на экране дисплея или с помощью пишу-

щей машинки КОНЕЦ или НОВЫЙ ПОДДОН, ПОЖАЛУЙСТА и обождать немного, позволяя оператору или автоматизированной транспортной системе заменить пустой поддон на полный. Ниже приведена соответствующая программа.

TRAY.TO.B	(начать с угла поддона)
95 SETA-AA	(AA)
SETI NBR=0	(ввести целочисленную переменную NBR=0. Эта переменная относится к подсчету деталей до 30 во время установки их в позицию B)
85 GOSUB TRAY	(обращение к подпрограмме с наименованием TRAY, которая определяет сдвиг от ряда к ряду и от колонки к колонке)
OPEN	
APPRO A, 25	
MOVE A	
CLOSEI	
DEPART 25	
APPRO B, 25	(аналогично программе BETTER.A.TO.B.)
MOVE B	
OPENI	
DEPART 25	
SETI NBR=NBR+1	(прибавить единицу к сумме каждый раз, когда очередная деталь перенесена в позицию B)
IF NBR LT 30 THE 30	(если сумма NBR еще не равна 30, перейти к строке команды, обозначенной 85, и перенести следующую деталь)
THEN 85	
TYPE NEW TRAY, PLEASE	(если сумма NBR равна 30, разгрузка поддона закончена, напечатать об этом соответствующее сообщение)
DELAY 10	(ожидать нового поддона в течение 10 с)
GOTO 95	(перейти к строке команды с обозначением 95 и начать все сначала с новым поддоном)

Поскольку программа TRAY.TO.B содержит инструкцию GOSUB TRAY, нужно написать другую программу (точнее, подпрограмму), к которой затем следует переходить. Целью подпрограммы является сведение в единый блок последовательности команд, необходимых для того, чтобы при описанной выше расстановке деталей сдвигать руку робота на 40 мм по мере того, как она переходит от колонки к колонке в каждом ряду. Затем, закончив ряд, рука робота перемещается назад в крайнее левое положение, чтобы начать новый ряд. Направление этого перемещения вдоль оси X отрицательно, если рассматривать его в мировых координатах:

Перемещение = $-40 \text{ мм} \times 5 \text{ колонок} = -200 \text{ мм}$.

Множитель 5, а не 6 используется, поскольку руке робота не надо сдвигаться, чтобы забрать первую деталь в углу поддона. Кроме того, что рука робота возвращается вдоль ряда, для перехода в новый ряд ее надо сдвинуть на 30 мм, на этот раз вдоль оси Y.

TRAY	(наименование подпрограммы)
IF NBR EQ 0 THEN 60	(если это первая деталь в углу поддона, не сдвигаясь, перейти к команде с обозначением 60)

IF COL EQ 6 THEN 50	(если номер колонки COL уже равен 6, ряд закончен, и можно перейти к первой колонке слева и начать новый ряд. Для этого перейти к команде с обозначением 50)
SETI COL=COL+1	(переместиться на 40 мм по оси X. Перемещение происходит в одном ряду)
RETURN	(при подсчете количества пройденных колонок добавить единицу)
50 SHIFT A BY—200, 30	(вернуться в основной программе к команде, идущей после команды GOSUB, и взять деталь из новой смещенной позиции)
60 SETI COL=1	(переместиться на расстояние 5·40 мм назад по оси X и сдвинуться на 30 мм к новому ряду)
RETURN	(приравнять целочисленную переменную COL единице, чтобы показать, что начинается новый ряд с крайней левой колонки COL 1)
	(вернуться в основной программе к команде, идущей после команды COSUB, и забрать деталь)

Новая программа TRAY. TO. B, снабженная подпрограммой TRAY, обеспечивающей работу с поддонами, значительно более совершенна, чем программа A. TO. B. Рассмотрим еще раз некоторые особенности новых команд на языке ВАЛ, использованных в последней версии программы. Инструкция SHIFT вызывает обычный сдвиг из известного положения по осям X, Y и Z в мировых координатах. Поддон был точно сориентирован по оси X вдоль рядов и по оси Y вдоль колонок. Если их надо поменять местами, две команды на осуществление сдвига следует записать следующим образом:

```
SHIFT A BY 0, 40
SHIFT A BY 30, —200
```

Поскольку сдвиг по оси X в первом из двух предписаний равен 0, его можно опустить, написав:

```
SHIFT A BY, 40
```

Обратите внимание, что запятую перед числом 40 необходимо сохранить с тем, чтобы сдвиг на 40 мм был правильно понят: как сдвиг по оси Y, а не по оси X. Допускается указывать сдвиг сразу по трем осям:

```
SHIFT PLACE BO 25, —8, 16
```

что означает по оси X сдвиг + 25 мм; по оси Y сдвиг — 8 мм; по оси Z сдвиг + 16 мм.

Поддон тем или иным образом был сориентирован по осям координат, при установке под некоторым углом работу нельзя было бы осуществлять по обычной команде SHIFT. Для работы в этих условиях в языке ВАЛ имеется инструкция FRAME, однако рассмотрение перемещений такого типа выходит за рамки этой книги.

Другой новой инструкцией, использованной в программе TRAY. TO. B, была команда SETI. Эта инструкция позволяет

программисту легко устанавливать значения целочисленной переменной (I обозначает целое число), что удобно при разработке логики программ. Наименование переменной произвольно присваивается программистом, однако при условии, что он будет следовать тем же правилам, что и при наименовании программ и позиций. Как было показано в программе TRAY.TO.B, инструкция SETI позволяет прибавлять целочисленное постоянное (или переменное) значение к целочисленной переменной. Кроме того, можно вычитать (—), умножать (×), делить на целое число (/) и вычислять целочисленный остаток (%). Инструкцию SETI не следует путать с командой SET, которая используется, чтобы в мировой системе координат и системе координат инструмента установить координаты позиции с данным наименованием, если они совпадают с координатами другой позиции, уже хранящимися в памяти.

Еще одной инструкцией, использованной в программе TRAY.TO. B, является инструкция IF (если). Эта инструкция позволяет проверить значение ранее установленной целочисленной переменной и принять соответствующие меры. В программе TRAY. TO. B инструкция IF сочетается с сокращением LT, обозначающим «меньше чем». В подпрограмме TRAY две инструкции IF сочетаются с сокращением EQ, обозначающим «равен». Допустимые сокращения, сочетающиеся с инструкцией IF, приведены ниже:

EQ — равен; NE — не равен; LT — меньше чем; GT — больше чем; LE — меньше или равен; GE — больше или равен.

Во всех высказываниях IF, если условное предложение истинно, программа переходит к тому высказыванию, у которого обозначение совпадает с обозначением, следующим за словом THEN (тогда). Если надо всегда переходить к другой инструкции с обозначением, инструкция IF не используется. Вместо нее следует применять инструкцию «GOTO», как было показано в программе TRAY. TO. B.

В программе TRAY. TO. B инструкция DELAY (выдержка времени) предписывает роботу выждать 10 с, пока оператор или автоматизированная система не подадут новый поддон с деталями. Инструкция DELAY отличается от инструкции PAUSE, которая предписывает ожидание роботу до тех пор, пока оператор с клавиатуры не напечатает PROCEED (продолжать). Ввод с клавиатуры требует много времени, и его нецелесообразно применять каждый раз, когда требуется новый поддон. При помощи инструкции DELAY период 10 с может быть достаточно увеличен, чтобы установить поддон, однако, если время ожидания слишком велико, время расходуется непроизводительно и при использовании этой функции. Поскольку время смены поддона может несколько меняться, время ожидания лучше устанавливать с запасом. Однако это не самое лучшее решение, потому что время, напрасно потраченное во время простоя робота, когда поддон уже заменен, может решительным образом повлиять на оценку экономической

целесообразности внедрения робота. Целесообразно использовать датчик, который автоматически информирует робот об установке поддона. Кроме того, робот может выдавать сигнал о том, что он закончил разгрузку поддона и готов разгружать следующий, что заставит немедленно сработать систему, подающую ему детали. В этом случае можно не только обойтись без произвольной паузы 10 с, но и отказаться от инструкции TYPE (печать). Роботу, может быть, придется ожидать нового поддона, но он сможет начать работать, как только поддон будет установлен. Напишем новую программу на языке ВАЛ под названием BEST (самая лучшая), поскольку на этом изучение языка ВАЛ заканчивается.

Программа BEST

```

.
.
.
(Первые 12 команд идентичны первым 12 командам программы TRAY. TO. B.)
.
.
IF NBR LT 30 THEN 85 (то же, что и ранее)
SIGNAL 1              (подать сигнал на выходные зажимы № 1, чтобы
                      информировать систему подачи поддонов о готовности
                      робота разгружать новый поддон)
WAIT 1                (ожидать появление входного сигнала на входных
                      зажимах № 1, что должно свидетельствовать об уста-
                      новке нового поддона и его готовности к разгрузке)
SIGNAL —1            (снять сигнал готовности робота, как только робот
                      начинает разгружать новый поддон)
GO TO 95              (аналогично предыдущей программе TRAY. TO. B)

```

Входной сигнал от робота или входной сигнал, подаваемый на вход робота, может быть двух уровней: высокого или низкого. Это говорит о том, что сигналы являются переменными логических схем на постоянном токе, что будет рассмотрено в гл. 10—12. Инструкция SIGNAL на языке ВАЛ дает возможность подать сигналы высокого или низкого уровня на столько входных и выходных каналов, сколько требуется. Знак при номере входного или выходного канала означает уровень подаваемого сигнала. Например:

SIGNAL 6, —3, 5

предписывает роботу одновременно включить выходной канал 6 (высокий уровень сигнала), выключить выходной канал 3 (низкий уровень) и включить выходной канал 5 (высокий уровень).

Инструкция WAIT может действовать только один входной канал и заставляет робот приостановить дальнейшее выполнение программы до тех пор, пока на указанном входе появится сигнал требуемого уровня. Например, инструкция WAIT-3 предписывает роботу ожидать, пока уровень сигнала на канале 3 станет низким, и затем перейти к следующей инструкции. Инструкция WAIT-2 предписывает роботу ожидать, пока уровень сигнала на канале 2 станет высоким, и затем перейти к следующей инструкции.

В приложении А приведен пример программы на языке ВАЛ, разработанной студентами Центра робототехники и автоматизации Арканзасского университета. Программа показывает, каким образом робот ПУМА может быть обучен синхронизации движения с движением дополнительного оборудования даже при сборке «на лету» на движущемся конвейере.

Другая программа на языке ВАЛ (приложение С) дает возможность роботу собирать кубик Рубика, начиная с любого запутанного положения. Кажется, что такие примеры имеют мало общего с производством, но на самом деле они позволяют продемонстрировать, как гибко можно запрограммировать робот на выполнение длинных и сложных последовательностей действий — более сложных, чем это посильно для большинства людей. Сборка кубика Рубика хорошо иллюстрирует способность робота приспосабливаться к данной ситуации, поскольку решения для каждой из $43 \cdot 10^{18}$ исходных комбинаций различны. Перспективы использования таких возможностей в робототехнике восхитительны.

7.4. ЯЗЫК АРМБЕЙСИК

Универсального языка программирования для роботов не существует. Язык ВАЛ является собственностью фирмы «Юнимейшн», а язык АРМБейсик принадлежит фирме «Микробот» [37]. Многие колледжи и университетские лаборатории оснащены роботами марок «Микробот Минимумер» и «Тичмувер», поэтому язык АРМБейсик может оказаться полезным для студентов в их лабораторных занятиях. Однако действительной целью ознакомления с этим языком, которая здесь преследуется, будет вооружение студента общим пониманием возможностей и ограничений, которые имеет язык робота, в качестве подготовки к работе на производстве.

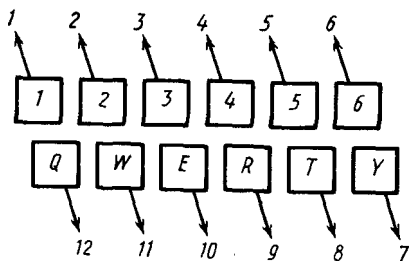
Язык для микроЭВМ Бейсик применяется почти на всех промышленно выпускаемых микроЭВМ, и для получения языка АРМБейсик достаточно заложить в память ЭВМ дополнительную информацию с магнитной кассеты или диска. Конечно, от языка АРМБейсик мало пользы, пока рука робота также не присоединена к микроЭВМ, поэтому типоразмерный ряд роботов модели «Микробот Минимумер» сконструирован именно для такого присоединения. Робот «Минимумер» и «Тичмувер», оснащенные шаговыми двигателями, относятся к роботам без обратной связи, поэтому на языке АРМБейсик программируется подача на их входы серий импульсов, как это было объяснено в гл. 2 и 6. Таким образом, материал этой главы охватывает языки программирования для обоих основных типов роботов — с обратной связью (ВАЛ фирмы «Юнимейшн») и без обратной связи (АРМБейсик фирмы «Микробот»).

Поскольку АРМБейсик используется для управления роботами без обратной связи, управляющая ЭВМ в действительности

Рис. 7.10

Клавиатура управления движением робота «Микробот Миннмувер-5»:

1 — вращение основания по часовой стрелке; 2 — подъем плеча; 3 — подъем локтевого сустава; 4 — разгибание запястья; 5 — вращение запястья по часовой стрелке; 6 — раскрытие захвата; 7 — закрытие захвата; 8 — вращение запястья против часовой стрелки; 9 — сгибание запястья; 10 — опускание локтевого сустава; 11 — опускание плеча; 12 — вращение основания против часовой стрелки



никогда не знает, где находится рука робота. Однако ЭВМ способна сохранять в памяти алгебраическую сумму импульсов, которые она подала на каждое из звеньев робота. Она хранит результаты подсчета этих импульсов в отдельных регистрах, имеющих для каждого из приводов осей. Эти регистры можно назвать регистрами положения, однако куда действительно переместится робот в соответствии с показаниями этих регистров, зависит от наличия препятствий на пути робота, превышения скорости перемещения или других причин, способных привести к неотработке шаговым двигателем части импульсов. Программист в любой удобный момент может обнулить регистры положения. Это обнуление можно рассматривать как проверку положения робота, если робот перед обнулением подходит к известной фиксированной точке.

Вместо создания специального языка, подобного языку Бейсик, в языке АРМБейсик применен подход, согласно которому команды роботу подготавливаются в виде подпрограмм на языке Бейсик. Это означает, что программист на языке АРМБейсик должен знать и язык Бейсик. В то же время отсюда следует, что программист может воспользоваться всеми стандартными возможностями, предоставляемыми языком Бейсик для решения с помощью ЭВМ совершенно других задач, по мере необходимости обращаясь к специфичным программам на языке АРМБейсик. Кроме того, вся программа редактирования, имеющаяся в распоряжении пользователя обычного языка Бейсик, действительна и во время программирования на языке АРМБейсик.

Команды или подпрограммы языка АРМБейсик удивительно немногочисленны и просты. Команд всего шесть:

@ RESET — обнуляет «регистры положения», устанавливая исходное положение для отсчета импульсов всех шаговых двигателей.

@ STEP — обеспечивает перемещения робота, подавая в требуемом режиме управляющие импульсы.

Пример: @ STEP SP, A, B, C, D, E, F. Эта команда предписывает роботу двигаться со скоростью, задаваемой переменной SP. Переменная A определяет, какое количество импульсов подается на первую ось робота (основание), переменная B — на вторую ось (плечевой сустав), C — на третью ось (локтевой сустав) и т. д. Переменная F относится к захвату робота (шестая ось). Движение программируется «в приращениях», т. е. перемещения по каждой из осей задаются от положения, непосредственно предшествующего команде @ STEP.

@ SET — переключает робот в ручной режим управления по командам с клавиатуры (рис. 7.10). По сути, это режим обучения для каждого звена робота.

@ READ — дает команду записать алгебраическую сумму импульсов, выданных на каждый из шаговых двигателей, начиная с предыдущей команды @ RESET.

Пример: @ READ A1, A2, A3, A4, A5, A6. По этой команде в регистрах положения записываются импульсы, подаваемые на шаговые двигатели приводов робота по осям, начиная с самой последней команды @ RESET, когда эти регистры были обнулены. В этом примере переменная A1 соответствует алгебраической сумме импульсов, поданных на двигатель основания, A2 — плечевого сустава и т. д. до A6 — захвата. К этим переменным можно позже обращаться, а импульсы могут быть поданы снова, чтобы возвратить робот в прежнее положение по команде @ STEP.

@ CLOSE — дает команду закрыть захват до тех пор, пока датчик не покажет, что рука закрылась с заданным усилием.

@ ARM — выбор номера порта ЭВМ, который будет использоваться. Применяется при наладке, а не при программировании.

Каждая команда на языке АРМБейсик предваряется знаком @, что отличает ее от обычной команды языка Бейсик. В языке имеется команда @CLOSE, но нет соответствующей команды OPEN (открыть). Открыть или закрыть захват можно, просто посылая серии импульсов на двигатель робота, открывающий и закрывающий захват. Это можно выполнить при помощи команд @STEP или @SET. Отличие от команды @CLOSE состоит в том, что вращение двигателя зажима будет происходить до тех пор, пока двигатель не отключится по команде от датчика на роботе (концевой выключатель на роботе модели «Микробот Минимумер 5»).

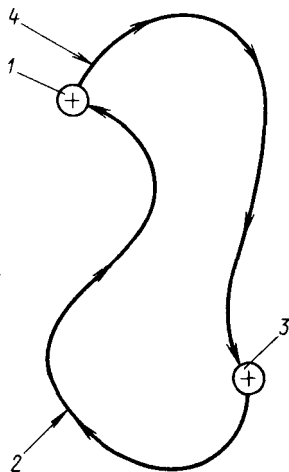
Может показаться, что при программировании на языке АРМБейсик от программиста потребуются выполнять подробные вычисления, включая тригонометрические преобразования с использованием размеров звеньев роботов и подсчет угловых перемещений для каждого импульса, поданного на двигатель, однако это не так. Благодаря наличию режима ручного управления, обеспечиваемого командой @SET, звенья робота можно привести в требуемое положение, после чего выполняется команда @READ. При этом алгебраическая сумма импульсов, поданных после последней команды @READ, запоминается для последующих манипуляций. В следующий раз, когда программист захочет вернуть робот в то же самое место, он использует команду @STEP, чтобы установить количество импульсов, подаваемое на каждую ось. Это количество равно разности между зарегистрированным количеством импульсов в исходном положении и в точке назначения. На рис. 7.11 приведено объяснение процедуры программы движения. По программе робот движется плавно, поскольку АРМБейсик обеспечивает подачу импульсов на шаговые двигатели в пропорции, гарантирующей плавное перемещение непосредственно из точки А в точку В. Обычно траектория не является прямой, но и не будет так изогнута, как показано на рис. 7.11. Там это сделано для наглядности.

Главное, что нужно усвоить из рис. 7.11, это то, что программисту не требуется подсчитывать и даже вообще знать действительное количество импульсов, зарегистрированных перемен-

Рис. 7.11

Использование возможностей языка АРМБейсик для сокращения количества непосредственно программируемых точек с использованием экономного количества в дальнейшем программировании:

1 — используя команду @ SET, введите координаты A1, A2, A3, A4, A5, A6, определенные при работе с клавиатурой в ручном режиме. Команда @ READ экономит координаты; 2 — для перемещения из точки A в точку B по программе используйте команду @ STEP SP. B1—A1, B2—A2, B3—A3, B4—A4, B5—A5, B6—A6; 3 — используя команду @ SET, введите координаты B1, B2, B3, B4, B5, B6, определенные при работе с клавиатурой в ручном режиме. Команда @ READ экономит координаты; 4 — для перемещения из точки B в точку A по программе используйте команду @ STEP SP. A1—B1, A2—B2, A3—B3, A4—B4, A5—B5, A6—B6



ными A1 — A6 и B1 — B6. Программист, указывая переменные, просто предписывает ЭВМ занести в свою память данные о количествах поданных импульсов. После этого, во время процесса обучения, программное обеспечение присваивает этим переменным значения, соответствующие подсчитанному количеству импульсов, и заносит их в память. Вся процедура происходит в системе без обратной связи, работоспособность которой обеспечивается только в том случае, если нет пропуска импульсов шаговым двигателем из-за попадания робота в препятствие или из-за слишком быстрого вращения. Если импульсы не будут отработаны, ЭВМ не узнает, что робот не находится в том положении, в котором он должен находиться по данным, имеющимся в ее памяти.

Легче всего это уяснить на примере. Подготовим программу для робота на языке АРМБейсик. Мы запрограммируем ту же операцию переноски деталей, для которой была составлена программа BETTER. А. ТО. В на языке ВАЛ. Прежде всего нужно оговорить переменные, значения которых будут определять отсчет шагов по всем осям робота для прихода в требуемую точку. Под отсчетом шагов подразумевается суммарное число шагов, отрабатываемых шаговым двигателем, которое требуется для прихода в данную точку из исходного положения, определяемого командой @RESET.

Пусть A_i — отсчет шагов по оси i до точки, где производится захват детали; B_i — отсчет шагов по оси i до точки установки детали; C_i — отсчет шагов по оси i для подхода к точке A; D_i — отсчет шагов по оси i для подхода к точке B.

Предположим, что имеется робот с шестью осями, у которого шестая ось — захват. Это означает, что имеется 20 переменных (4 точки $\times$ 5 осей), для которых нужно определить отсчет шагов из исходного положения. Кроме того, следует установить разность отсчета шагов между открытым положением захвата и положением, когда он зажимает предмет.

Самым практичным способом введения в управляющую ЭВМ значений этих переменных (в количестве 21) будет физическое перемещение робота в требуемые положения по команде @SET.

При этом по команде @READ после каждой команды @SET производится регистрация количества шагов. Эти действия похожи на обучение вручную, аналогичное части HERE A, HERE B программы BETTER.A.TO. В на языке ВАЛ. Правильной стратегией будет выбрать одну из точек A, B, C или D в качестве исходной. За счет этого количества переменных сократится с 21 до 16, т. е. на пять. Кроме того, сокращается объем арифметических операций в команде @STEP во время обработки программы. Подход к точке C при захвате робота, раскрытом достаточно для помещения в нем детали, выбирается как удобное исходное положение. Оно будет начальной и конечной точкой при каждом выполнении цикла. На языке АРМБейсик программа будет выглядеть следующим образом:

100 @ SET (перевести робот в ручной режим управления, чтобы оператор мог подвести его к точке C, точке подхода для захвата)

Оператор использует клавиатуру (в режиме управления звеньями робота), чтобы установить робот в точке C при захвате, раскрытом достаточно (с некоторым запасом), чтобы поместить в нем деталь, которую предполагается зажать.

Оператор нажимает клавишу 0, чтобы вывести систему из режима ручного управления.

110 @ RESET (команда фиксирует точку C как исходную)
120 @ SET (режим ручного управления)

Оператор использует клавиатуру, чтобы перевести робот в другую точку, на этот раз в точку A, затем нажимает клавишу 0.

130 @ CLOSE (зажать деталь в захвате)
140 @ READ A1, A2, (запомнить отсчет шагов по отношению к исходной
A3, A4, A5, A6 точке C. Отсчет шагов для зажима детали производится в переменной G)
150 @ SET (режим ручного управления)

Оператор использует клавиатуру для перевода робота в точку D, затем набирает 0.

160 @ READ D1, D2, (запомнить отсчет шагов по отношению к исходной
D3, D4, D5 точке, а не к точке A)
170 @ SET (режим ручного управления)

Оператор переводит робот в точку B, затем набирает 0.

180 @ READ B1, B2, (запомнить отсчет шагов по отношению к исходной
B3, B4, B5 точке)

Теперь робот знает все необходимые точки, включая точку C, принятую за исходную. Читатель может удивиться, почему команды @READ в предписаниях 140, 160 и 180 относятся только к пяти осям робота, а не к шести. Вспомните, однако, что шестая ось (захват) сохраняется в одном и том же положении во время

большинства перемещений. Таким образом достаточно было оставить всего 16 переменных: пять осей в трех положениях плюс одно значение G , соответствующее количеству шагов для закрытия захвата. Для выполнения программы нужно освободить перенесенный объект из захвата в процессе обучения и вернуться в исходное положение. Это осуществляется при помощи следующих предписаний:

```
190 @ STEP 50, 0, 0, 0, 0, —G (открыть захват, скорость 50)
200 @ STEP 50, D1—B1, (вернуться из положения В в положение D,
D2—B2, D3—B3, D4—B4, скорость 50)
D5—B5
210 @ STEP 50, —D1, —D2, (вернуться в исходную точку, скорость 50)
—D3, —D4, —D5
```

Сейчас робот готов к выполнению цикла переноски, который осуществляется по следующим командам на языке АРМБейсик:

```
220 @ STEP 50, A1, A2, A3, A4, A5 (переместиться в положение А)
230 @ CLOSE (закрыть захват, зажимая деталь)
240 @ STEP 50, —A1, —A2, —A3, —A4, —A5 (вернуться в исходную точку С)
250 @ STEP 50, D1, D2, D3, D4, D5 (подойти к точке D)
260 @ STEP 50, B1—D1, B2—D2, B3—D3 (переместить деталь в точку В)
270 @ STEP 50, 0, 0, 0, 0, —G (отпустить деталь)
280 @ STEP 50, D1—B1, D2—B2, D3—B3, (отойти)
D4—B4, D5—B5
290 @ STEP 50, —D1, —D2, —D3, —D4, (вернуться в исходное положение)
—D5
300 GOTO 220 (непрерывно повторять последо-
вательность действий)
310 END (конец программы)
```

Предписания 270, 280 и 290 идентичны предписаниям 190, 200 и 210. Действительно, предписание 190 можно заменить на GOTO 270, а предписания 200 и 210 можно исключить из программы. Как бы то ни было, как только оператор даст знать, что робот обучен последней точке (точке В), введя с клавиатуры 0 в предписание 170, робот немедленно перейдет к выполнению программы.

То, что было показано, является не более чем основами программирования на языке АРМБейсик для выполнения простой программы переноски деталей, подобной программе BETTER. А. ТО. В, подготовленной ранее. Чтобы сделать программу более практичной, необходимо:

1. Обеспечить вывод на экран дисплея указаний, помогающих оператору определять точки в фазе обучения и вводить 0, когда новые точки определены.

2. Запрограммировать паузу между окончанием обучения и началом выполнения цикла с тем, чтобы робот выждал, пока оператор подготовится к работе.

3. Дать пользователю возможность устанавливать количество выполняемых циклов.

Эти и другие удобства могут быть обеспечены при использовании обычного языка программирования Бейсик. Например,

паузу можно запрограммировать с помощью следующей подпрограммы на языке Бейсик:

```
350 FOR N=1 TO 2000
```

```
360 NEXT
```

```
370 RETURN
```

Эта подпрограмма выполняет только индексацию переменной *N*, чтобы протянуть время. Сколько времени пройдет в действительности, будет зависеть от скорости работы конкретной ЭВМ, предела индексации (в данном случае 2000) и, вероятно, длины программы, в которой подпрограмма будет использована. Обычно программист экспериментальным путем определяет для каждого случая требуемый предел индексации цикла. Для дальнейшего рассмотрения и использования в упражнениях, приведенных в конце этой главы, предположим, что подпрограмма, приведенная выше, обеспечивает паузу 1 с.

Чтобы запрограммировать включение или выключение внешнего устройства, например инструмента робота, программист может использовать следующие предписания на языке Бейсик:

```
OUT 250, NN (включить инструмент)
```

```
OUT 250, FF (выключить инструмент),
```

где переменные *NN* и *FF* представляют числовые значения (в двоичном исчислении), обозначающие включение и выключение соответственно.

В приложении *B* приведена сложная программа для робота, написанная на языке АРМБейсик студентом Центра робототехники и автоматизации Арканзасского университета. С помощью оборудования, спроектированного студентами, программа демонстрирует способности робота «Микробот Минимумер 5» собирать пожертвования, различать монеты, деревянные кубики и поднимать протестующий крик, если кто-нибудь забирает монету с площадки до того, как робот успевает наклониться и захватить ее.

В ы в о д ы

Именно программируемость обеспечивает роботам широкую универсальность и стимулирует творческие способности инженеров по автоматизации при использовании роботов в производстве. Процесс программирования промышленного робота может быть различным по сложности, начиная с простейшего проведения его инструмента в требуемой последовательности движений, которую робот затем будет воспроизводить, и кончая программированием компактного встроенного микропроцессора с использованием шестнадцатеричного кода.

Одной из основных проблем программирования робота является проблема перехода от системы координат робота к системе координат оператора и наоборот. Люди представляют пространство в понятиях «направо-налево», «вперед-назад» и «вверх-вниз»,

а робот представляет его себе в движениях осей (звеньев), необходимых, чтобы переместиться в данную точку. Еще одной системой координат является система координат инструмента. Для осуществления наиболее практичного и эффективного обучения и программирования промышленного робота система программирования должна обладать способностью переходить из одной системы координат в другую, всего систем три: система координат звеньев, инструмента и мировая система координат.

Наиболее мощные и универсальные методы программирования объединяют обучение с помощью панели обучения с обычным программированием. Чтобы проиллюстрировать взаимоотношения между режимом обучения и программированием с клавиатуры в системах, где используются оба режима, были использованы языки ВАЛ и АРМБейсик.

У п р а ж н е н и я и в о п р о с ы

7.1. На языке ВАЛ напишите программу для робота, который разгружает палету с установленными на ней бронзовыми отливками, размеры которых показаны на рис. 7.12. Отливки расположены равномерно в три ряда по 4 шт. в каждом. Зазор между отливками составляет 2 дюйма (50,8 мм).

7.2. Потребуется ли проводить обучение робота для реализации программы на языке ВАЛ, написанной при выполнении упражнения 7.1? Аргументируйте ответ.

7.3. Для условий, заданных в упражнении 7.1, подготовьте управляющую программу на языке АРМБейсик.

7.4. Какие точки требуется обойти с роботом в режиме обучения для реализации программы на языке АРМБейсик, написанной при выполнении упражнения 7.3?

7.5. Необходимо запрограммировать робота, чтобы он нанес на деталь клей в соответствии со схемой, изображенной на рис. 7.13. Устройство для нанесения клея, закрепленное на роботе, включается выходным сигналом, равным 1. Дополнительное количество клея наносится в критических точках (см. пять точек на рис. 7.13) за счет выстоя в течение 0,5 с над каждой из точек при включенном устройстве. Подготовьте программу на языке ВАЛ для выполнения этой задачи.

7.6. Напишите программу на языке АРМБейсик для операции нанесения клея, описанной в упражнении 7.5. Можно принять, что выход ЭВМ $\neq 250$ связан с устройством для нанесения клея, которое включается переменной NN и выключается переменной FF.

7.7. Графитовые диски, похожие на хоккейные шайбы, забираются с конвейера и подаются под пресс для обработки. Предположим, что входные и выходные сигналы в робототехнической системе для обработки дисков распределяются следующим образом:

Вход	Выход	Определение
1		Графитовый диск находится на конвейере положения для захвата
4		Плунжер пресса опущен или находится в движении
5		Деталь находится в захвате
	1	Аварийная сигнальная лампа
	2	Звуковой аварийный сигнал (звонок)

Подготовьте программу на языке ВАЛ, используя входные и выходные сигналы во время работы. (Безусловно, существует множество правильных решений)

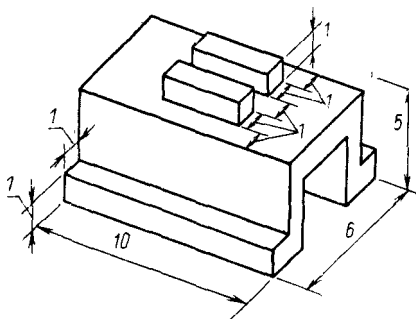


Рис. 7.12
Бронзовые отливки, разгружаемые роботом с транспортной палеты. Размеры даны в дюймах (1 дюйм = 25,4 мм)

этой задачи.) Сделайте так, чтобы звонки раздавались с интервалом 1 с в случае, если оператор не отреагирует на сигнальную лампу.

7.8. Промышленный робот загружает и разгружает ковочный пресс, манипулируя сильно разогретыми заготовками. Чтобы избежать перегрева захвата после каждого цикла загрузки робот погружает захват в ванну для охлаждения. Подготовьте программу на языке ВАЛ для реализации следующего цикла: захватить разогретую заготовку, поместить ее под пресс, подать сигнал на срабатывание пресса, охладить захват, разгрузить пресс, сбросить заготовку в бункер-палету и приступить к выполнению следующего цикла.

7.9. Время цикла ковочного пресса в упражнении 7.8 составляет 1,5 с. Чтобы вернуться к ванне для охлаждения и вернуться в исходную позицию для разгрузки пресса, роботу требуется 3,8 с. Для повышения общей производительности на захвате робота установлен датчик, подающий сигнал 1 (включить) на вход, когда температура захвата превышает определенный предел. Запрограммируйте робот на охлаждение захвата только в случае необходимости.

7.10. В упражнении 7.9 горячая заготовка, которую робот держит в захвате, активизирует датчик температуры, даже когда сам захват еще не перегрет. Объясните, как можно решить эту проблему с помощью соответствующего изменения программы.

7.11. Для развлечения к выходному порту 4 ЭВМ, управляющей роботом в упражнении 7.9, подключен синтезатор голоса. Он запрограммирован на подачу звука «Ах!» при появлении на выходе сигнала. Введите соответствующую команду или команды на языке ВАЛ, чтобы заставить робот говорить «Ах!», когда он охлаждает захват.

7.12. Для робота установлено десять точек, в которых он устанавливает электронные компоненты на печатной плате. Другие десять точек определяют исходные позиции, в которых робот получает эти компоненты. Подготовьте программу на языке ВАЛ, по которой робот устанавливал бы все десять компонентов по сигналу готовности от системы, подающей компоненты на исходные позиции. Установив десятый компонент, робот подает сигнал на исходные позиции. Установив десятый компонент, робот подает сигнал на систему, подающую компоненты.

7.13. Напишите программу установки электронных компонентов на печатной плате, пользуясь языком АРМБейсик.

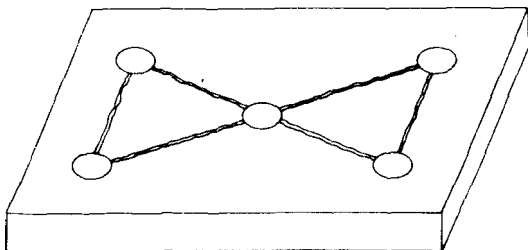


Рис. 7.13
Схема нанесения клея на деталь с помощью промышленного робота

8. ВНЕДРЕНИЕ РОБОТОВ

Во время работы над этой книгой автору довелось быть свидетелем контрольных испытаний роботизированного комплекса на заводе по выпуску двигателей, принадлежащем крупнейшему производителю США компании «Зингер» (г. Кларксвилл, шт. Арканзас). Представители компании собрались посмотреть на испытания прежде всего потому, что робот впервые был установлен на этом заводе, а кроме того проекты роботизации всегда привлекают повышенное внимание. Все было готово, и испытания начались. Прошло несколько секунд, и ошибка в последовательности действий привела к столкновению робота с макетом обслуживаемой установки, который был буквально опрокинут и упал в поставленные руки наблюдавшего инженера. Замешательство? До некоторой степени. Провал? Нет. Ключевым словом описанного случая является слово «макет». Испытания были имитацией реальных условий, и настоящее заводское оборудование было заменено макетом. Инженер на этом заводе обладал даром предвидения и опытом, которые подсказали ему, что автоматизация — дело хитрое и могут возникнуть непредвиденные осложнения. То, что могло привести к провалу проекта автоматизации и осмеянию его со стороны рабочих и руководства, вылилось просто в пересмотр последовательности действий в цикле, переналадку испытательного стенда и в повторные испытания. Нам всем необходимо следовать примеру, показанному инженерами по автоматизации, работающими в упомянутой компании, к чему и призывает эта глава.

Усвоив основные понятия о роботах и их программировании, изложенные в гл. 6 и 7, рассмотрим проблемы планирования, выбора, дооснащения, испытаний и установки роботизированных комплексов с целью обеспечения их успешного применения.

Рекомендации общего характера и концепции в словесной форме резко контрастируют с математической точностью, ассоциирующейся с программированием роботов, системами промышленной логики и другими более техническими вопросами, затрагиваемыми в этой книге. Однако принципы внедрения и подводные камни на пути к нему, рассматриваемые в этой главе, вероятно более важны, чем технические детали, раскрываемые в других

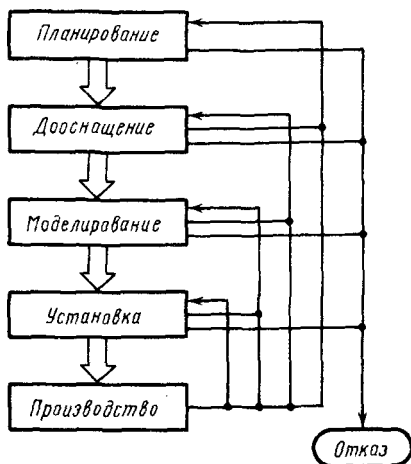


Рис. 8.1
Основные стадии процесса осуществления проектов роботизации

главах книги. Хотя разговор в этой главе пойдет о роботах, читатель увидит, что многие изложенные принципы применимы к любой автоматизированной производственной системе.

8.1. ОЦЕНКА ПРОЕКТА РОБОТИЗАЦИИ

После того как мы поняли, что такое робот, узнали о некоторых его возможностях и изучили основы его программирования, естественно возникает желание применить свои новые познания на практике где-нибудь на своем заводе или фабрике. Однако наилучшей политикой здесь будет осторожность. Роботы так восхитительны, а проекты реализации прогрессивной технологии так привлекательны, что появляется искушение рискнуть, купить робот и «делать что-нибудь, даже рискуя ошибиться». Многие фирмы так и поступили и, как и следовало ожидать, ошиблись. Серьезность проблем, возникающих при внедрении роботов и осуществлении других проектов автоматизации, почти всегда недооценивается. Проекты автоматизации имеют сверхъестественную способность следовать закону Мэрфи. Однако заблаговременное предвидение неизбежности трудностей автоматизации позволяет принять меры по сведению к минимуму отрицательных последствий, особенно если проблемы могут быть выявлены в лаборатории по автоматизации, а не на производстве. Наряду с этим тщательное планирование может умерить ожидания руководства, доведя их до уровня реально достижимых.

В процессе осуществления проектов роботизации, несмотря на большое разнообразие типоразмеров роботов, можно выделить пять основных стадий. Эти стадии и их взаимосвязь показаны на рис. 8.1, иллюстрирующем последовательность действий. Основная последовательность идет сверху вниз, как показано более толстыми стрелками, но на каждом уровне рассматриваются альтернативы вернуться к предыдущей стадии или отказаться от проекта.

Вначале большинство фирм пытаются пропустить одну или несколько стадий внедрения, показанных на диаграмме, но при этом им неизбежно приходится возвращаться по цепочке назад. Практически невозможно успешно пропустить какую-либо из стадий и при этом получить удовлетворительные результаты от внедрения.

8.2. ПЛАНИРОВАНИЕ

Стадия планирования требует для своей реализации соответствующего образования, но в этой главе предполагается, что основы знаний о типах роботов, их возможностях и методах управления известны. Если на заводе имеется только один специалист по робототехнике, разумно провести несколько внутренних семинаров, чтобы расширить кругозор других инженеров, техников и рабочего персонала в области применения роботов в производстве. Это способствует получению хороших результатов первого применения роботов на данном заводе.

Желательно пригласить ведущих представителей рабочих для участия в предварительных ознакомительных семинарах. Учитывая некоторые преимущества, получаемые от автоматизации и робототехники, вполне возможно, что представители рабочих положительно отнесутся к конкретному проекту роботизации. В то же время, если дать почувствовать рабочим неприязнь к их участию в роботизации, они скорее всего станут противниками этих нововведений. Представители рабочих, вероятно, являются наилучшим источником информации о том, что именно рабочие считают наиболее целесообразным автоматизировать. Несмотря на то, что точка зрения рабочих лишь одна из многих, иногда противоположных, ее ни в коем случае нельзя не заметить или отбросить.

Выбор потенциального объекта автоматизации. Важность первого шага нельзя недооценивать. В большей степени, чем при реализации любого другого типа проектов, рассчитанных на повышение производительности труда, успех или провал первой попытки установить робот на заводе интересует весь персонал. Не удивляйтесь, если информация просочится даже в прессу. Истории о роботах с хорошим или плохим финалом всегда оказываются в центре внимания. Используя имеющиеся знания основ робототехники, обычно можно выбрать подходящий объект автоматизации, если такой вообще имеется на изучаемом производстве. Следующие этапы стадии планирования выполняются применительно к этому объекту с сознанием того, что результат любого этапа может продиктовать решение отказаться от выбранного объекта в пользу лучшего.

Определение целей автоматизации (с точки зрения полезности). На рис. 8.2 приведена структурная схема стадии планирования. Этапом, следующим за выбором потенциального объекта автоматизации, является этап определения целей автоматизации с точки зрения получаемой выгоды. Такая рекомендация может показаться очевидной и банальной, однако цель осуществления проекта роботизации легко забывается. Иногда причина роботизации выглядит так: «Руководитель хочет попробовать как-нибудь использовать робот где-нибудь на заводе». Такое желание — действенная, хотя и не очень разумная причина, и если это так, что бывает часто, об этом следует открыто заявить и договориться

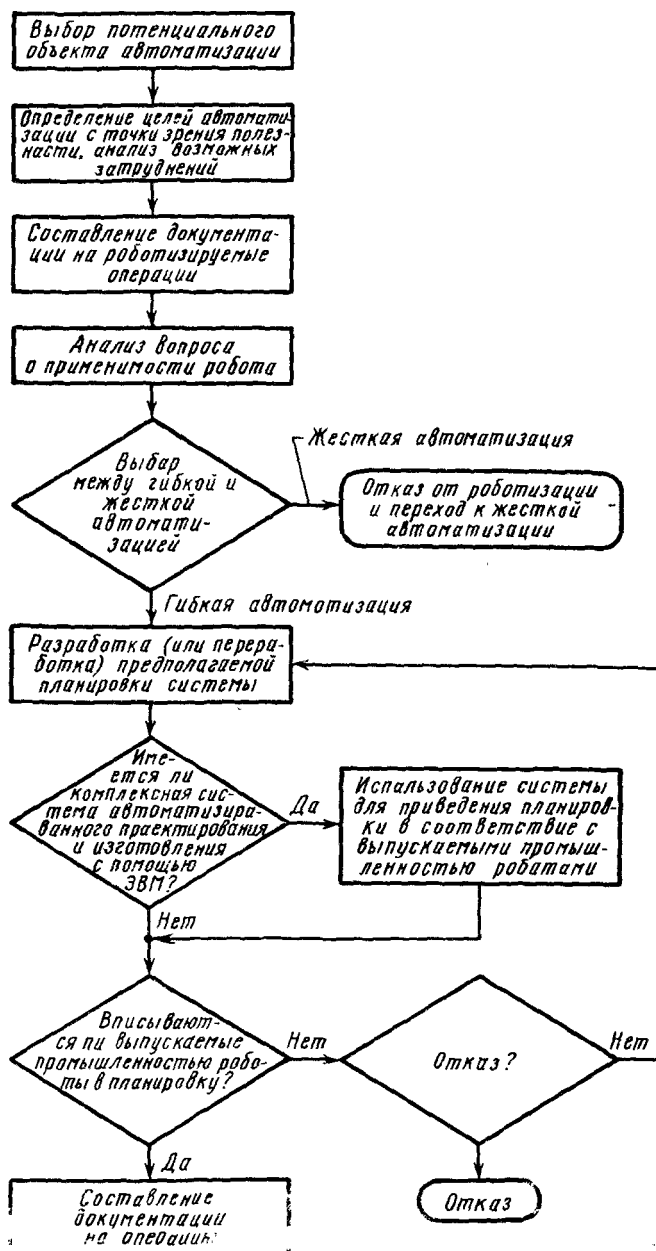


Рис. 8.2
Структурная схема стадии планирования

со всеми заинтересованными сторонами. Есть рациональное зерно во внедрении новой технологии в интересах получения образования или накопления опыта, однако лучше всего приобретать положительный опыт от осуществления успешного и плодотворного проекта, чем от такого, который остается чисто экспериментальным.

Более оправданной целью будет повышение производительности труда, сокращение стоимости рабочей силы, сокращение длительности цикла критической операции. Еще одной разумной целью будет исключение нежелательных работ или помощь в наиболее нежелательных моментах их выполнения. Обеспечение безопасности является действенным критерием, и проект может быть полностью посвящен освобождению рабочих от выполнения опасных или вредных работ, например, при воздействии вредных химических агентов и в других тяжелых условиях. Критерий обеспечения безопасности и защиты здоровья рабочих оправдывает даже повышение стоимости и сокращение производительности в результате проведения роботизации. Наряду с этим критерием может служить качество продукции или комбинации целого ряда задач. В этой начальной фазе предварительного планирования рискованно и, вероятно, просто бесполезно пытаться устанавливать значения количественных показателей, таких, например, как нормы прибыли от капитальных вложений.

Анализ затруднений. Определяя цели осуществления проекта, необходимо проанализировать потенциально возможные затруднения, возникающие при этом. Внедрение робота вероятно серьезно повлияет на судьбу одного или нескольких рабочих мест. Это влияние может сказаться:

- 1) положительно для фирмы, но отрицательно для служащих на ней;
- 2) отрицательно и для фирмы, и для служащих;
- 3) отрицательно для одного служащего, но положительно для большинства;
- 4) положительно и для фирмы, и для служащих.

Стремиться реализовать четвертую из перечисленных ситуаций — весьма благородная задача, но некоторые проекты роботизации достойны реализации даже, если несколько рабочих временно потеряют свои рабочие места. Что совершенно необходимо, это чтобы разработчики проекта учитывали его влияние на дальнейшие судьбы рабочих и заранее определились в этом отношении. Конкретно, если робот будет делать что-то, что сейчас делает рабочий, при подготовке проекта роботизации необходимо заранее подумать, что с этим рабочим делать после того, как проект будет реализован. Рабочий и сам будет думать о своем будущем, и если его вопросы останутся без ответа или, что хуже, ответы будут обескураживающими, на пути к успешному осуществлению проекта могут легко возникнуть значительные препятствия.

Обеспечение безопасности само по себе является преимуществом, которое можно получить, установив робот, однако и сам робот может представлять собой опасность. В фазе предварительного планирования самое время продумать, какие ограждения, блокировки, аварийные сигналы, отключения питания и другие меры понадобятся реализовать, чтобы сделать предполагаемое применение робота безопасным.

Необходимо непредвзято рассмотреть влияние реализации предполагаемого проекта на выполнение графиков производства и обслуживания. Иногда ожидается, что внедрение роботов немного снизит стоимость производства больших партий, а в действительности оно не оправдывает даже перерыва в выпуске, вызванного установкой и проведением испытаний. Наряду с этим следует рассмотреть возможности замены модели выпускаемого изделия, прекращения производства и изменения технологического процесса. Если эти проблемы проанализировать заранее, повышается вероятность создания более гибкой и реалистичной роботизированной системы.

Рассматривая проблемы производства и обслуживания, целесообразно разработать сценарии, охватывающие самые неблагоприятные сочетания обстоятельств, и подготовить планы действий в этих ситуациях. Продумайте следующие вопросы:

1. Что произойдет с системой в случае внезапного перерыва в энергоснабжении?

2. Будут ли «сожжены мосты» для возврата к производству продукции вручную после внедрения новой системы?

3. Осуществимо ли ручное производство во время остановки системы для ремонта или обслуживания?

4. Предположим, что качество поставляемых извне комплектующих изделий, с которыми имеет дело робот, ухудшилось. Будет ли система способна работать при этих отклонениях или хотя бы определить, что они имеют место?

5. Можно ли использовать робот в другой системе, если та, в которой он работал, по какой-либо причине прекратила свое существование?

В отношении движений робота не забудьте рассмотреть возможность возникновения ошибки функционирования и ошибки несрабатывания.

Ошибка функционирования. Предположим, что робот движется неправильно: совершает неправильные движения в нужное время, или движется правильно, но когда не нужно. Что произойдет?

Ошибка несрабатывания. Предположим, что робот не движется, когда это требуется. Что произойдет?

Любая ошибка может отрицательно сказаться на планах производства, качестве, привести к повреждению оборудования и даже к созданию опасной для персонала ситуации. При проектировании системы необходимо учитывать возможность возник-

новения обеих ошибок. С точки зрения качества, применение роботов обычно обеспечивает большую стабильность, вопрос, однако, состоит в том, будет ли качество в результате стабильно лучше или стабильно хуже?

Стабильность процесса в общем является предпосылкой высокого качества, но иногда роботы грубо обращаются с продукцией и могут повредить обработанную поверхность или разрушить хрупкие изделия.

Составление документации на роботизируемые операции. Если проект роботизации удалось довести до этого места фазы планирования, наступает время тщательно задокументировать существующую роботизируемую операцию. Эти данные будут полезны как при детальной разработке предлагаемой автоматизированной системы, так и для ее окончательной оценки. В документацию должны быть занесены планировки, длительность циклов работы оборудования, продолжительность операций, выполняемых вручную, данные о выполненных работах по обслуживанию и ремонту, данные, характеризующие производительность и качество. В документацию необходимо включить данные о конструкции изделия, такие, как размеры, допуски и другие показатели качества. Большинство из этих данных может уже иметься в архивах и даже храниться в памяти ЭВМ. Детальное изучение операций, требующееся для составления подробной документации, обычно позволяет выявить пути их некоторого усовершенствования еще до установки робота. Это служит аргументом в пользу рассмотрения проектов внедрения роботов, даже если в результате будет принято решение роботов не использовать.

Анализ вопроса о применимости робота. После составления документации на отобранные операции в их исходном виде следует приступить к изучению технических характеристик промышленно выпускаемых роботов, чтобы определить, подходят ли они для выполнения данного типа работ. Упор здесь следует делать на техническую сторону, а не на экономическую целесообразность. Необходимо обратить внимание на грузоподъемность, номинальную скорость, повторяемость и систему управления движением. В качестве предпосылки к принятию следующего решения — о выборе между жесткой и гибкой автоматизацией — имеет смысл в самом общем виде рассмотреть и экономические вопросы.

Выбор между гибкой и жесткой автоматизацией. На этом этапе следует тщательно проанализировать альтернативы применению роботов. Даже если уже просматриваются удовлетворительные решения, связанные с применением роботов, это не значит, что не существует еще лучших решений. Самые обычные технические решения проблемы автоматизации базируются на принципах жесткой или специальной автоматизации. Робот с его универсальностью и перепрограммируемой системой управления обеспечивает наибольшую гибкость, однако иногда при повышенной стоимости. Жесткая автоматизация также может оказаться

дорогой, дороже, чем роботы, если используется в высшей степени специальное оборудование.

Если серийность изделий не очень велика, жесткая автоматизация обходится особенно дорого. Жесткая автоматизация сильно обесценивается, когда выпуск данной продукции прекращается, и даже может обесцениться совсем. Промышленный робот, напротив, скорее всего сохранит свою ценность потому, что он может быть использован в изменившихся условиях.

В гл. 6 указано, что обычно роботы целесообразно применять не в крупносерийном, а в среднесерийном производстве. Скорость выполнения операций может служить важным критерием при принятии решения о типе автоматизации. Кроме собственно скорости, следует принимать во внимание возможность совмещения операций. Хотя робот может быть оснащен двойными захватами, а его система управления может выдавать команды на срабатывание других устройств, сами роботы, в общем, являются однофункциональными устройствами. Жесткая автоматизация часто оказывается лучше там, где целые комплексы одновременно выполняемых операций находятся под непосредственным управлением одного контроллера.

Эта книга в основном посвящена роботам и методам гибкой автоматизации, поэтому оставшаяся часть главы написана исходя из предположения, что на этой стадии планирования принято решение об использовании промышленного робота вместо приобретения оборудования для жесткой автоматизации. Однако принципы и пути рационального внедрения проектов автоматизации почти идентичны вне зависимости от типа автоматизации.

Предполагаемая планировка системы. Сделав выбор между промышленным роботом и жесткой автоматизацией, можно перейти к следующему этапу в процессе внедрения: разработать чертежи предполагаемой автоматизированной системы, включая робот, конвейеры, устройства подачи деталей и технологическое оборудование, с которым робот будет взаимодействовать. Эту работу целесообразно выполнить вне зависимости от того, будет ли робот оснащен захватами, сварочными головками, краскопультами или другими рабочими органами.

На операции, выполнявшиеся ранее, в том числе и ручные, была заблаговременно подготовлена документация, включающая чертежи планировок и технические данные. Имеющиеся чертежи могут служить хорошим исходным материалом для разработки планировки предполагаемой системы, но реализация процесса с установкой робота может измениться. Изменения могут быть минимальными и заключаться, например, в увязке рабочей зоны робота с технологическим процессом, а могут включать в себя и разработку нового подхода к процессу. На этом этапе планирования следует использовать возможность выявить и оценить совершенно новые подходы.

Разрабатывая планировку предполагаемого процесса, не забудьте уроки, преподанные в гл. 3, и обдумайте возможный вариант применения буферных складов. Даже если робот, предполагаемый к использованию, оценивается как в высшей степени надежный, окажется ли он надежным в нашем конкретном случае? Будут ли обладать такой же высокой степенью надежности устройства, связывающие конвейер с роботом и технологическое оборудование с роботом? Если оснащение буферным складом не предусмотрено, роботизированная система может иметь такое время простоев, что минимально допустимый уровень экономической эффективности производства не будет достигнут.

Как было выяснено в гл. 6, скорость не является основным достоинством промышленного робота, и в автоматизированной системе, где человек просто заменен роботом без изменения ее структуры, могут быть упущены реальные возможности для повышения производительности труда. Приведем в этой связи следующий реальный случай. При обслуживании ручную оператору не составляло труда доставать детали из-за небольшого ограждения, предназначенного для удержания деталей в требуемом положении на конвейере. Но для робота этот тип движения занимал больше времени и потребовал применения двойного захвата необычной конструкции. Обнаружилось, что ограждение может быть смещено на небольшое расстояние вдоль конвейера непосредственно у точки захвата, без ущерба его функциям, но чрезвычайно облегчая выполнение функций робота. Это якобы незначительное изменение весьма значительно повысило эффективность новой роботизированной системы.

Разработка планировки робототехнической системы является итерационным процессом, при котором пробный вариант планировки сначала сравнивается с характеристиками и размерами рабочей зоны выпускаемых роботов. Затем, если этот вариант не проходит, для проведения сравнения подготавливается новый вариант. В чем заключается итерационный процесс, показано на рис. 8.2, где сведены воедино все стадии планирования внедрения роботов. Подготавливая планировки, не забудьте включить в них, наряду с видами в плане, также размеры по вертикали.

Робот будет работать в трехмерном пространстве, и все размеры должны быть взаимосвязаны. Руководители проекта, спеша протолкнуть роботизированную систему в производство, могут и не увидеть необходимости в усердном прочерчивании предполагаемой системы в трех проекциях, однако инженер по автоматизации и робототехнике, желающий добиться успеха, найдет, что время на это тратится не зря.

Одним из способов экономии времени при итерационном процессе рассмотрения предполагаемой роботизированной системы и сравнения ее с геометрическими характеристиками выпускаемого робота является использование какого-либо пакета управ-

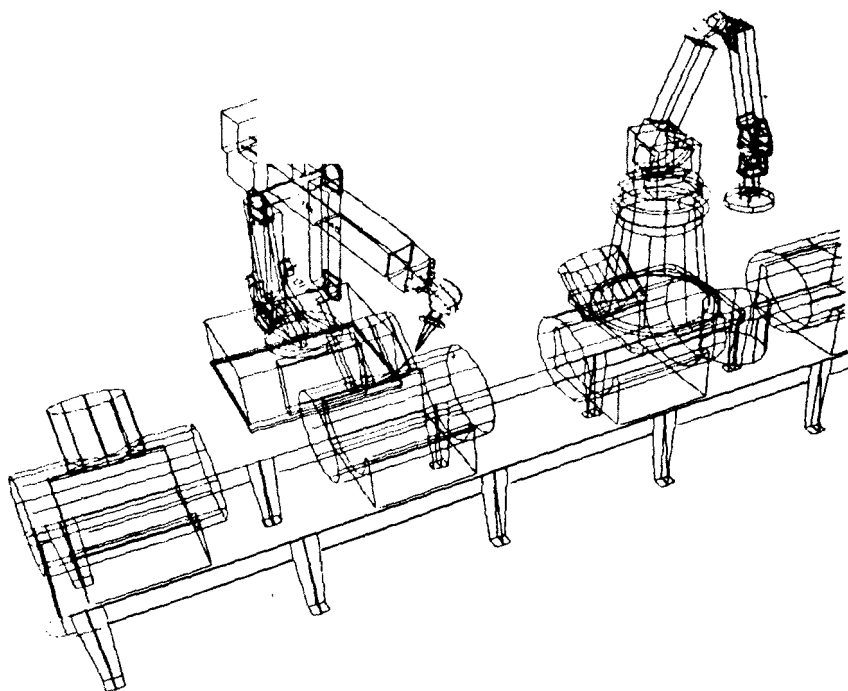


Рис. 8.3

Проволочные модели, используемые при автоматизированном проектировании роботизированных комплексов с помощью системы проектирования ПЛЕЙС

яющих программ для автоматизированного проектирования на ЭВМ, такого, как система ПЛЕЙС фирмы «Мак Авто». На рис. 8.3 иллюстрируется метод автоматизированного проектирования с помощью «проволочных» моделей, предназначенный для сравнения рабочих зон роботов и изучения потенциальных возможностей столкновения между машинами. Некоторые из таких пакетов включают в себя программное обеспечение, позволяющее представлять различные выпускаемые роботы на экранах дисплеев. Инженер по автоматизации просто вводит наименование того робота, который он хочет увидеть на экране, и «проволочная» модель этого робота немедленно возникает на базовом изображении производственной системы.

Для укомплектования документации предполагаемой роботизированной системы надо тщательно определить значения грузоподъемности, длительность операций и траектории звеньев робота. Эти характеристики могут быть позже изменены для приведения в соответствие с конкретной конструкцией робота, но знание

основных характеристик, являющихся целью разработки процесса, особенно длительности операций, может оказаться очень полезным в стадии дооснащения, которая следует после стадии планирования.

Последним предостережением в отношении стадии планирования внедрения роботов в целом будет предостережение против переусложнения. Это же предостережение распространяется и на следующую стадию — стадию дооснащения. Легко излишне переоценить возможности системы и взяться за выполнение нерентабельного проекта. При первой попытке использования робота на заводе не слишком увлекайтесь, особенно в отношении сложных систем очувствления и искусственного интеллекта. Скромный успех куда лучше, чем блестящий провал. Используйте простейший робот, соответствующий задаче, и в поисках простейшего решения не исключайте возможность того, что от робота совсем придется отказаться.

8.3. ДООСНАЩЕНИЕ

Четкие границы между стадиями планирования и дооснащения установить трудно, однако свидетельством того, что мы перешли к следующей стадии, является наличие выбранного основного оборудования. Применение роботов для сварки, окраски и практически всех других видов работ требует некоторого дооснащения связанного с ними оборудования, но особенно это справедливо в отношении роботов, манипулирующих штучными деталями. Наиболее важными областями дооснащения являются следующие связи робота: 1) с изделием, особенно со штучными деталями; 2) с транспортно-складским оборудованием; 3) с технологическим оборудованием.

Самой сложной проблемой обычно бывает разработка захвата, соответствующего детали. Изготовители роботов поставляют роботы без захватов и предоставляют тем самым пользователю возможность самостоятельно дооснастить руку. Пользователь знает свою собственную продукцию, знает, каким образом с этой продукцией обращались при обработке вручную, имеет информацию о предыдущих безуспешных попытках автоматизации обработки этих изделий и имеет возможность изменить их конструкцию. Учитывая важность аспекта конструирования захватов, он будет рассмотрен подробнее. Сначала проанализируем требования по стабильности процесса и других составляющих, определяющих связь оборудования с обрабатываемой продукцией, таких, как оснастка, приспособления и конструкция деталей.

Анализ стабильности процесса. Ключевым моментом в конструировании системы захвата робота является стабильность размеров захватываемых деталей. Вопрос качества выступает на первый план, если допуски размеров станут критичными по от-

ношению к способности робота захватывать деталь. Главной характеристикой служит не номинальный размер, а допуск. Кроме стабильности размеров, существует проблема стабильности позиционирования в смысле как ориентации, так и положения в пространстве. При использовании существующих конвейеров и крепежных приспособлений нужно оценить возможность оборудования с точки зрения позиционирования.

Анализ стабильности процесса может быть ограничен изучением документации, подготовленной для существующей операции, но обычно этим дело не заканчивается. Целесообразно при анализе изучить экстремальные случаи столкновения с бракованной продукцией и экстремальные случаи неправильной ориентации.

Задайтесь следующими вопросами:

1. Насколько часто робот сталкивается с перевернутой деталью?

2. Столкнется ли робот когда-либо со случаем потери деталей?

3. Могут ли сборки для робота поступить некомплектными или иметь неточно установленные детали?

4. Могут ли детали иметь непредусмотренные заусенцы или облой?

Анализ стабильности процесса может заставить разработчиков вернуться к стадии планирования, но проблемы стабильности вероятно можно решить с помощью изменения конструкции детали, конструирования соответствующей оснастки и зажимных приспособлений или использования оборудования для предварительной подготовки деталей.

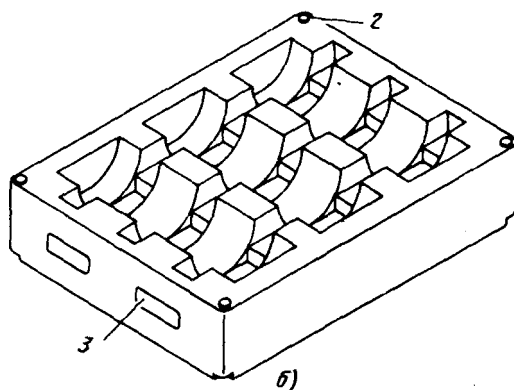
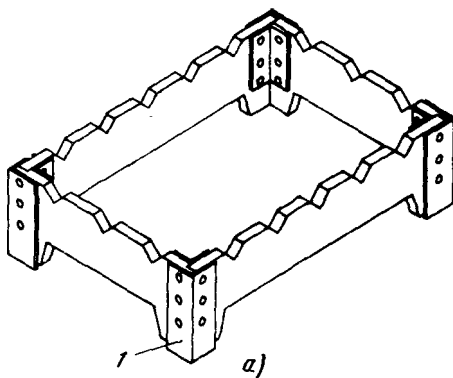
Изменение конструкции. В гл. 1 изложены основы конструирования изделий применительно к роботам и производственной автоматизации. Придерживаться этих базовых принципов — значит обеспечить успех программы автоматизации производства, но на данной стадии процесса внедрения робототехники разумно вернуться к этим принципам и определить, будут ли дополнительные изменения конструкции полезными для роботизированной системы. Особое внимание должно быть уделено согласованию деталей друг с другом или даже с оборудованием, которое будет выполнять над ними технологические операции. Например, для конструкторов исходного изделия совершенно очевидна необходимость установки конических штифтов для обеспечения соединения двух половин кожуха изделия. Но те же конструкторы могут не сделать коническим вал, который входит в отверстие в сборочной установке, что облегчило бы процесс сборки. На самом деле они могут вообще не предусмотреть такого вала, потому что он не нужен для выполнения функционального назначения продукции. Однако причины, кроющиеся в самом процессе изготовления, например в процессе роботизированной сборки, могут сделать изменение конструкции изделия очень полезным.

Конструирование оснастки и крепежных приспособлений. Наступление роботизации способствовало разработке специальных

Рис. 8.4

Образцы изготовленных под конкретные детали установочных приспособлений в виде палет (а) и лотков (б) из пластмассы, которые обеспечивают ориентацию деталей для их захвата роботом:

1 — металлические крепежные уголки, ориентирующие приспособления при укладке в штабель; 2 — штыри для ориентации при укладке пустых лотков в штабель; 3 — сквозные окна под вилы транспортера



палет и поддонов для сохранения как положения в пространстве, так и ориентации деталей и сборок, с которыми будет иметь дело робот. Разработка процесса штамповки из пенопласта сделала применение таких специальных установочных приспособлений эффективным и экономически оправданным. На рис. 8.4 изображены установочные приспособления палетного и лоткового типа, используемые при погрузке и разгрузке деталей роботами. К сожалению, специальные лотки и палеты могут оказаться слишком дорогими, чтобы применять их во время стадии дооснащения в процессе внедрения робота. Обычно из положения выходят, изготавливая установочные приспособления вручную из подходящих материалов, например гофрированного картона или дерева.

Конструирование захватов. Обратимся к конструированию непосредственно захватов. Захваты являются наиболее нагруженным элементом, и они подвержены сильному износу. Чтобы выйти из положения, предлагается изготавливать губки захватов из закаленной инструментальной стали, однако такие губки нельзя рекомендовать к использованию для стадий дооснащения и моделирования. Конструкция захватов обычно претерпевает ряд изменений в процессе внедрения робота. Поэтому для облегчения проведения экспериментов часто применяются мягкие металлы, чаще всего алюминий.

Конструкция захвата может обеспечить успех или погубить проект роботизации, поэтому важность этого вопроса нельзя не-

дооценивать. Конструкция изделия и выполняемые с ним манипуляции определяют тип и сложность применяемого захвата. Обычная загрузка и разгрузка установки требует, по всей вероятности, захвата простейшей конструкции. Но если задачей является манипулирование поковками, сложность возрастает, о чем будет рассказано в гл. 9 при обсуждении примеров применения роботов. Захваты наиболее сложных конструкций используются для автоматической сборки. Для таких захватов возможно использование нескольких осей движения и промежуточных положений, принимаемых под воздействием пружин. Кроме того, на захватах может быть закреплена вспомогательная оснастка.

При конструировании захвата робота следует рассмотреть возможность использования двойных захватов. Такие захваты наиболее эффективны при загрузке и разгрузке установок, когда робот одновременно манипулирует и заготовкой и обработанной деталью. Таким образом, за каждый подход к установке робот загружает и разгружает ее. Преимуществом такой стратегии является то, что робот может перемещать детали в другие места, пока установка производит обработку, вместо того, чтобы бесполезно простаивать перед ней, ожидая завершения цикла. Такие системы будут детально проанализированы в гл. 9.

8.4. МОДЕЛИРОВАНИЕ И ИСПЫТАНИЕ

До этого момента в процессе внедрения робота он сам еще не был приобретен. С покупкой его можно подождать еще некоторое время. Известно, что более или менее твердый выбор робота был сделан в конце фазы планирования. Результаты этого выбора были частично использованы при разработке захвата и соответствующих установочных приспособлений. Сам же робот понадобится на стадиях моделирования и испытания, но и на этих стадиях можно обойтись без его приобретения. Сначала необходимо создать испытательный стенд.

Испытательный стенд. Большинство промышленных роботов взаимодействуют с другим технологическим оборудованием, которое может оказаться слишком массивным, очень дорогим и слишком необходимым для осуществления существующего процесса, чтобы использовать его в испытательном стенде. Однако только часть каждой технологической установки необходима, чтобы испытать работоспособность внедряемого комплекса. Поэтому разумно изготовить модель или просто макет каждой установки, воплощающий только важнейшие ее черты. На рис. 8.5 изображена скелетная модель автоматической установки для намотки роторов электродвигателей, изготовленная на заводе компании «Зингер». Предполагалось осуществить роботизацию двух имеющихся в производстве установок, но ни одну из них нельзя было изъять из производства на время испытания. Единственным элементом намоточной установки, представлявшим интерес во время

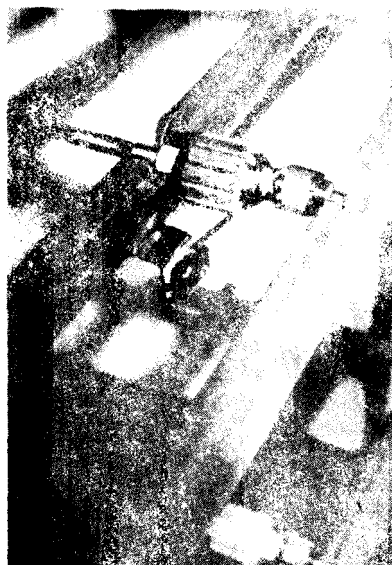


Рис. 8.5

Скелетная модель установочного приспособления автоматической установки для намотки роторов электродвигателей. Модель применялась при проверке возможности использования робота для загрузки и разгрузки установки. Справа показана имитация места установки детали на конвейере



Рис. 8.6

Применение лазера для контроля положения скелетной модели, установленной на испытательном стенде

испытаний, было установочное приспособление, на котором помещались роторы. Поэтому, как показано на рис. 8.6, для испытательного стенда потребовалась простая рама для закрепления приспособления.

Датчики и исполнительные механизмы. Промышленный робот редко работает самостоятельно без какого-либо взаимодействия со связанными с ним процессами. Концевые и путевые выключатели, фотоэлементы и другие датчики процессов должны быть предусмотрены в качестве датчиков входных сигналов для робота с тем, чтобы робот мог синхронизировать свои движения с процессом или изделием, которое он обслуживает. И наоборот, робот может передать процессу выходной сигнал, обуславливающий начало какой-либо функции. Наличие таких входных и выходных сигналов характерно не только для роботов, предназначенных для переноски и манипулирования деталями. Роботы для окраски, сварки и сборки также нуждаются в входных и выходных сигналах. Принципиально этот вопрос решается раньше, на стадии планирования, но выбор устройств для реализации продолжается и во время моделирования системы. Как выполняются логический анализ входных и выходных сигналов и управление соответствующими устройствами робота (обычно с помощью программируемых контроллеров), рассмотрено в гл. 10—12. Функционирование приборов, обеспечивающих входные и выходные сигналы, может быть проверено при помощи устройств программного управления еще до приобретения самого робота.

Испытания робота. Для проведения экспериментов и подтверждения пригодности на испытательный стенд должен быть установлен настоящий робот. Если фирма уже имеет в эксплуатации подобный робот, его, вероятно, можно временно включить в состав экспериментального стенда. Идя на это, необходимо учесть возможные последствия перебоев в производственном процессе, связанных с изъятием из производства работающего робота.

Некоторые университеты и частные лаборатории робототехники имеют центры роботизации, где представители промышленности могут опробовать варианты применения роботов, не закупая оборудование. В общем случае такие центры создают на кооперативных основах, когда представители промышленности ассигнуют средства на их основание, что позволяет университетам и лабораториям приобретать роботы, обслуживать их и поддерживать функционирование робототехнических центров. Такая организация позволяет пользователю свести к минимуму риск при капиталовложениях в роботизацию, внося при этом значительный вклад в дело образования инженеров и техников. Соприкосновение с настоящими промышленными роботами и сопутствующим оборудованием тем более важно для студентов, что роботы появились совсем недавно. Промышленные концерны в поисках выгоды от образования робототехнической лаборатории совместно с колледжем или университетом должны также принимать во вни-

мание льготы налогообложения при вложении средств в колледжи или университеты. Налоговое законодательство США начала 80-х годов ввело кредитование налогов на суммы, вкладываемые в колледжи, университеты и другие организации, не ставящие целью получение прибыли, когда эти фонды предназначаются на исследовательские цели. Здесь важно понимать различие: кредитование налогов значительно превосходит налоговую скидку.

При использовании институтского центра роботизации для проведения моделирования и испытаний данной робототехнической системы от фирмы-пользователя обычно требуется поставка испытательного оборудования, воспроизводящего технологическую операцию. Обычно оно состоит из образцов изделий, сопутствующей оснастки, приспособлений, захватов робота и макетов технологического оборудования. Центр обычно предоставляет сам робот, его систему управления и программируемый контроллер, управляющий всей системой. Наряду с этим он может предоставить лазеры и высокоточное оборудование для выверки, измерения и оценки. Студенты-техники могут разработать конструкции захватов по образцам изделий.

Организуя испытательный стенд в центре роботизации, иногда полезно выпустить чертежи планировки предполагаемой установки в масштабе 1 : 1. Такие чертежи можно разложить на полу или верстаке, где будет устанавливаться стенд для привязки оборудования к помещению.

Описанный подход применялся при организации испытаний установки для намотки роторов электродвигателей компании «Зингер», о которой рассказано выше. Испытания проводились в Центре робототехники и автоматизации Арканзасского университета.

Потенциальный пользователь роботов может пойти и по другому пути, опять же не покупая робот сразу. Как известно, некоторые поставщики роботов выдают потенциальным покупателям свои роботы на период около 60 дней. За это время целесообразность применения робота может быть полностью проверена и оценена. Если результаты удовлетворительны, робот покупают, в противном случае его можно вернуть.

Проведение экспериментов. Получив доступ к настоящему роботу, приобретенному или приобретаемому, можно предварительно несколько поэкспериментировать в ручном режиме, если робот оснащен соответствующим устройством. Используя панель обучения, оператор или техник может проиграть предполагаемый цикл вручную, не проводя обучения робота. Это может быть сделано вхолостую или с настоящим изделием, зажатым в захвате, или с краской в краскопульте, или с поданным сварочным током в зависимости от области применения робота. Большинство фирм предпочитают проводить предварительный эксперимент и ознакомление с роботом, не устанавливая модели дополнительного оборудования технологического процесса на испытательный стенд.

Следующим шагом в проведении экспериментов является программирование робота для замедленного выполнения предполагаемого цикла. Это будет соответствовать режиму обучения *A*, как было описано в гл. 7. Роботы с контурным управлением смогут быть запрограммированы с использованием замедленного режима. При работе в замедленном темпе вся робототехническая система, включая входные и выходные сигналы, связывающие робота с технологическим процессом, может быть проверена на соответствие задуманному.

После испытания робота в замедленном режиме, он может быть перепрограммирован на режим обучения *B*, т. е. его скорость может быть увеличена до нормальной. Это испытание является первым испытанием данной системы в полном масштабе и на рабочей скорости, поэтому его успех не гарантирован. Персонал все еще должен быть готов ко всяким неожиданностям, описанным в начале главы. Может понадобиться ручная подача роботу заготовок и отбор после обработки.

При проведении экспериментов опасность нахождения человека в рабочей зоне является наивысшей. Множество инженеров, техников, операторов и просто любопытных собираются посмотреть и помочь в опробовании нового робота. Границы рабочей зоны при этом четко не обозначены какими-либо барьерами.

Техникам часто требуется входить в рабочую зону для настройки клапанов, механических упоров, захватов или крепежных приспособлений, установленных на рабочем месте. Лица, обеспечивающие энергопитание и работу системы управления, в этот период должны быть уверены, что рабочая зона свободна во время действия робота. Использование надписей типа «Покиньте опасную зону» может способствовать обеспечению безопасности, но оператор, перед тем как включить робота, должен удостовериться, что требование освободить рабочую зону действительно выполнено.

Оценка повторяемости. Как только рабочий цикл полностью отлажен для работы в реальном масштабе времени, его необходимо осуществить несколько раз подряд для проверки повторяемости. До того как перенести выполнение операции в производственные условия, проверку следует провести на испытательном стенде. Иногда обнаруживается, что роботизированная система может отлично отработать несколько циклов подряд на полной скорости, а затем начинают возникать проблемы. Во время многократных испытаний необходимо часто проверять выверенность оборудования и контролировать повторяемость робота.

Полезным прибором для выверки испытательных стендов и контроля повторяемости роботов является маломощный лазер. В приведенном выше примере, относящемся к намотке электродвигателей на заводе фирмы «Зингер», допуски на взаимное расположение были малы, и для контроля малейших изменений положения объектов во время испытаний применяли лазер (см.

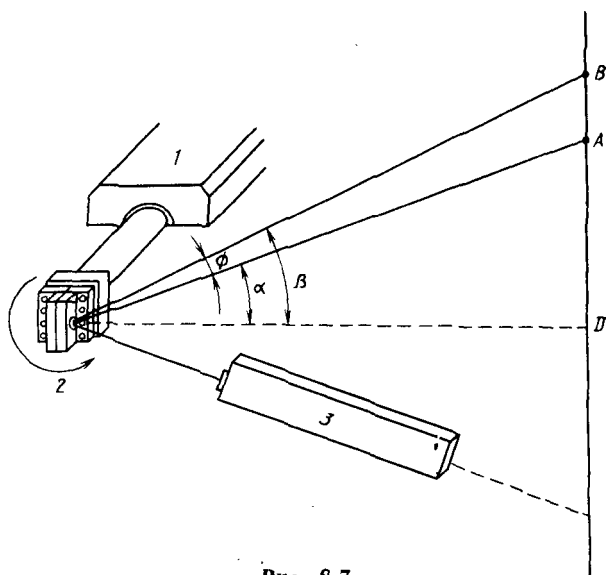


Рис. 8.7

Луч лазера позволяет значительно увеличить разрешающую способность оптического устройства для контроля углового положения частей испытательного стенда. Точки A и B являются маленькими световыми пятнами, образованными лучом лазера, отраженным от стоматологического зеркала, закрепленного на руке робота. Расстояние от робота до стены, на которую отбрасывается свет, может достигать 40 футов (12,2 м), причем чем больше это расстояние, тем больше разрешающая способность устройства. Расстояния AB , AD , BD и CD могут быть легко измерены:

1 — рука робота; 2 — запястье руки поворачивается вокруг точки C ; 3 — гелиево-неоновый лазер мощностью 0,95 мВт

рис. 8.6). На каркасе модели установки было смонтировано обыкновенное стоматологическое зеркальце, а луч лазера, отраженный от него, для увеличения разрешающей способности направили на стену лаборатории. Принцип работы проиллюстрирован на рис. 8.7. Положение маленького светового пятна на стене, образуемого лучом сильно коллимированного света от лазера, попросту отмечали карандашом, чем регистрировалось исходное положение (точка A). В любой момент испытаний можно проконтролировать положение светового пятна от лазера по отношению к карандашной отметке. Выбранный для этой цели лазер имеет малую мощность (0,95 мВт), так что обслуживающий персонал находится в безопасности даже при пересечении луча, конечно, если никто не смотрит прямо по лучу на источник света. Следующий пример иллюстрирует увеличение разрешающей способности, обеспечиваемое лазером, когда им пользуются для выверки.

Пример 8.1. Проверка точности позиционирования робота с использованием оптического рычага.

Испытательный стенд для робота установлен в большой комнате. Захват робота находится на высоте $h = 3,5$ фута (1 м) от пола. Маломощный лазерный луч

направлен на стоматологическое зеркальце, жестко закрепленное на захвате, и отражается на стену, расположенную на расстоянии $a = 40$ футов (12 м). В исходном положении робота луч света от лазера падает на стену на высоте $h_1 = 6$ футов (1,8 м) от пола. После отработки $n = 200$ циклов, включая $n_1 = 200$ поворотов (на 180°) по пятой оси (поворот запястья), робот возвращается в исходное положение цикла. На этот раз световое пятно оказывается на $h_2 = 6$ дюймов (152 мм) выше, чем перед началом испытаний.

Принимая, что зеркало жестко связано с роботом, а положение источников света осталось неизменным, определите, какое произошло смещение в угловом положении запястья.

Решение. В соответствии с рис. 8.7 можно определить (1 фут = 12 дюймам):

$$A = h_1 - h = 6 - 3,5 = 30 \text{ дюймов (762 мм);}$$

$$B = A + h_2 = 30 + 6 = 36 \text{ дюймов (914 мм);}$$

$$C = a = 40 \text{ футов} = 480 \text{ дюймов (12 192 мм);}$$

$$\varphi = \alpha - \beta = \arctg \frac{BD}{CD} - \arctg \frac{AD}{CD} = \arctg \frac{36}{480} - \arctg \frac{30}{480} = 0,71^\circ,$$

угол поворота зеркала

$$\theta = 0,5 \cdot 0,71 = 0,36^\circ.$$

Причесывание. Термин причесывание (*tweaking*), вероятно, появился в среде студентов инженерных специальностей, засиживавшихся допоздна за последними приготовлениями накануне сдачи курсового проекта по робототехнике. Суть его заключается в использовании метода проб и ошибок при внесении мелких усовершенствований в цикл работы робота с целью выжать из установки наивысшую возможную производительность. Некоторые инженеры пренебрегают этим приемом, отдавая предпочтение аналитической оптимизации. Однако в промышленности «причесывание», или назовите этот прием как хотите, является обычной процедурой при решении возникающих проблем. Промышленность, подобно студенту накануне сдачи проекта, часто не имеет времени для теоретических изысканий или совершенствования формул, особенно предназначенных для введения незначительных усовершенствований в цикл, направленный на повышение производительности труда. Когда установка работает на грани оптимальности, метод проб и ошибок или «причесывание» может оказаться самым быстрым и эффективным методом тонкой отладки процесса.

8.5. УСТАНОВКА

Испытания, отладка, оценка и «причесывание» закончены, и роботизированный технологический комплекс наконец готов к установке в цеху. Если планировка, моделирование и испытания проводились правильно, подлинная установка не вызовет затруднений. Так и должно быть, если мы хотим, чтобы линия не простаивала, а обслуживающий персонал как можно меньше отвлекался от работы во время наладки. Установка робота всегда привлекает внимание на производстве.

8.1. Данные к сетевому графику установки робота

Работа	Время, необходимое на выпол- нение работы	Срок начала работы	Срок окон- чания работы	Обязатель- ный срок окончания работы	Резерв времени
Разработка плани- ровки.	3 дня	12.07	15.07	15.07	Нет
Вскрытие контейне- ра, проверка содер- жимого	3 дня	12.07	15.07	15.09	33 дня
Очистка площади под установку	3 дня	12.07	15.07	15.07	Нет
Демонтаж коробов воздуховода конди- ционера	2 дня	15.07	19.07	19.07	Нет
Подвод электропи- тания	8 недель	19.07	14.09	14.09	Нет
Подвод водоснабже- ния и отвод воды в градирню	3 недели	15.07	9.08	14.09	5 недель
Подготовка основа- ния, заделка фунда- ментных болтов (включая двухне- дельную выдержку), грунт класса А	4 недели	15.07	12.08	6.09	16 дней
Обеспечение грузо- подъемным оборудо- ванием	2 недели	12.07	26.07	5.09	29 дней
Установка робота	1 день	12.08	15.08	7.09	16 дней
Подключение водо- снабжения и электро- питания	3 дня	14.09	19.09	19.09	Нет
Установка огражде- ния	1 неделя	15.08	22.08	15.09	17 дней
Установка блокиров- ки ограждения	2 дня	22.08	24.08	16.09	17 дней
Оборудование осве- щения	3 дня	15.07	20.07	19.09	42 дня
Подписание докумен- тации, заседания, де- монстрация работы	2 недели	15.09	29.08	21.09	16 дней
Предпусковые испы- тания	2 дня	19.09	21.09	21.09	Нет

Примечание. Учитываются только рабочие дни. Воскресенье, субботы и понедельник 5 сентября (День труда) исключены.

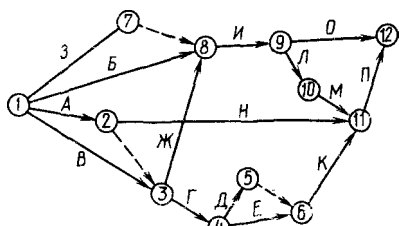


Рис. 8.8

Сетевой график установки робота в лаборатории Арканзасского университета

Обычные приемы организации работ могут оказаться полезными при планировании, управлении и руководстве роботизированным комплексом. К ним относится система планирования и руководства разработками ПЕРТ и метод критического пути, которые применимы при выполнении любого проекта, не только проекта роботизации. Однако из-за срочности, обычно сопутствующей

установке робота, ценность этих приемов, облегчающих проведение работ, возрастает. В табл. 8.1 приведены данные по работам сетевого графика, которые в действительности имели место при установке модели НТ⁸ фирмы «Цинциннати Милакрон» в лаборатории Арканзасского университета. Соответствующий сетевой график изображен на рис. 8.8. Проведение реальных расчетов критического пути выходит за рамки данной книги и является предметом изучения в другом курсе. По этому вопросу рекомендуем обратиться к [45].

Последней предосторожностью на стадии установки робота будет проверка: не попал ли случайно общий выключатель системы в рабочий объем робота. Известен случай, когда настенный телефон был установлен на внутренней стороне ограждения, проходившего по периметру рабочей зоны промышленного робота. Первоначально телефон стоял вне рабочего пространства робота, и перенос имел свой смысл, потому что позволял технику, обслуживающему установку, при поиске неисправностей, легко устанавливать контакт с инженером, обеспечивающим обслуживание во время эксплуатации. В то же время переустановка имела и обратную сторону, поскольку телефон может зазвонить и во время выполнения операций роботом, что вызывает искушение у рабочего персонала войти в запретную зону, чтобы ответить на звонок.

Опытная эксплуатация. Просто установить робот — это только часть дела. Новый комплекс должен некоторое время поработать в опытном режиме, чтобы можно было отработать технический процесс. Это может показаться дублированием испытаний на модели, но следует помнить, что опытная эксплуатация происходит на реальном технологическом оборудовании и с реальным сырьем. «Опытная эксплуатация» подразумевает производство в сокращенных объемах или при пониженной производительности. Это может оказаться недопустимой роскошью, если технологическое оборудование установлено в «узком месте» и работает с полной загрузкой. Если осуществление опытной эксплуатации невозможно по экономическим причинам, необходима организация работы

в параллель, хотя бы и при полной производительности. Под «работой в параллель» мы подразумеваем одновременную работу новой автоматизированной системы и старой, ручной, до тех пор, пока новая себя не зарекомендует. В некоторых случаях параллельная работа может быть просто неосуществима, и автоматизированное производство в полном масштабе и без всякой подстраховки становится неизбежным. В этом случае большая доля ответственности ложится на предыдущие стадии внедрения: моделирование и испытание.

Производство. Даже когда роботизированный комплекс полностью вошел в строй, его не следует оставлять без внимания. Продолжать учет его деятельности требуется прежде всего, чтобы выявлять проблемы, требующие решения, а кроме того, не следует упускать из вида необходимость иметь информацию для принятия решений по новым проектам роботизации. Когда роботизированное производство уже достаточно поработало, следует ответить на следующие прямые вопросы:

1. Достигнуты ли первоначально поставленные цели?
2. Если они не достигнуты, оказалась ли роботизация экономически эффективной?
3. Отличается ли качество выпускаемой продукции при новой организации производства от того, что было прежде? Если да, стало оно лучше или хуже?
4. Подтвердила ли зарегистрированная продолжительность исправного состояния достижение установленных показателей по надежности и ремонтпригодности?

8.6. ОТНОШЕНИЕ РУКОВОДСТВА И РАБОЧИХ

В этой главе явно обойден вопрос об одобрении руководством и поддержке рабочими осуществляемого проекта роботизации. В начале этой главы мы предостерегали против автоматического одобрения руководством сразу на весь период проведения работ — тот случай, когда руководитель просто хочет внедрить робот. Длительный успех проекта роботизации базируется на его очевидной технической и экономической обоснованности.

Если установка робота происходит на предприятии первый раз, будет разумным заручаться одобрением руководства на каждой из пяти стадий внедрения. При таком подходе руководство каждый раз может утвердить только некоторую часть проекта. Это, в свою очередь, обяывает исполнителей подтверждать жизнеспособность проекта на каждой стадии по мере его осуществления. Каждый раз, собираясь произвести расходование средств фирмы, необходимо обращаться к критериям капиталовложений, таким, как нормы прибыли. Завоевать признание рабочих, может быть, сложнее, чем получить одобрение руководства, но решение этой задачи можно облегчить, если открыто обращаться к ним за идеями и предложениями по проекту. Есть существенная разница

в поиске признания рабочих «лишь бы они не были против» или с искренним желанием получить их предложения по совершенствованию. Рабочие в целом больше знают о процессах, которыми они управляют, чем инженеры и техники, пытающиеся внедрить робот. Знание и опыт этих рабочих можно использовать для того, чтобы изменить, усовершенствовать, ограничить или изменить направленность проекта роботизации. При этом они будут гордиться соавторством, если, конечно, все получится удачно. Преимуществом этой стратегии является то, что при таком подходе проект будет выполнен более технически грамотным и одновременно получит одобрение рабочих.

Сопротивление рабочих проведению роботизации часто имеет серьезное основание. Если кому-то грозит потеря работы из-за того, что начальнику, технику или инженеру захотелось вдруг установить робот у себя на заводе, не увязав этот шаг с экономической целесообразностью его для фирмы, рабочие имеют полное право поставить под сомнение рациональность такого проекта. Это является вопросом этики в автоматизации, темой, которая будет обсуждена в гл. 15.

Сопротивление роботизации со стороны состава мастеров может вызвать недоумение. Мастера относятся к руководящему составу, однако когда дело доходит до роботизации, реакция мастеров сродни реакции рабочих, потому что они могут почувствовать в ней такую же угрозу для себя. Автоматизация может привести к сокращению рабочей силы на предприятии, к уменьшению количества рабочих, руководимых мастером и, в свою очередь, к снижению его статуса. Роботы и автоматизированное оборудование могут оказаться более технически сложными, чем прежнее оборудование, и мастер может почувствовать недостаточную квалифицированность, чтобы принять на себя ответственность за них. Аналогично тому, что говорилось ранее о рабочих, мастера обладают значительными познаниями и опытом в области технологического процесса, которым они руководят, и в своей области они могут сделать неоценимый вклад в совершенствование проекта роботизации.

8.7. ТЕХНИЧЕСКОЕ ОБУЧЕНИЕ

На ранней стадии функционирования новых роботизированных технологических комплексов техники, которые устанавливали робот, обычно остаются ответственными за его работу, изменение программы и обслуживание. Для передачи их знаний по управлению и обслуживанию персоналу, который обычно этим занимается в зоне, где установлен робот, организуется техническое обучение на рабочем месте. Иногда производители роботов считают целесообразным содержать школы для обучения рабочего и обслуживающего персонала обращению с выпускаемым оборудованием. Особенно разумно это по отношению к техническому обслужива-

нию. Здесь требуется оговорка. Не следует ожидать, что производители роботов обеспечат начальное техническое обучение обслуживающего персонала по таким вопросам, как основы гидравлики и электроники. Некоторые пользователи роботов посылают на курсы новичков, не обученных техническому обслуживанию и не имеющих в этом вопросе опыта, рассчитывая, что они за несколько дней станут специалистами. На деле же так не получается.

Часто требуется проводить обучение программированию робота, особенно если робот имеет режим программирования с клавиатуры. Опытные техники или инженеры лучше программируют с клавиатуры, чем операторы или обслуживающий персонал. Если персонал не обладает опытом программирования роботов до начала стадии планирования, ему необходимо пораньше начать проходить курс обучения, который производитель покупаемого робота организует для потребителя.

В ы в о д ы

Многие руководители и инженеры на производстве преждевременно обращаются к внедрению роботов. Легко переоценить возможности роботов, особенно когда имеешь дело с необычным объектом производства или с необычной разновидностью технологического процесса.

При систематическом подходе к внедрению роботов этот процесс разбивается на пять стадий, а именно: планирование, дооснащение, моделирование, установка и производство. Процесс до некоторой степени итеративен, с возможностью возвращения на каждой ступени к более ранним стадиям. Не исключена возможность полного отказа от задуманного ими обращения к жесткой автоматизации. Если даже от проекта в конце концов отказались, само внедрение стадия за стадией дает ценную информацию и способствует повышению эффективности и стабильности технологического процесса. Иногда эти побочно получаемые выгоды более существенны, чем прямые, определяемые непосредственно применением робота.

Поскольку наличие промышленного робота не является обязательным на протяжении большей части процесса внедрения, стадии планирования, дооснащения и испытаний часто могут быть выполнены в сторонних лабораториях или в университетских центрах робототехники и автоматизации. Такие центры могут служить идеальной базой для экспериментирования и предоставить необходимые приборы для контроля времени цикла и повторяемости процесса. Например, маломощные лазеры с успехом применяют для контроля положения робота после отработки нескольких циклов. Эти центры располагают также совершенными таймерами, программируемыми контроллерами и штатом исследователей-студентов.

Вне зависимости от того, происходит ли экспериментальное апробирование и внедрение новой идеи роботизации на месте или в сторонней лаборатории, как руководство, так и рабочие должны быть осведомлены о целях проводимых работ и о состоянии дел. Причастность к проекту не должна ограничиваться простой оценкой происходящего, должен быть обеспечен свободный обмен идеями и возможность активного участия. Рабочие — операторы технологических процессов, подлежащих автоматизации, и их непосредственное начальство — мастера обладают таким опытом и знанием выпускаемой продукции и технологического процесса, который не могут иметь техники и инженеры, непосредственно ответственные за разработку проекта роботизации. Участие рабочих и мастеров во внедрении принесет также свои дивиденды на стадии производства, когда производственный персонал нужно обучить управлению и обслуживанию нового роботизированного технологического комплекса.

Упражнения и вопросы

- 8.1. Перечислите пять основных стадий внедрения роботизации.
- 8.2. Какие предпосылки рекомендуется иметь для осуществления пяти основных стадий внедрения роботизации?
- 8.3. В какой момент внедрения к ней следует подключать ведущих представителей от рабочих?
- 8.4. Почему успех первого опыта внедрения роботов важнее, чем последующие?
- 8.5. На какой стадии внедрения следует приобрести робот?
- 8.6. Какие основные препятствия к осуществлению проекта роботизации могут возникнуть на стадии планирования?
- 8.7. Какие преимущества можно получить, задокументировав старый технологический процесс, выполняемый вручную, перед тем как установить новый робот?
- 8.8. В чью пользу сопоставление между роботизацией и жесткой автоматизацией, если говорить о скорости?
- 8.9. В чем потенциальное преимущество оснащения роботизированного технологического комплекса буферным складом?
- 8.10. Какие три основные связи робота с остальным оборудованием можно выделить?
- 8.11. Почему изготовители роботов предоставляют потребителям возможность самим оснащать роботы захватами?
- 8.12. Что представляет собой анализ стабильности процесса и почему он так важен для внедрения роботизации?
- 8.13. Какие два вида стабильности выпускаемой продукции имеют особое значение для внедрения роботов?
- 8.14. Приведите несколько примеров дефектов деталей или случаев неправильной ориентации, которые могут сделать роботизированный технологический комплекс бесполезным.
- 8.15. Почему в качестве материала для изготовления экспериментальных конструкций захватов часто используют алюминий?
- 8.16. Какая область применения роботов требует захватов наиболее сложных конструкций?
- 8.17. Каким образом использование двойных захватов сокращает время цикла работы робота при загрузочно-разгрузочных операциях?
- 8.18. В чем заключается первоочередное использование лазера во время внедрения роботизированного технологического комплекса?

8.19. Луч неподвижного лазера, отражаясь от маленького зеркальца, закрепленного на роботе, образует световое пятно на стене. Зеркальце находится в 50 футах (15,25 м) от стены. Если основание робота поворачивается по часовой стрелке до положения, когда отраженный луч лазера перпендикулярен стене, пятно на стене оказывается на четыре фута (1,22 м) в стороне от исходного положения. После того как робот осуществил несколько операций «поворот основания» и затем вернулся в исходное положение, оказалось, что пятно сдвинулось на 12 дюймов (30,5 см) по горизонтали, что говорит об увеличении угла, под которым отражается свет. На какой угол сместилось исходное положение основания робота, если лазер, зеркальце и другое оборудование остаются надежно закрепленными?

8.20. Что такое «причесывание»?

8.21. Что такое «ПЕРТ» и метод критического пути и какое отношение они имеют к внедрению робота?

8.22. Какие существуют две альтернативы опытно-промышленной эксплуатации роботизированного технологического комплекса?

8.23. Почему мастера часто противятся внедрению робота на своих участках?

9. ИСПОЛЬЗОВАНИЕ РОБОТОВ В ПРОМЫШЛЕННОСТИ

Эта глава является обзором реальных примеров применения роботов различных типов. Она призвана стимулировать зарождение идей, относящихся к роботизации производства, на котором работает читатель, а также помочь при оценке осуществимости этих идей.

9.1. СВАРКА

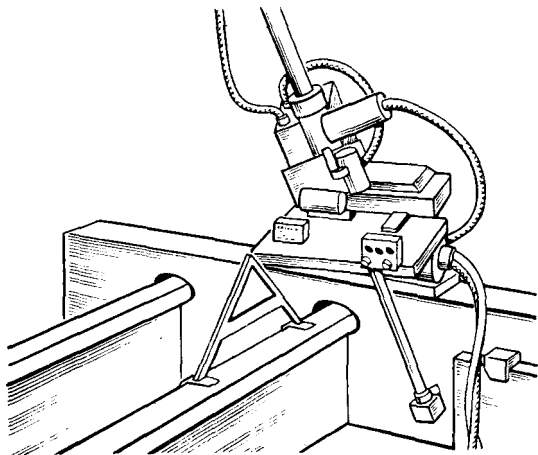
В первую очередь мы обратимся к сварке, потому что в середине 80-х годов именно она являлась основной областью применения промышленных роботов. Наиболее широко распространена автоматизированная точечная сварка, характерная для автомобилестроения, тяготеющего к крупносерийной конвейерной сборке, однако наблюдается рост применения роботов и для дуговой сварки.

Как точечная, так и дуговая сварка при работе вручную неблагоприятны с точки зрения техники безопасности. Кроме того, сварка непривлекательна для рабочих из-за необходимости носить защитную одежду, особенно при дуговой сварке. При отсутствии роботов персоналу приходится вручную управляться с тяжелыми грузами и с особенно тяжелым мощным оборудованием для точечной сварки. Не в последнюю очередь использование роботов привлекает возможностью обеспечить качество и высокую стабильность производства как при точечной, так и при дуговой сварке.

Точечная сварка наиболее широко применяется на линии по изготовлению автомобильных кузовов. Поскольку линия находится в состоянии постоянного движения, от работающих в ее составе роботов для точечной сварки требуется способность автоматически отслеживать движение конвейера.

Одним из примеров применения дуговой сварки является сварка элементов конструкций корпусов судов. Дуговая сварка внутри судового корпуса часто производится в весьма стесненных условиях и представляет собой источник опасности в связи с выделением токсичных паров и газов в ограниченном объеме. Полнейший дискомфорт при попытках сварить что-либо в узком месте.

Рис. 9.1
Робот для дуговой сварки
сваривает днище корпуса
судна



Такие условия создают идеальную область применения роботов для дуговой сварки, один из которых изображен на рис. 9.1.

Важным фактором, определяющим превосходство роботизированной дуговой сварки над ручной, является увеличение «времени горения дуги». При ручной сварке время горения дуги часто составляет лишь 20—30 % всего времени цикла, поскольку оставшиеся 70—80 % уходят на манипулирование защитным шлемом, респиратором и другим защитным оборудованием, чего не требуется при использовании робота. Иногда один опытный сварщик может управлять и обслуживать несколько установок для дуговой сварки, обеспечивая производительность труда в 2—4 раза большую, чем у одного сварщика, работающего вручную. В примере показано, за счет чего возникают такие преимущества.

Пример 9.1. Дуговая сварка мостовых ферм. Изготовление мостовых ферм из стандартного стального проката является основной областью применения дуговой сварки. Условимся, что среднее время горения дуги при ручной сварке $t_1 = 30\%$, а при роботизированной $t_2 = 80\%$. Примем, что робот кладет шов на 40 % быстрее человека ($P_2 = 1,4P_1$), а один сварщик может обслуживать сразу четыре роботизированные установки для сварки ($n = 4$). Сколько сварщиков потребовалось бы, чтобы обеспечить такой же уровень производительности, как у одного сварщика, обслуживающего сварочный робот?

Решение. Производительность роботизированной системы

$$P = n \frac{t_2}{t_1} P_2 = 4 \frac{80}{30} 1,4P_1 = 15P_1.$$

Столь заметное повышение производительности, полученное в примере 9.1, не является преувеличением. В действительности производительность дуговой сварки можно повысить еще больше, если при получении шва применить метод «волнистого шва».

Этот метод заключается в том, что сварщик получает шов повышенного качества, придавая концу электрода осциллирующее движение поперек шва. Другими словами, сварщик делает зигзагообразный шов, наподобие специальной швейной машины, которая шьет зигзагом для подобной же цели.

Для получения волнистого шва от сварщика требуется значительный навык, но и опытный сварщик навряд ли сможет обеспечить равномерность волны при длинном шве. Даже если опыт сварщика позволяет получить качественный шов, скорость процесса значительно сокращается в связи с требованием придавать электроду осциллирующее движение. Сварочный робот по программе может осуществлять осцилляцию сварочной головки во много раз быстрее, чем это способен делать человек.

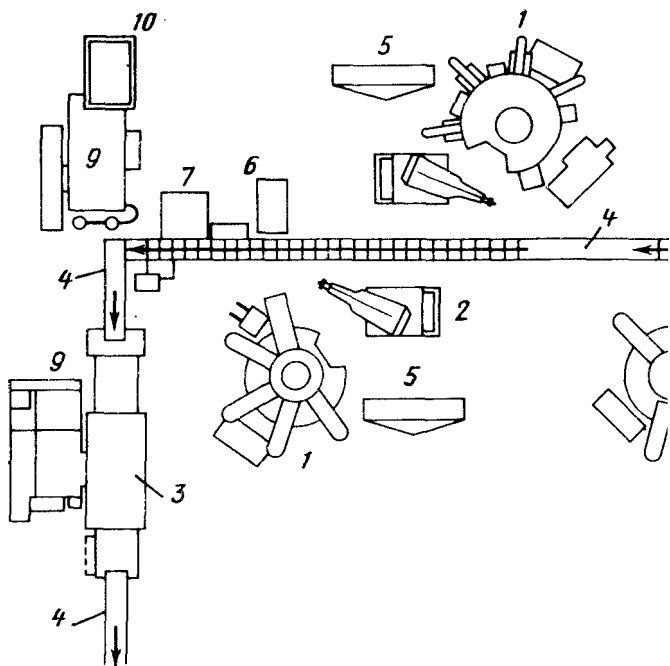
Автор имел возможность наблюдать промышленную дуговую сварку вольфрамовым электродом в среде инертного газа с поперечной осцилляцией электрода, при которой робот обеспечивал производительность, в 8—10 раз большую, чем сварщик, которого он заменил. Кроме того, что робот показал превосходство в скорости, превосходство в качестве сварного шва было совершенно очевидным.

9.2. ЗАГРУЗКА ТЕХНОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ

Загрузка и разгрузка технологического оборудования с помощью промышленных роботов имеет те же самые основные преимущества, что и роботизированная сварка, а именно: обеспечение безопасности и освобождение от необходимости вручную переносить тяжести. Исторически сложилось так, что работа оператора дыропробивного прессы оказалась наиболее опасной работой на заводе из-за риска ампутации при управлении прессом вручную. Этот риск был значительно снижен благодаря возрастающему использованию роботов и устройства для автоматического управления прессами и возрастающему вниманию к стандартам по обеспечению безопасности на производстве, особенно стандартам, относящимся к безопасности работы у прессов и другого оборудования, представляющего опасность, изложенным в законе об охране труда и здоровья на производстве (OSHA).

Оба направления имеют отношение к области роботизации, первое — непосредственное, второе опосредствованное, поскольку упомянутое серьезное отношение к стандартам по технике безопасности способствует движению в направлении к автоматизации загрузки. Первый пример загрузки и разгрузки технологического оборудования роботом приведен на рис. 6.7, где показан робот, загружающий и разгружающий металлорежущий станок. Рассмотрим еще несколько примеров и проанализируем взаимодействие между роботом и оборудованием, которое он обслуживает.

Загрузка нескольких технологических установок несколькими роботами. На рис. 9.2 представлена планировка роботизированного технологического участка, в котором четыре робота загружают и разгружают четыре фрезеро-машинных станка, а конвейер транспортирует изделия между ними. Изделие представляет собой корпус двигателя стеклоочистителя грузовика, выпускаемого в количестве 28 000 шт. в день (двухсменная работа) при длитель-



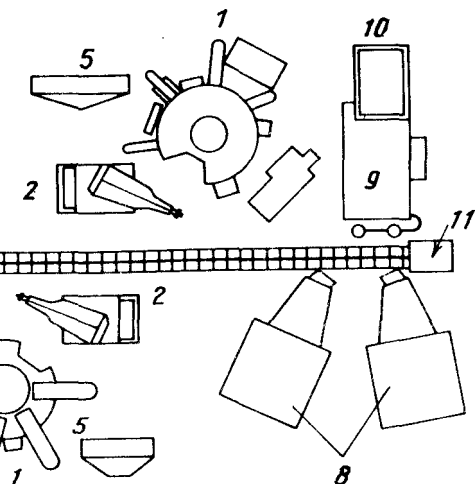


Рис. 9.2

Планировка роботизированного технологического участка для обработки корпусов двигателей стеклоочистителей:

1 — фрезеровальные станки фирмы «Кингсбери»; 2 — робот «Юнимейт»; 3 — модельная машина для отливки; 4 — конвейер; 5 — панель управления; 6 — верстак; 7 — сосуд; 8 — опрокидыватель сосудов; 9 — фильтр; 10 — опрокидыватель; 11 — натяжное устройство

ности цикла 8 с. Фрезеровальные станки карусельного типа сами по себе являются высокоавтоматизированным оборудованием. Линия обладает достаточной гибкостью, чтобы обеспечить партионный выпуск нескольких слегка отличающихся моделей без переоснащения. Основной задачей, которая преследовалась при создании линии, было избежать замедления темпа выпуска, вызываемого застреванием деталей при ручной загрузке станков. Высокий темп работы и малые допуски делали загрузку вручную затруднительной. В то же время робот обеспечил требуемые темп и точность работы, позволяющие избежать застревания деталей. Побочным получаемым преимуществом являлось сокращение брака. Суммарный экономический эффект должен был обеспечить срок окупаемости в один год. Следующий пример иллюстрирует расчет производительности системы, состоящей из нескольких роботов и нескольких технологических установок.

Пример 9.2. Производительность роботизированного участка. Роботизированный участок, планировка которого показана на рис. 9.2, обеспечивает длительность цикла $T_{\text{ц}} = 8$ с и при двухсменной работе ($T = 16$ ч) выпускает свыше 2800 деталей в сутки ($P_1 = 2800$ деталей/сут). Обрабатывают ли фрезеровальные станки фирмы «Кингсберн», установленные на участке, каждую деталь из партии последовательно или каждый из станков выполняет полный цикл обработки, функционируя параллельно, т. е. каждая деталь проходит полную обработку на одном станке, а не на четырех ($n' = 4$)?

Решение. Последовательная обработка. Суточный выпуск при двухсменной работе

$$P_2 = (1/T_{\text{ц}}) \cdot 3600 \cdot 16 = (1/8) \cdot 3600 \cdot 16 = 7200 \text{ деталей/сут.}$$

Параллельная обработка. Суточный выпуск при двухсменной работе $P_3 = (1/T_{\text{ц}}) \cdot 3600 \cdot 16 \cdot n' = 1/8 \cdot 3600 \cdot 16 \cdot 4 = 28\,800$ деталей/сут.

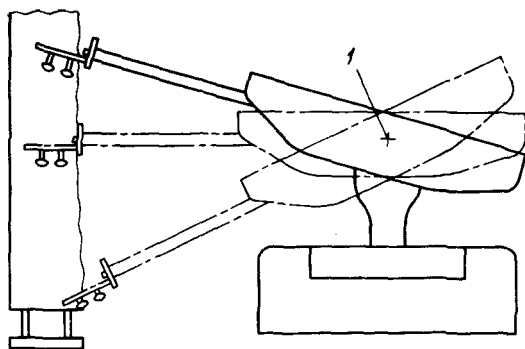
Поскольку выпуск за сутки был установлен «свыше 28 000», можно сделать вывод, что система работает по принципу параллельной обработки, когда каждый фрезеровальный станок полностью обрабатывает деталь, в то время как другие делают то же самое.

В качестве второго примера загрузки технологического оборудования рассмотрим загрузку прессов, являющихся первоочередным объектом роботизации. В [67] Стауффер сообщает о применении робота для загрузки 150-тонного пресса, используемого в производстве заготовок треугольных угловых соединений для печных дверей. Чаще всего заготовки изготавливаются из листового или полосового материала, но заготовки угловых соединений изготавливались из отходов предыдущей операции. В нашем случае требовалось обеспечить такую гибкость комплекса, чтобы пресс можно было перевести на обслуживание вручную или даже переходить от обслуживания роботом к обслуживанию вручную, и наоборот, в зависимости от нумерации модели и обозначения детали. Для описываемой технологической операции была характерна частая смена штампов при низкой серийности выпуска каждой модели и соответствующей детали. Эти особенности определили целесообразность использования робота вместо жесткой автомати-

Рис. 9.3

Позиционирование робота, работающего в полярных (сферических) координатах и оснащенного механическими упорами при захвате деталей из стопы:

1 — ось поворота



зации, которая часто применяется при загрузке прессов в крупносерийном производстве. Оборудование для изготовления заготовок угловых соединений не отличалось миниатюрностью, поэтому для смены штампов во время переналадки при переходе к обработке новой партии был использован автопогрузчик с вилочным захватом. Установка большого робота не позволяла автопогрузчику подъехать к прессу. Эта проблема была решена за счет установки робота на роликах. В то же время базирующая плита обеспечивала точность позиционирования робота при повторных установках, связанных с переходом от партии к партии деталей.

Робот гидравлического типа, примененный в данном случае, был выбран из соображений получения высокой повторяемости при минимальной стоимости. Робот, работавший в полярных (сферических) координатах, имел четыре координаты, программируемые механическим способом. Из изложенного в гл. 6 известно, что этот робот оснащен регулируемыми упорами по каждой оси движения и должен обеспечивать прекрасную повторяемость, но без контроля движения между крайними положениями. Эта особенность перерастает в проблему, когда робот должен забирать заготовки, уложенные в стопу, как это показано на рис. 9.3. По мере уменьшения высоты стопы расстояние от оси поворота (ось 2) немного изменяется, что хорошо видно из несколько утрированного рисунка. Для захвата заготовки роботом это изменение несущественно, но затем, когда робот помещает заготовку под пресс, она оказывается в неправильном положении относительно штампа. Эту проблему можно решить, расположив заготовки по кривой линии в соответствии с дугой, очерчиваемой вытянутой рукой робота. Для этого требуется немного изменить конструкцию магнитного устройства, укладывающего заготовки веером для последующего захвата их роботом. Стоимость переделки устройства легко компенсируется за счет того, что отпадает необходимость приобретать более дорогой робот с более совершенным контролем движения по осям.

В данном случае захват робота был выполнен в виде вакуумных присосок. Основным преимуществом захватов такого типа является некоторая податливость резиновых присосок, позволяющая компенсировать небольшие отклонения от правильного от-

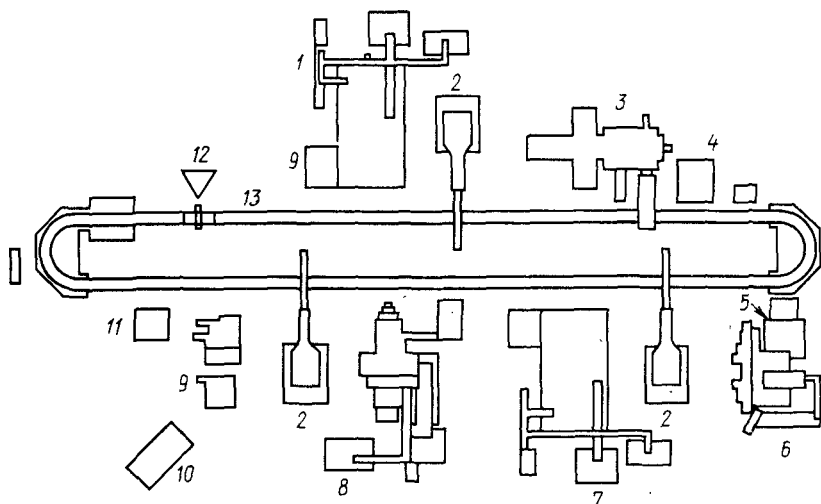


Рис. 9.4

Планировка автоматизированной системы для изготовления смесительных барабанов для множительных машин:

1 — расточной станок; 2 — робот «Юннмейт»; 3 — машина для пайки; 4 — контроль пайки; 5 — шлифовальный станок; 6 — измерительное устройство фирмы «Марпосс»; 7 — станок для обточки торцов; 8 — обточка наружного диаметра; 9 — протяжной станок; 10 — контроль по калибрам; 11 — контроль готовой продукции; 12 — загрузка и разгрузка; 13 — транспортная труба

носительного расположения, когда детали подаются под пресс и позиционируются по базировочному штифту в штампе. Резиновые присоски являются весьма популярными приспособлениями даже при загрузке прессов вручную, но описанные свойства делают их еще более полезными при роботизированной загрузке.

На рис. 9.4 показана планировка производства с комплексным управлением от ЭВМ, состоящего из трех роботов и ряда станков, обслуживаемых единым конвейером. Изготавливаемой продукцией являются медные смесительные барабаны для множительных машин фирмы «Ксерокс». К концам барабанов припаивают крышки из нержавеющей стали, а затем барабаны проходят несколько стадий механообработки. Объем производства линии составляет 100 000 сборок в год.

Успех роботизации изготовления базируется на применении двойных захватов на каждом роботе. Одна половина захвата держит заготовку, в то время как другая половина зажимает отработанную на станке деталь. Затем новая заготовка немедленно устанавливается на станке, что позволяет сэкономить время, обойдясь без лишнего возврата к конвейеру, в течение которого станок простаивает. Сокращение времени простоев во время загрузки и разгрузки повышает производительность каждого станка и тем самым производительность всей линии. В следующих двух примерах даны расчеты для сравнения производительности загрузки с помощью одинарных и двойных захватов.

Пример 9.3. Загрузка и разгрузка фрезерного станка. Робот, связанный с центральным конвейером, загружает и разгружает станок для обточки торцов деталей. Положим, что отдельные операции имеют следующую среднюю продолжительность:

Забрать деталь с конвейера (включая среднее время ожидания подхода детали в позицию для захвата)	2,6 с
Переместить руку робота от конвейера до станка	1,7 с
Загрузить деталь в станок и отвести руку от станка, чтобы станок мог начать обработку	1,1 с
Разгрузить станок	0,8 с
Переместить руку робота от станка до конвейера	1,7 с
Поместить деталь на конвейер	0,3 с

Предположим, что цикл обработки на станке составляет $T_{\text{ц}} = 24$ с. Приняв, что время простоев, связанных с обслуживанием, восстановлением после сбоев и другими причинами, составляет в среднем $T_{\text{р}} = 0,2T_{\text{ц}}$, определить, каков будет выпуск деталей за смену $T = 8$ ч: а) при работе с одинарным захватом, б) при работе с двойным захватом?

Решение. При одинарном захвате предположим, что робот ожидает станок, чтобы разгрузить его, как только цикл обработки закончен. Произвольно приняв начало цикла обработки на станке за начало цикла системы, запишем типичную последовательность операций:

Обработать деталь	24 с
Разгрузить станок	0,8 с
Переместить детали к конвейеру	1,7 с
Поместить обработанную деталь на конвейер	0,3 с
Забрать новую деталь	2,6 с
Переместить деталь к станку	1,7 с
Загрузить станок	1,1 с
Итого $T'_{\text{ц}}$	32,2 с

Выпуск деталей за восьмичасовую смену при эффективности $E = 0,8$ составит $P_1 = (1/T'_{\text{ц}}) \cdot 3600 \cdot TE = 1,8 \cdot 0,8 = 715$ деталей/смена.

При двойном захвате последовательность операций будет следующей:

Обработать деталь	24 с
Разгрузить станок	0,8 с
Загрузить станок	1,1 с
Итого $T'_{\text{ц}}$	25,9 с

Обратите внимание, что при двойном захвате такие операции робота, как перемещение к конвейеру, установка детали, захват новой детали и перемещение к станку, могут быть осуществлены во время цикла обработки и поэтому опущены в приведенной последовательности. Увеличение выпуска можно рассчитать следующим образом:

$$P_2 = (1/T'_{\text{ц}}) \cdot 3600 \cdot TE = (1/25,9) \cdot 3600 \cdot 0,8 = 890 \text{ деталей/смена.}$$

Следовательно, двойной захват позволяет увеличить выпуск деталей за смену на $\Delta P = P_2 - P_1 = 890 - 715 = 175$ деталей, или на $175/715 \approx 24,5\%$ без закупки дополнительного робототехнического или стационарного оборудования.

Пример 9.4. Загрузка и разгрузка роботом с двойным захватом быстродействующего пресса для массового производства. В предыдущем примере длительность цикла обработки ($T_{\text{ц}} = 24$ с) значительно превышает время, занимаемое собственно загрузкой и разгрузкой с помощью робота. Сейчас давайте посмотрим, как изменяется картина, если цикл обработки значительно короче, как например, при штамповке на механическом прессе. Положим, что длительность цикла штамповки составляет $T_{\text{ц}} \approx 0,9$ с, и сравним, как и в предыдущем примере, работу с одинарным и двойным захватом.

Одинарный захват. Соответствующая последовательность операций будет следующей (используя данные по работе робота из примера 9.3):

Длительность цикла штамповки	0,9 с
Разгрузить пресс	0,8 с
Переместить деталь к конвейеру	1,7 с
Поместить обработанную деталь на конвейер	0,3 с
Забрать новую деталь	2,6 с
Переместить деталь к прессу	1,7 с
Загрузить деталь под штамп	1,1 с
Итого $T_{\text{ц}}$	9,1 с

Выпуск деталей за смену $T = 8$ ч при принятой эффективности $E = 0,8$;

$$P_1 = (1/T_{\text{ц}}) \cdot 3600 \cdot TE = (1/9,1) \cdot 3600 \cdot 8 \cdot 0,8 = 2531 \text{ деталей/смена.}$$

Двойной захват. Когда робот с двойным захватом используется для загрузки и разгрузки быстродействующего механического пресса, он не успевает выполнить такую значительную часть своих манипуляций во время цикла обработки, как это наблюдалось при обслуживании станка с более длительным циклом обработки, описанным в примере 9.3.

Следовательно, цикл штамповки будет осуществлен во время манипуляций, производимых роботом, а не наоборот, как это было в предыдущем примере. Записывая последовательность выполнения цикла, мы начнем с операции робота «разгрузить пресс», поскольку операция штамповки происходит во время выполнения других операций и ее длительность не повлияет на длительность всего цикла:

Разгрузить пресс	0,8 с
Загрузить деталь под штамп	1,1 с
Переместить деталь к конвейеру	1,7 с
Поместить обработанную деталь на конвейер	0,3 с
Забрать новую деталь	2,6 с
Переместить деталь к прессу	1,7 с
Итого $T_{\text{ц}}''$	8,2 с

Выпуск деталей за смену

$$P_2 = (1/T_{\text{ц}}'') \cdot 3600 \cdot TE = (1/8,2) \cdot 3600 \cdot 8 \cdot 0,8 = 2809 \text{ деталей/смена.}$$

Увеличение выпуска за смену в связи с применением двойного захвата $\Delta P = 2809 - 2531 = 278$ деталей, или $278/253 \approx 11\%$, что, однако, не так много, как при более долгом цикле обработки, рассмотренном в примере 9.3.

Последовательная загрузка технологического оборудования.

Анализ применения роботов для загрузки технологического оборудования немного усложняется, когда перед одним роботом стоит задача обслуживать несколько единиц оборудования в заданной последовательности. Если инженер по автоматизации тщательно определил длительность операций и спланировал их выполнение, робот может быть запрограммирован на ожидание выполнения цикла соответствующей установкой, причем он будет подходить к ней заблаговременно, сокращая простои в ожидании загрузки и разгрузки.

В действительно имеющем место случае робот, изображенный на рис. 9.5, последовательно загружает и разгружает два сверлильных и расточной станки, обрабатывающие двадцатифунтовые (90 Н) литые детали дифференциала грузовика. На рис. 9.6

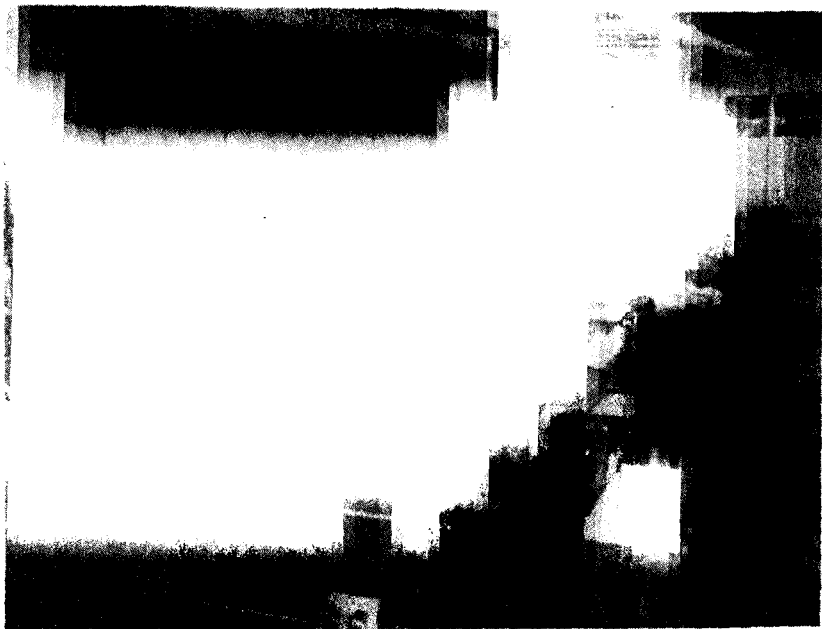


Рис. 9.5

Загрузка и разгрузка роботом сверлильных и расточного станков, предназначенных для обработки литых деталей дифференциала грузовика

показана планировка этого роботизированного комплекса. В позиции *А* робот забирает детали, позиции *Б*, *В* и *Г* представляют собой три станка, а позиция *Д* является конвейером для эвакуации обработанных деталей. Захват робота, так же как и станки, для обеспечения повышенной производительности были спроектированы для обработки одновременно двух деталей.

Очевидно, что в данном случае ставилось целью избавиться от трудоемкой загрузки и разгрузки двадцатифунтовых (90 Н) деталей. До того как комплекс был автоматизирован, оператор перетаскивал около 12 т заготовок в смену. Автоматизация освободила оператора от неприятной обязанности манипулировать тяжелыми заготовками и позволила ему сконцентрировать внимание на наблюдении и контроле качества. Кроме экономии прямой рабочей силы производительность комплекса возросла приблизительно на 60 %.

В предыдущих примерах было показано, что оснащение робота двойным захватом заметно повышает производительность. Это повышение еще более значительно в многостаночных системах, подобных описанной выше, особенно если технологическое оборудование приспособлено для одновременной обработки двух деталей. При работе с одинарным захватом такие станки были бы вынуждены простаивать лишнее время в ожидании, пока робот загружает детали по одной.

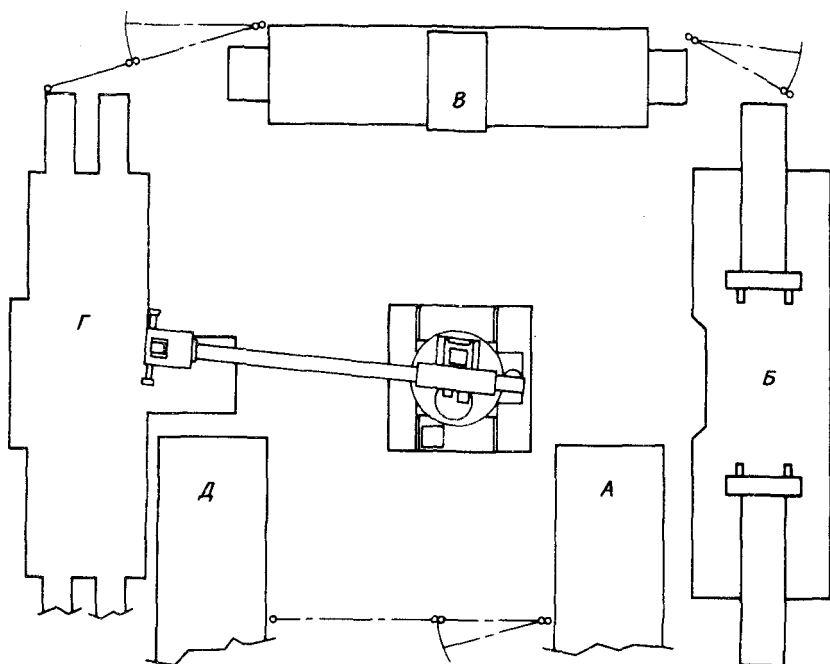


Рис. 9.6

Планировка роботизированного комплекса для обработки литых деталей дифференциала грузовика, обслуживаемого роботом модели А фирмы «Праб» (буквы обозначают направление технологического маршрута):

А — шаговый конвейер для поступающих заготовок; Б — сверление отверстия под штифт (операция 50А); В — растачивание отверстия под шестерню (операция 50В); Г — сверление отверстия в проушине (операция 50С); Д — ленточный конвейер для готовых изделий

Анализ функционирования робота, загружающего и разгружающего в определенной последовательности несколько установок, представляет значительный интерес и включает в себя изучение производительности роботизированных систем.

Инженеры по робототехнике и промышленной автоматизации должны хорошо разбираться в основах такого анализа, поэтому следующие два примера иллюстрируют соответствующие подходы к решению проблемы и одновременно показывают пути сокращения объема вычислений.

Пример 9.5. Анализ производительности при последовательной загрузке технологического оборудования с помощью робота. Предположим, что при обработке детали дифференциала грузовика (см. рис. 9.5) была зафиксирована следующая длительность операций:

Время обработки на первом станке (позиция Б)	9,1 мин
Время обработки на втором станке (позиция В)	9 мин
Время обработки на третьем станке (позиция Г)	5 мин
Время на зажим в захвате (все позиции)	0,1 мин
Время на освобождение из захвата (все позиций)	0,1 мин

9.1. Время (мин), затрачиваемое на перемещение захвата

Из позиции	В позицию				
	А	Б	В	Г	Д
А	—	0,3	0,6	0,9	1,2
Б	0,3	—	0,3	0,6	0,9
В	0,6	0,3	—	0,3	0,6
Г	0,9	0,6	0,3	—	0,3
Д	1,2	0,9	0,6	0,3	—

Время, затрачиваемое на перемещение захвата робота из позиций в позицию, указано в табл. 9.1. Необходимо определить длительность цикла системы и ее производительность.

Решение. Анализ работы и определение производительности системы начнем с составления табл. 9.2, в которую занесем все составляющие цикла. Производительность системы может быть определена, когда цикл начнет повторяться.

Помните, что станок нельзя разгружать до тех пор, пока он не закончил обработку детали. Строки в таблице, отмеченные знаком *, соответствуют моментам в цикле, когда позиция Г разгружена, произвольно принятым за моменты начала и конца цикла. После того как система полностью загружена, разность между показаниями отсчета для операций «разгрузить Г» соответствует длительности цикла. Например:

Показание отсчета времени, когда робот разгружает станок Г, освобождая его от детали 2	35,4
Показание отсчета времени, когда робот разгружает станок Г, освобождая его от детали 1	—24,7
Длительность цикла	<u>10,7 мин</u>

Обратите внимание, что длительность цикла можно определить, зная любые два показания отсчета времени идентичных операций, после того как система вышла на нормальный режим работы. Например, можно выбрать операцию «Переместиться к Г».

Показание отсчета времени для операции «Переместиться к Г», когда станки обрабатывают детали 4, 3 и 2	33,2
Показание отсчета времени для операции «Переместиться к Г», когда станки обрабатывают детали 3, 2 и 1	—22,5
Длительность цикла $T_{\text{ц}}$	<u>10,7 мин</u>

Производительность характеризуется величиной, обратной длительности цикла, поэтому

$$P = \frac{60/T_{\text{ц}} \cdot 60}{10,7} = 5,6 \text{ деталей/ч.}$$

Сейчас давайте рассмотрим аналогичный пример, где оснастка и робот остались теми же, а время обработки на станках сократилось.

**9.2. Анализ цикла: последовательная загрузка
технологического оборудования с помощью робота**

Состояние перед началом операции					Содержание операции	Длительность, мин	Запланированное окончание
Отсчет времени	Робот в позиции	Обслуживаемая позиция					
		Б	В	Г			
0	А				Забрать из А	0,1	0,1
0,1	А				Переместить к Б	0,3	0,4
0,4	Б				Загрузить Б	0,1	0,5
Деталь							
0,5	Б	1			Обработка на позиции Б	9,1	9,6
9,6	Б	1			Разгрузить Б	0,1	9,7
9,7	Б				Переместиться к В	0,3	10,0
10,0	В				Загрузить В	0,1	10,1
10,1	В		1		Обработка на позиции В	9,0	19,1
10,1	В		1		Переместиться к А	0,6	10,7
10,7	А		1		Забрать из А	0,1	10,8
10,8	А		1		Переместиться к Б	0,3	11,1
11,1	Б		1		Загрузить Б	0,1	11,2
11,2	Б	2	1		Обработка на позиции Б	9,1	20,3
11,2	Б	2	1		Переместиться в В	0,3	11,5
19,1	В	2	1		Разгрузить В	0,1	19,2
19,2	В	2			Переместиться к Г	0,3	19,5
19,5	Г	2			Загрузить Г	0,1	19,6
19,6	Г	3		1	Обработка на позиции Г	5,0	24,6
19,6	Г	2		1	Переместиться к Б	0,6	20,2
20,3	Б	2		1	Разгрузить Б	0,1	20,4
20,4	Б			1	Переместиться к В	0,3	20,7
20,7	В			1	Загрузить В	0,1	20,8
20,8	В		2	1	Обработка на позиции В	9,0	29,8
20,8	В		2	1	Переместиться к А	0,6	21,4
21,4	А		2	1	Забрать из А	0,1	21,5
21,5	А		2	1	Переместиться из Б	0,3	21,8
21,8	Б		2	1	Загрузить Б	0,1	21,9
21,9	Б	3	2	1	Обработка на позиции Б	9,1	31,0
21,9	Б	3	2	1	Переместиться к Г	0,6	22,5
24,6	Г	3	2	1	Разгрузить Г	0,1	24,7*
24,7	Г	3	2		Переместиться к Д	0,3	25,0
25,0	Д	3	2		Переместить на позиции Д	0,1	25,1
25,1	Д	3	2		Переместиться к В	0,6	25,7
29,8	В	3	2		Разгрузить В	0,1	29,9
29,9	В	3			Переместиться к Г	0,3	30,2
30,2	Г	3			Загрузить Г	0,1	30,3
30,3	Г	3		2	Обработка на позиции Г	5,0	35,3

Состояние перед началом операции					Содержание операции	Длительность, мин	Запланированное окончание
Отсчет времени	Робот в позиции	Обслуживаемая позиция					
		Б	В	Г			
30,3	Г	3		2	Переместиться к В	0,6	30,9
31,0	Б	3		2	Разгрузить Б	0,1	31,1
31,1	Б			2	Переместиться к В	0,3	31,4
31,4	В			2	Загрузить В	0,1	31,5
31,5	В		3	2	Обработка на позиции В	9,0	40,5
31,5	В		3	2	Переместиться к А	0,6	32,1
32,1	А		3	2	Забрать из А	0,1	32,2
32,2	А		3	2	Переместиться к Б	0,3	32,5
32,5	Б		3	2	Загрузить Б	0,1	32,6
32,6	Б	4	3	2	Обработка на позиции Б	9,1	41,7
32,6	Б	4	3	2	Переместиться к Г	0,6	33,2
35,3	Б	4	3	2	Разгрузить Г	0,1	35,4*

Пример 9.6. Анализ производительности при последовательной загрузке технологического оборудования с помощью робота (при сокращенной длительности обработки). В этом примере мы примем следующую длительность обработки:

Время обработки на первом станке (позиция Б)	2,0 мин
Время обработки на втором станке (позиция В)	1,0 мин
Время обработки на третьем станке (позиция Г)	2,6 мин

Определите длительность цикла системы и ее производительность.

Решение. Цикл детально проанализирован в табл. 9.3.

Показание отсчета времени, когда робот разгружает станок Г, освобождая его от детали 2	12,8
Показание отсчета времени, когда робот разгружает станок Г, освобождая его от детали 1	—8,4
Длительность цикла	<u>4,4 мин</u>

Производительность

$$P = \frac{60}{4,4} = 13,6 \text{ деталей/ч.}$$

Длительность обработки на станках в примерах 9.5 и 9.6 была специально тщательно подобрана, чтобы проиллюстрировать задачу.

В примере 9.5 длительность обработки повлияла на цикл системы, потому что роботу приходилось ожидать один из станков во время выполнения цикла. В примере 9.6 циклы обработки на каждом из трех станков были слишком малы, чтобы роботу приходилось ожидать. Поэтому в примере 9.6 длительность цикла системы сократилась до времени, которое требовалось роботу

**9.3. Анализ цикла последовательной загрузки
технологического оборудования с помощью робота
при измененной длительности обработки**

Состояние перед началом операции					Содержание операции	Длительность, мин	Запланированное окончание
Отсчет времени	Робот в позиции	Обслуживаемая позиция					
		Б	В	Г			
0	А				Забрать из А	0,1	0,1
0,1	А				Переместиться к Б	0,3	0,4
0,4	Б				Загрузить Б	0,1	0,5
Деталь							
0,5	Б	1			Обработка на позиции Б	2,0	2,5
2,5	Б	1			Разгрузить Б	0,1	2,6
2,6	Б				Переместиться к В	0,3	2,9
2,9	В				Загрузить В	0,1	3,0
3,0	В		1		Обработка на позиции В	1,0	4,0
3,0	В		1		Переместиться к А	0,6	3,6
3,6	А		1		Забрать из А	0,1	3,7
3,7	А		1		Переместиться к Б	0,3	4,0
4,0	Б		1		Загрузить Б	0,1	4,1
4,1	Б	2	1		Обработка на позиции Б	2,0	6,1
4,1	Б	2	1		Переместиться к В	0,3	4,4
4,4	В	2	1		Разгрузить В	0,1	4,5
4,5	В	2			Переместиться к Г	0,3	4,8
4,8	Г	2			Загрузка Г	0,1	4,9
4,9	Г	2		1	Обработка на позиции Г	2,6	7,5
4,9	Г	2		1	Переместиться к Б	0,6	5,5
6,1	Б	2		1	Разгрузить Б	0,1	6,2
6,2	Б				Переместиться к В	0,3	6,5
6,5	В				Загрузить В	0,1	6,6
6,6	В		2	1	Обработка на позиции В	1,0	7,6
6,6	В		2	1	Переместиться к А	0,6	7,2
7,2	А		2	1	Забрать из А	0,1	7,3
7,3	А		2	1	Переместиться к Б	0,3	7,6
7,6	Б		2	1	Загрузить Б	0,1	7,7
7,7	Б	3	2	1	Обработка на позиции Б	2,0	9,7
7,7	В	3	2	1	Переместиться к Г	0,6	8,3
8,3	Г	3	2	1	Разгрузить Г	0,1	8,4
8,4	Г	3	2		Переместиться к А	0,3	8,7
8,7	Д	3	2		Поместить в Д	0,1	8,8
8,8	Д	3	2		Переместиться к В	0,6	9,4
9,4	В	3	2		Разгрузить В	0,1	9,5
9,5	В	3			Переместиться к Г	0,3	9,8
9,8	Г	3			Загрузить Г	0,1	9,9
9,9	Г	3		2	Обработка на позиции Г	2,6	12,5

Состояние перед началом операции					Содержание операции	Длительность, мин	Запланированное окончание
Отсчет времени	Работ в позиции	Обслуживаемая позиция					
		Б	В	Г			
9,9	Г	3		2	Переместить к Б	0,6	10,5
10,5	Б	3		2	Разгрузить Б	0,1	10,6
10,6	Б			2	Переместиться к В	0,3	10,9
10,9	В			2	Загрузить В	0,1	11,0
11,0	В		3	2	Обработка на позиции В	1,0	12,0
11,0	В		3	2	Переместиться к А	0,6	11,6
11,6	А		3	2	Забрать из А	0,1	11,7
11,7	А		3	2	Переместиться к В	0,3	12,0
12,0	Б		3	2	Загрузить Б	0,1	12,1
12,1	Б	4	3	2	Обработка на позиции Б	2,0	14,1
12,1	Б	4	3	2	Переместиться к Г	0,6	12,7
12,7	Б	4	3	2	Разгрузить Г	0,1	12,8

для выполнения всех составляющих цикла без учета механообработки. Сказанное можно подтвердить для примера 9,6, суммировав время работы робота по всем составляющим цикла, как это сделано в табл. 9.4.

Обратите внимание, что расчеты по табл. 9.4 приводят к тому же результату, что получен в примере 9.6. Это может значительно облегчить расчет длительности цикла и производительности для операций загрузки с помощью роботов, потому что табл. 9.4 очевидно проще, чем табл. 9.3. В то же время табл. 9.4 не объясняет: при каком максимальном времени механообработки можно проводить вычисление, принимая во внимание только время работы робота. Это время можно рассчитать, как показано ниже.

Весь цикл работы робота, приведенный в табл. 9.3, можно разделить на две основные составляющие:

Разгрузить данный станок

$$\left. \begin{array}{c} \bullet \\ \bullet \quad \bullet \\ \bullet \end{array} \right\}$$
составляющая α

Снова загрузить данный станок

Все другие операции робота } Составляющая β

В табл. 9.5 произвольно выбран станок на позиции Γ в качестве базы для подсчета составляющих цикла α и β , однако вне зависимости от того, какой станок выбран, α и β останутся

9.4. Подсчет времени работы робота при загрузке оборудования

Содержание операции	Длитель- ность операции, мин
Разгрузить Г	0,1
Переместиться к Д	0,3
Поместить в Д	0,1
Переместиться к В	0,6
Разгрузить В	0,1
Переместиться к Г	0,3
Загрузить Г	0,1
Переместиться к Б	0,6
Разгрузить Б	0,1
Переместиться к В	0,3
Загрузить В	0,1
Переместиться к А	0,6
Забрать из А	0,1
Переместиться к Б	0,3
Загрузить Б	0,1
Переместиться к Г	0,6
Итого	4,4 мин

9.5. Расчет составляющих цикла α и β

Содержание операции		Длитель- ность опе- рации, мин
Состав- ляющая $\alpha =$ $= 1,6 \text{ мин}$	Разгрузить Г	0,1
	Переместиться к Д	0,3
	Поместить в Д	0,1
	Переместиться к В	0,6
	Разгрузить В	0,1
	Переместиться к Г	0,3
Состав- ляющая $\beta =$ $= 2,8 \text{ мин}$	Загрузить Г	0,1
	Переместиться к Б	0,6
	Разгрузить Б	0,1
	Переместиться к В	0,3
	Загрузить В	0,1
	Переместиться к А	0,6
	Забрать из А	0,1
	Переместиться к Б	0,3
	Загрузить Б	0,1
Переместиться к Г		0,6
Итого		4,4 мин

для данного примера постоянными. Читатель может удостовериться в этом, выбрав станок на позиции В или Б и получив те же значения α и β , что и для позиции Г.

В том случае, если время работы робота в цикле не меняется от станка к станку, выделение составляющих α и β и их подсчет значительно облегчат расчет длительности всего цикла и производительности при последовательной загрузке технологического оборудования с помощью робота. Расчет можно выполнить следующим образом:

1. Подсчитать составляющие цикла α и β .

2. Определить наибольшее время обработки M среди всех станков.

3. а) Если $M \leq \beta$, длительность цикла $T = \alpha + \beta$.

б) Если $M > \beta$, длительность цикла $T = \alpha + M$. Проверим сказанное для примеров 9.5 и 9.6:

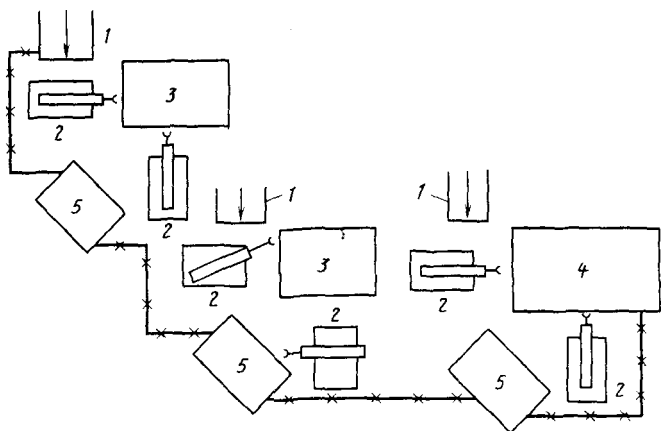


Рис. 9.7

Три роботизированных комплекса, обслуживаемых шестью промышленными роботами:

1 — индукционная нагревательная печь; 2 — робот; 3 — ковочный пресс усилием 25 МН; 4 — ковочный пресс усилием 50 МН; 5 — пресс для обрубki заусенцев

Для примера 9.5: $\alpha = 1,6$ мин, $\beta = 2,8$ мин, $M = 9,1$ мин. Поскольку $M > \beta$, $T = \alpha + \beta = 1,6 + 2,8 = 4,4$ мин/цикл.

Для примера 9.6: $\alpha = 1,6$ мин, $\beta = 2,8$ мин, $M = 2,6$ мин. Поскольку $M \leq \beta$, $T = \alpha + \beta = 1,6 + 2,8 = 4,4$ мин/цикл.

Оба результата точно соответствуют полученному ранее при решении примеров 9.5 и 9.6, при этом удалось обойтись без занимающего много времени анализа, представленного в табл. 9.2 и 9.3.

Штамповка и литье под давлением. Когда детали, которыми необходимо манипулировать, находятся в разогретом состоянии, это в еще большей степени способствует использованию роботов при загрузке и разгрузке. Поэтому роботы находят применение в операциях штамповки, литья под давлением и других, связанных с манипулированием горячими деталями. Всякий, кто присутствовал при объемной штамповке на падающем молоте и ощутил жару и шум, сопровождающие эту операцию, поймет, что она является первейшим кандидатом на применение роботов.

На рис. 9.7 показана схема загрузки и разгрузки штамповочных прессов роботами по два в каждом из трех связанных роботизированных комплексов. Первый робот в каждом комплексе забирает разогретую заготовку из индукционной печи и помещает ее под пресс. Второй робот разгружает пресс и передает заготовку на вспомогательную операцию обрубki заусенцев. Все операции на трех комплексах выполняются внутри огражденной зоны.

Хотя руки роботов более прочны и более термостойки, чем руки людей, они также могут перегреться. В то же время можно составить программу таким образом, чтобы робот через определенные интервалы времени погружал руку в бак с охлаждающей

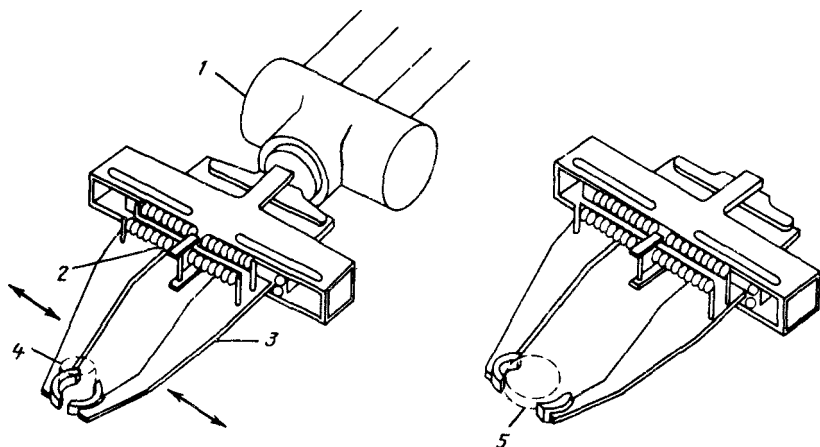


Рис. 9.8

Специальный захват робота для поковок:

1 — зубчатая передача привода руки; 2 — пружины растяжения; 3 — пальцы захвата, имеющие плоскопараллельное перемещение; 4 — стальная заготовка; 5 — расплюснутая заготовка

средой. Такой прием был рассмотрен в качестве упражнения по программированию роботов в гл. 8.

Другой проблемой, возникающей при использовании роботов в штамповке, является сопутствующее этой операции изменение формы заготовки. Это может привести к необходимости разработать специальный захват для обеих форм сразу. На рис. 9.8 показан захват робота, оснащенный пружинами растяжения, позволяющими захватить заготовку с небольшим поперечным сечением, а затем, после штамповки, расплюснутую заготовку с большим поперечным сечением. Удлиненная плоская форма пальцев захвата обеспечивает передачу необходимого усилия зажима без передачи излишнего количества теплоты от горячей заготовки к деталям захвата и руки робота.

9.3. ОКРАСКА РАСПЫЛЕНИЕМ

Задача оборудовать кабину для окраски распылением с соблюдением соответствующих стандартов по технике безопасности и охране здоровья делает окраску распылением естественной областью для применения промышленных роботов. Кроме того, робот может обеспечить такую стабильность процесса, которую трудно воспроизвести и опытному маляру. Хотя опытный маляр нужен для того, чтобы вначале обучить робот выполнению операции окраски, обученный робот воспроизведет операцию тысячу раз со стабильностью, не доступной для учителя.

В крупномасштабной линии для окраски распылением кузовов автомобилей роботы не только выполняют саму операцию ок-

раски, но даже открывают двери кузова, чтобы ускорить процесс. Линии для окраски представляют собой такое же драматическое зрелище, как и линии для точечной сварки; в захватывающем фильме «Балет роботов» * движения роботов положены на музыку. Роботы для окраски распылением, в несколько более скромном масштабе, находят применение не только в автомобилестроении, но и в других отраслях промышленности.

В гл. 4 рассмотрено влияние, которое оказывают отказы или низкая надежность функционирования отдельных устройств на производительность производственной линии с последовательной обработкой. Особенно тяжелые проявления наблюдаются у линий окраски распылением. Отказ оборудования на такой линии может вылиться в очень дорогостоящую остановку всей линии, которая вероятно приведет к появлению целого ряда дефектов на изделиях, связанных с перерывом в постоянном процессе окраски. Устранение неисправностей и новая наладка оборудования также могут потребовать много времени, значительно больше, чем при простом освобождении сборочных линий от закладных деталей, рассмотренном в гл. 4.

Самой практичной стратегией в случае возникновения отказа в линии окраски распылением обычно будет продолжение операции, если это возможно, и возникшие дефекты устранить уже в зоне ручного исправления, следящей за автоматической линией. К сожалению, некоторые неисправности роботизированного краскопульта в зоне автоматического распыления могут лишить линию работоспособности. Примерами таких неисправностей могут послужить отказы, связанные со столкновениями оборудования с обрабатываемой продукцией, и отказы, приводящие к отключению распылителя в критический момент операции. Влияние таких отказов рассмотрено среди других факторов в примерах, иллюстрирующих экономические вопросы организации окраски на роботизированных линиях.

Пример 9.7. Экономика линии по окраске распылением. В ожидании резкого роста спроса на выпускаемую продукцию фирма предполагает вложить $K_1 = 1,7$ млн. долл. в систему из 11 роботизированных краскопультов (включая один резервный), связанных соответствующим конвейером, призванную заменить линию по окраске вручную, где было занято 14 опытных маляров ($n_2 = 14$).

Существующая линия по окраске вручную позволяет окрашивать 90 изделий в час при действительном производственном времени $T_1 = 6$ ч в сутки из смены $T_2 = 8$ ч, куда включены перерывы, время на обслуживание, ремонт, уборку и переодевание в спецодежду. Затраты на каждого маляра, включая зарплату, пособия и накладные расходы, составляют $C_1 = 16$ долл./ч. Дополнительные расходы на ремонт системы для окраски вручную составляют около $C_2 = 400$ долл./м.

В новой линии стоимостью 1,7 млн. долл. одновременно работают десять роботизированных краскопультов ($n_1 = 10$), которые позволяют достигнуть производительности $P_1 = 150$ изделий/ч при полном использовании смены 8 ч в сутки. Амортизация капиталовложений с учетом прибыли, погашения кредитов, отчислений на реновацию и страхование для новой (проектируемой) и заме-

* «Балет роботов» можно приобрести в виде 16-мм фильма или на видеокассете у фирмы «Pyramid Film and Video» P. O. Box 1048, Santa Monica, CA 90406.

няемой систем эквивалентна сметным годовым расходам, составляющим $K_2 = 800\,000$ долл. В эту сумму не включены стоимость обслуживания, ремонта и оплата двух опытных операторов, наблюдающих за автоматической системой. Предполагается, что стоимость обслуживания, ремонта и наблюдения за системой составит около $C_3 = 75$ долл/ч. Сравните себестоимость изделия при существующей системе окраски вручную и при роботизированной системе, которую предполагается установить.

Решение. *Обработка вручную.*

$$C' = C'_1 + C'_2 = n_2 C_1 T_2 / T_1 + C_2 n_2 / (n_3 n_4) = 14 \cdot 16 \cdot 8 / 6 + (400 \cdot 12) / (50 \cdot 5) = 299 + 3,2 = 302 \text{ долл/ч,}$$

где $n_2 = 12$ мес (в году); $n_3 = 50$ недель (в году); $n_4 = 5$ дней (в неделю).

Производительность $P_1 = 90$ изделий/ч.

Себестоимость $C'_1 = C' / P_1 = 302 / 90 = 3,36$ долл/изделие.

Роботизированная система.

Себестоимость производства

$$C'' = K_2 / (T_2 n_3 n_4) + C_3 = 800\,000 / (8 \cdot 50 \cdot 5) + 75 = 475 \text{ долл/ч.}$$

Производительность $P_2 = 150$ изделий/ч.

Себестоимость изделий

$$C''_1 = C'' / P_2 = 475 / 150 = 3,17 \text{ долл/изделие.}$$

Проведенный анализ показывает, что внедрение новой роботизированной системы имеет смысл, но полученные результаты близки, даже слишком близки, чтобы оправдать риск капиталовложений для некоторых фирм, хотя доход при определенной норме прибыли уже был заложен в сумму 800 тыс. долл. в год, предусмотренную на амортизацию капитальных вложений в роботизированную систему.

К расстройству проникнутого энтузиазмом инженера по роботизации и автоматизации, предлагающего новую роботизированную систему для окраски распылением, рассмотренную в примере 9.7, может появиться скептически настроенный руководитель, который осознал опасность выхода из строя одного из десяти роботов системы окраски, возможно сопровождающегося ее остановкой, повреждением изделия, а также нанесением ущерба производству. При анализе примера 9.8 эти соображения приняты во внимание.

Пример 9.8. Анализ работы линии для окраски распылением с учетом возможных отказов робота. Предположим, что в роботизированной системе для окраски распылением, рассмотренной в примере 9.7, у каждого из десяти роботов могут наблюдаться отказы, приводящие к остановке всей линии. Среднее время между отказами каждого из роботов для отказов указанного типа, по сообщениям изготовителей роботов, составляет приблизительно $t_1 = 2000$ ч или один год работы при обычной односменной работе. В случае возникновения такого отказа потеря производственного времени составит около $t_2 = 4$ ч, а дополнительный расход материала и убытки от повреждений оцениваются в сумму до $C_4 = 5000$ долл. на отказ. Рассчитать себестоимость обработки изделия, исходя из неизбежности появления случайных отказов, выводящих систему из строя.

Решение. Поскольку для всех десяти роботов было установлено одно и то же среднее время между отказами 2000 ч, для системы оно будет составлять $t_3 = 2000 \text{ ч} / 10 = 200 \text{ ч}$. Если десять отдельных роботов имеют разные значения среднего времени между отказами, вычисление этого показателя для системы становится более сложным [35]. Следует иметь в виду издержки по двум статьям: 1) потерям производственного времени, 2) расходам на дополнительные мате-

риалы и устранение результатов повреждения. Потери производственного времени, оцениваемые в 4 ч на один отказ, соответствуют значению коэффициента готовности системы, получаемому в результате следующих вычислений:

$$K_r = \frac{t_3}{t_3 + t_2} = \frac{200}{200 + 4} = 0,98, \text{ или } 98 \ \%.$$

Отсюда из времени $T = 2000$ ч, составляющего стандартный годовое фонд времени, только $T_{\Pi} = K_r t_1 = 0,98 \cdot 2000 = 1960$ ч следует считать временем производства для роботизированной системы окраски. Это увеличит себестоимость 1 ч работы, потому что эквивалентные сменные годовые расходы $K_2 = 800\,000$ долл. должны быть распределены между 1960 ч времени производства вместо 2000 ч.

Другим элементом затрат являются расходы $C_4 = 5000$ долл., идущие на дополнительные материалы и устранение результатов отказа. Поскольку среднее время между отказами для системы составляет $t'_1 = 200$ ч, система будет выходить из строя с интенсивностью I , несколько меньшей, чем 10 раз в год:

$$I = T_{\Pi}/t'_1 = 1960/200 = 9,8 \text{ отказов/год.}$$

Простейшим способом оценки расходов на дополнительные материалы и устранения результатов повреждения является пересчет их на производственное время следующим образом:

Расходы (на материалы и устранение результатов повреждения)

$$C'_4 = C_4/t'_1 = 5000/200 = 25 \text{ долл./ч.}$$

Подведем итоги вычислений для роботизированной системы. Себестоимость производства

$$C'' = K_2/T_{\Pi} + C_3 + C'_4 = 800\,000/1960 + 75 + 25 = 508 \text{ долл./ч.}$$

Производительность $P_2 = 150$ изделий/ч.

Себестоимость изделия

$$C'_1 = C''/P_2 = 508/150 = 3,39 \text{ долл./изделие.}$$

Пример 9.8 показывает, что учет возможности отказов роботов в системе для окраски, включающей десять роботов, склоняет результат экономического анализа в пользу окраски вручную по категории себестоимости обработки изделия. Этот вывод, однако, нельзя считать окончательным.

Начать с того, что существует критерий объема выпуска. В начале примера 9.7 было указано, что фирма ожидает резкого роста спроса на выпускаемую продукцию. Увеличение производительности за счет внедрения роботизированной системы для окраски распылением должно быть именно тем, что нужно, чтобы обеспечить соответствующий рост выпуска без значительного расширения производственных площадей, которое потребовалось бы для увеличения количества обычных линий. Сравним объем выпуска двух систем.

Система для окраски вручную. Годовой выпуск

$$n' = P_1 T_1 n_1 n_3 = 90 \cdot 6 \cdot 5 \cdot 50 = 135\,000 \text{ изделий /год.}$$

Роботизированная система для окраски. Годовой выпуск

$$n'' = \Delta_2 T_{\Pi} = 150 \cdot 1960 = 294\,000 \text{ изделий/год.}$$

Следовательно, роботизированная система обеспечивает увеличение годового выпуска более чем вдвое по сравнению с системой для окраски вручную.

Еще одним моментом, который следует учитывать при сравнении предполагаемой роботизированной системы с системой для окраски вручную, является возможность двух- или трехсменной работы.

С увеличением количества рабочих часов в сутки шансы на выбор роботизированной системы растут, потому что она является капиталоемкой, в то время как система для окраски вручную — трудоемкой. Рассмотрим влияние этих факторов на примере 9.9.

Пример 9.9. Линия для окраски распылением при двухсменной работе. Обратимся к примерам 9.7 и 9.8 и пересчитаем себестоимость производства изделия для предполагаемой роботизированной системы и системы для окраски распылением вручную при двухсменной работе обеих систем. Примем увеличение суммарных затрат на содержание обслуживающего персонала при работе во вторую смену $C'' = 1,1C'_1$, где C' — затраты во вторую смену.

Решение. Система для окраски вручную. Средняя себестоимость производства

$$C'_{\text{ср}} = 0,5 (C'_1 + C'') = 0,5 (299 + 299 \cdot 1,4) + 3,2 = 317 \text{ долл./ч.}$$

Производительность $P_1 = 20$ изделий/ч.

Себестоимость изделия

$$C'_1 = C'_{\text{ср}} / P_1 = 317 / 20 = 15,85 \text{ долл./изделие.}$$

Роботизированная система. Себестоимость производства

$$C''_{\text{ср}} = K_2 / (T_n \cdot 2) + 0,5 (C_3 + 1,1C_3) + C'_4 = 800\,000 / (1960 \cdot 2) + 0,5 (75 + 1,1 \cdot 75) + 25 = 308 \text{ долл./ч.}$$

Производительность $P_2 = 150$ изделий/ч.

Себестоимость изделия

$$C''_1 = C''_{\text{ср}} / P_2 = 308 / 150 = 2,05 \text{ долл./изделие.}$$

Таким образом, при работе в две смены роботизированная система выступает как значительно лучшая альтернатива. Это преимущество реализуется, только если фирме действительно необходима дополнительная продукция, получаемая при работе во вторую смену.

Из проделанного экономического анализа операции окраски распылением можно сделать вывод, приложимый к проектам автоматизации вообще. Решение автоматизировать ручную операцию является решением сделать капитальные вложения в оборудование, причем капитальные вложения, которые в общем будут давать большой доход, если оборудование используется больше часов в сутки. Поэтому, если роботы или другое оборудование для автоматизации кажется экономически непривлекательным при работе в одну смену, оно будет более привлекательным при двухсменной работе. Три смены, вероятно, дадут лучшие результаты, чем две, а три смены семь дней в неделю (что эквивалентно приблизительно четырем сменам), наверное, окажутся наилучшей альтернативой при условии, что существует потребность в объеме выпуска продукции, обеспечиваемом при таких условиях.

9.4. СБОРКА

Наиболее сложной областью применения роботов является сборка. Сборка требует точности, повторяемости, разнообразия движений, применения изощренных захватных приспособлений, а иногда комбинированных механизмов, в которых захват, кроме простого удержания деталей, играет активную и решающую роль в одной из операций сборки.

В основании решения применить промышленные роботы в автоматизированной сборке обычно лежит стремление к экономии заработной платы. Это предъявляет особое требование к скорости и производительности роботизированной операции. Можно допустить медлительность и непроизводительность роботизированной операции литья под давлением или дуговой сварки, если роботизация дает возможность избежать опасности или повышает качество сварки, однако роботизированная сборка не позволяет такую роскошь. В большинстве случаев сборочные операции просто не являются опасными, грязными, сопряженными с работой в жарких или неблагоприятных для здоровья условиях. Следовательно, сборочный робот должен быть быстрым и производительным, чтобы конкурировать с ручной сборкой без использования его особых способностей функционировать в неблагоприятных условиях.

В тех случаях, когда быстрота и эффективность не могут служить критерием использования сборочного робота, побудительной причиной может оказаться точность. Некоторые сборочные операции, такие, как сборка электронных печатных плат, очень чувствительны к ошибкам, совершаемым людьми. Промышленные роботы не склонны к совершению ошибок такого типа. Кроме того, появление некоторых дефектных или выходящих за предел допусков сборок, может быть связано с преднамеренными упущениями со стороны оператора — человека, который стремится увеличить выпуск и заработать премиальные. Некоторые дефекты сборки, такие, как отсутствие винтов или недостаточная их затяжка, могут остаться незамеченными при контроле сборки или окончательном контроле. В этих случаях неэтичные действия операторов могут случайно пройти безнаказанно и отрицательно сказаться на качестве изделий, в то время как робот запрограммирован осуществлять операцию каждый раз одинаково.

Сборка двигателя. Автомобильная промышленность является основным потребителем роботов для точечной сварки и окраски распылением. В меньшей степени сборочные роботы применяются в производстве автомобилей. На рис. 9.9 показана входящая в состав сборочной линии опытная роботизированная установка, разработанная в Фраунгоферовом институте автоматизации промышленности, ФРГ. Основная сборка блока цилиндров двигателя перемещается на палетном транспортном механизме. Роботы имеют свободный доступ к подсборкам, доставляемым не-

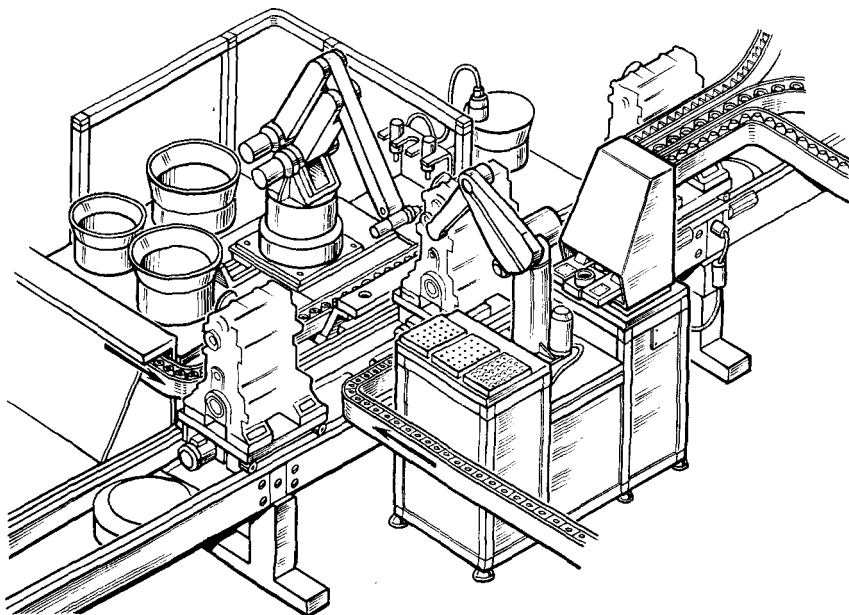


Рис. 9.9

Опытная роботизированная установка для сборки автомобильных двигателей

большими вспомогательными конвейерами. Вибробункеры подают и ориентируют мелкие детали, такие, как винты и болты. В наиболее сложных роботизированных установках, таких, как изображена на рис. 9.9, роботу необходимо автоматически менять различные захваты, каждый из которых предназначен для выполнения самостоятельной задачи. Это может быть выполнено за счет программирования гибко организованной памяти робота при наличии различных захватов, установленных в стеллажах или в магазине. Примером сменных устройств могут служить автоматические винтоверты, которых на одной установке может быть несколько типов. Робот, таким образом, обеспечивает автоматическую смену положения и ориентации автоматическим винтовертом, описанным в гл. 4.

Сборка механизмов электрических и электронных приборов. Индексируемые транспортные механизмы с палетами в роботизированной сборке применяются не только для тяжелых узлов типа блока цилиндров автомобильного двигателя. Малогабаритный лентопротяжный механизм кассетного магнитофона транспортируется аналогичным транспортом на высокоавтоматизированной сборочной линии в Кофу, где расположен завод японской фирмы «Санкио Сейки Манюфакчуринг». Линия из 12 роботизированных установок только в двух местах прерывается операциями, выполняемыми вручную: между роботами 9 и 10, а также между робо-

тами 11 и 12. На большинстве установок можно увидеть сложные движения и сменные захваты. Автоматизированная линия является восхитительным зрелищем, которое становится еще более привлекательным от большого разнообразия хитроумных сборочных операций, требующих сложных движений робота.

Для успешно роботизированной сборочной операции обычно характерно наличие разнообразных тактильных и визуальных цепей обратной связи, позволяющих роботам контролировать результаты их деятельности и предпринимать соответствующие действия. Чтобы понять, зачем это необходимо, сравните выполняемую вручную операцию сверления отверстия со сборкой вручную двух сопрягаемых изделий. Операция сборки потребует большого «ощущения» объектов, некоторой ориентации деталей, вероятно, повторов и переориентации, а может быть даже некоторого рыскания вокруг в поисках утерянных деталей. По этой причине на сборочных позициях часто приходится устанавливать телекамеры технического зрения, чтобы обеспечить подачу входного сигнала в систему управления роботом, и, что более важно, оснащать руку самого робота тактильными датчиками силы. От этих датчиков сигналы передаются на систему управления, чтобы создать у робота «ощущение» того, что деталь находится в неправильном относительном положении и ему надо вернуться в исходное положение и попытаться снова. Робот можно даже запрограммировать таким образом, что он вернется к предыдущей операции сборки и мягко толкнет перекошенную деталь, установив ее в требуемое положение с тем, чтобы туда могла встать следующая по очереди деталь. На рис. 9.10 показана возможность выполнения такой операции на примере сборки зубчатого зацепления на сборочном заводе фирмы ИБМ в г. Остин, шт. Техас. При установке прямозубой шестерни датчик силы, установленный на захвате робота, сигнализирует системе управления о том, что испытываемое сопротивление слишком велико. Если сопротивление наблюдается в начале движения, правильным объяснением этому явлению будет то, что шестерня, установленная ранее, находится в неправильном положении. Затем робот может поднять руку, все еще удерживая шестерню, которую необходимо ставить, вернуть ее назад и мягко толкнуть ранее установленную шестерню, устанавливая ее на место, а затем снова попытаться вставить удерживаемую шестерню. Во второй попытке робот может снова почувствовать сопротивление в нижней части хода. Проконтролировав положение руки робота и получив одновременно сигнал от датчика силы, установленного на захвате, система управления робота может достоверно определить, что устанавливаемая шестерня находится в правильном положении, но для согласования положения зубьев необходимо ее слегка повернуть. Рука робота соответственно немного поворачивает шестерню и опускается до тех пор, пока положение руки не свидетельствует о том, что зубья находятся в зацеплении.

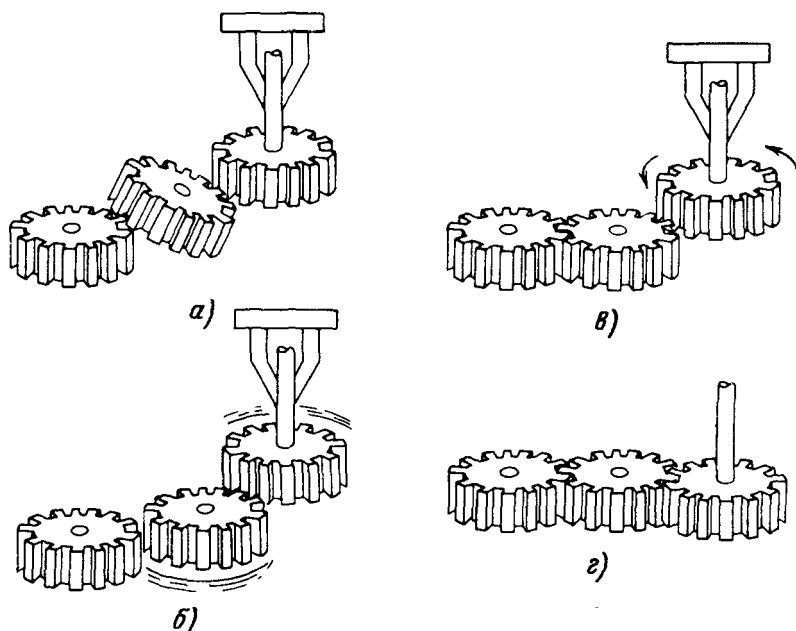


Рис. 9.10

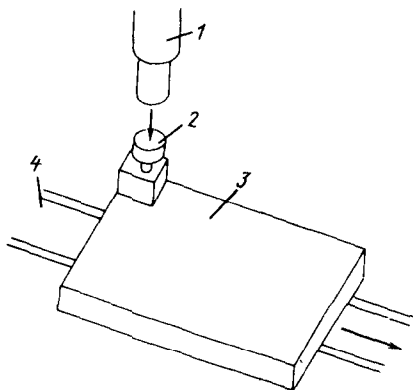
Датчики силы, установленные на захвате сборочного робота для контроля положения сопрягаемых прямозубых шестерен. Система управления робота запрограммирована на выполнение различных корректирующих воздействий:

а — датчик силы регистрирует сопротивление в начале движения; ранее установленная шестерня находится в неправильном положении; *б* — робот отступает, нависает над ранее установленной шестерней, слегка постукивает по ней, толкает и, может быть, слегка ее поворачивает, пока она не встанет в нужное положение; при этом третье колесо удерживается в захвате; *в* — робот приступает к установке третьей шестерни; если датчик силы опять регистрирует сопротивление, но в нижней части хода, робот просто слегка поворачивает шестерню, пока она не войдет в зацепление; *г* — сборка зубчатого зацепления завершена

Естественным последствием оснащения робота способностями видеть и осязать является применение этих способностей для контроля качества. Робот может распознать дефектные сборки и регистрировать их для принятия в будущем мер по их исправлению. Удобный прием для этого был применен на фирме ИБМ (рис. 9.11) при сборке катушек, на которые наматывается лента для пишущих машинок, выполняемой на шаговом конвейере, оснащенный палетами. Когда робот распознает дефектную сборку, он просто извлекает свою руку, выводит ее в запрограммированную позицию в углу палеты и нажимает на штырь или «регистрирующую кнопку», которая служит для идентификации дефектной сборки на конкретной палете. Информация, которая передается посредством регистрирующей кнопки, может быть передана и получена следующей автоматической сборочной установкой или использована как сигнал для перевода палеты на ответвление конвейера, где происходит исправление.

Рис. 9.11

Робот распознает дефектную сборку, подводит руку к углу палеты, установленной на конвейере, и нажимает на регистрирующую кнопку для идентификации брака и выполнения различных корректирующих воздействий: 1 — рабочий орган робота; 2 — регистрирующая кнопка; 3 — палета, несущая дефектную сборку; 4 — конвейер



Сборка самих пишущих машинок является сложной задачей, а значительное изменение стилей клавиатур и разнообразностей шрифтов от модели наталкивает на мысль о гибкой автоматизации при их сборке. Гибкий подход особенно необходим при сборке клавиш клавиатур терминалов. Вандербруг, Уилт и Дэвис в работе [75] рассказывают о системе для автоматической сборки клавиш, в которой применена телекамера, дающая возможность визуально проверять правильность выбора таста и выявлять значительные дефекты. Система, снабженная техническим зрением, набирает тасты в кассеты. Затем робот, работающий в прямоугольной системе координат и снабженный многозахватным рабочим органом, используется для одновременной установки нескольких таст, забираемых со стеллажа, на основании клавиатуры, подаваемое на столе, перемещающемся по направляющим. Главной особенностью конструкции, разработанной фирмой «Отоматикс Инкорпорейтед», было разделение системы на две части: установку для комплектования магазинов и роботизированную установку. Установка для комплектования магазина состоит из вибробункера и системы технического зрения, предназначенной для ориентации и контроля клавиши при комплектовании магазина. Укомплектованные магазины подаются вручную на стеллаж, используя для этого тележку, которая может вместить одновременно все магазины, необходимые для выполнения одного задания. Тележки, таким образом, становятся удобным буферным складом, расположенным между двумя установками. Преимущества, создаваемые применением буферных складов, очевидны из изложенного в п. 4. Еще одним преимуществом разделения установок является то, что если в будущем поставщик будет поставлять тасты, уложенные в магазины, необходимость в установке для комплектования магазинов отпадет. В следующем примере приведен анализ, который необходимо провести для того, чтобы согласовать работу двух автоматических сборочных систем с буферным складом между ними, как в случае роботизированной сборки клавиш пишущих машинок.

Пример 9.10. Согласование работы двух установок при сборке клавиш пишущих машинок. Установка для комплектования магазинов, используемых при

роботизированной сборке, состоит из вибробункера и системы технического зрения, предназначенной для ориентации и контроля деталей перед укладкой их в магазин емкостью $n_1 = 250$ шт., по одному магазину для каждого типа деталей. Магазины затем подают к стеллажу, где они располагаются в пределах досягаемости робота. Система «вибробункер—техническое зрение» может сортировать, проконтролировать детали и укомплектовать магазин за $t_1 = 10$ мин. Робот может собирать изделие, включающее $n_2 = 80$ деталей, с производительностью $P_1 = 50$ изделий. Сколько систем «вибробункер—техническое зрение» необходимо иметь, чтобы при параллельной работе они не отставали бы от робота?

Решение. Чтобы сортировать, проконтролировать и укомплектовать 80 магазинов, потребуется время $t_2 = n_2 \cdot t_1 = 80 \cdot 10 = 800$ мин. Робот освободит эти магазины за время $t_3 = n_1/P_1 = 250/50 = 300$ мин. Две установки, работая параллельно, могут укомплектовать магазины за время $t'_2 = 0,5t_2 = 400$ мин, но роботу все еще придется ждать довольно долго. Три установки сократят время комплектации магазина до $t'_2 = t_2/3 = 800/3 = 267$ мин, что позволит при идеальных условиях не отставать от робота.

Чтобы при комплектовании магазинов в соответствии с примером 9.10 оставалось время на манипуляции с дефектными, неправильно сориентированными или неправильно отобранными деталями, целесообразно иметь четыре системы «вибробункер—техническое зрение», параллельная работа которых позволила бы не отстать от робота. Соотношение стоимости оборудования для комплектования магазинов и стоимости роботизированной установки также влияет на выбор количества установок: их должно быть три или четыре.

Универсальная сборка. Предельной гибкостью обладала бы такая система, которая собирала бы любые изделия соответствующего размера при условии, что сборка изделия вообще осуществима, т. е. что изделие «собираемо». Эту цель можно считать претензионной, если принять во внимание, что крепежные приспособления и оснастка для сборки привязывают систему к конкретному изделию. Однако исследователи из международного Стенфордского исследовательского института при поддержке Национального фонда развития науки, не обескураженные сложностью стоящих проблем, приступили к разработке такой универсальной сборочной установки, состоящей из общего контроллера, модуля бинарного технического зрения, включающего в себя три телекамеры, и двух исполнительных модулей, именуемых «рука 1» и «рука 2» соответственно. Рабочий орган «руки 1» состоит из пластмассовой компенсирующей муфты, телекамеры и двухпалой руки. Рабочий орган «руки 2» состоит из шестиосевого датчика моментов и усилий, возникающих в запястье, и двухпалой руки. Смит и Нитцан в работе [65] сообщают, что универсальная сборочная установка характеризуется следующими особенностями.

Отсутствие специальных крепежных приспособлений: некоторые детали подаются к исполнительному модулю на палете, другие — на каретке. В сборке участвует простое приспособление, снабженное V-образными надрезами и служащее для центрирования цилиндрических деталей.

Использование макрокоманд среднего уровня типа MOVETO, FIND и GRASP.

Наличие бинарного технического зрения для определения положения и идентификации деталей на палете. Во время сборки используют три телекамеры: две прикрепленные к потолку и одну, установленную на рабочем органе одного из манипуляторов.

Наличие обратной связи по силе для обеспечения активного контроля податливости и сил сопротивления при сопряжении собираемых деталей.

Общая схема базирования для двух манипуляторов и телекамер, позволяющая осуществить установку деталей от одной общей базы.

Одновременная работа двух манипуляторов в определенной сборочной зоне, в то время как система бинарного технического зрения распознает и определяет положение деталей в любом месте.

Синхронизация движений манипуляторов, позволяющая избежать столкновений в центральной сборочной зоне.

Использование простых датчиков, позволяющих следить за выполнением операций, включая контроль за разведением пальцев для подтверждения наличия детали в руке и контроль защелкивания при помощи микрофона, установленного в сборочной зоне для подтверждения успешного соединения полужестких деталей или определения падения детали.

В ы в о д ы

Сварка является важнейшей областью применения промышленных роботов: в первую очередь точечная сварка, во вторую — дуговая. Применение роботов для точечной сварки началось с автомобилестроения, там же главным образом используют роботы для окраски распылением. Признавая достижения автомобилестроителей в использовании уже имеющихся у них роботов, следует отметить, что центр тяжести во внедрении роботов в настоящее время переместился на другие отрасли. Наблюдающийся в последнее время подъем в области роботизации дуговой сварки важен для судостроения, производства строительных конструкций, сельскохозяйственного машиностроения и множества других отраслей промышленности, где при изготовлении металлоизделий меньшего габарита применяют дуговую сварку. Использование защитной одежды при выполнении опасных операций дуговой сварки вызывает определенные неудобства, поэтому применение роботов позволяет значительно повысить производительность, поскольку роботы в среднем обеспечивают большее время горения дуги.

Загрузка оборудования является, вероятно, самой важной областью применения роботов вне автомобилестроения. Поскольку предполагалось, что большинство читателей книги не работают в автомобилестроении, загрузке в этой главе уделено особое внимание. Одной из побуждающих причин для автоматизации загрузки является вопрос обеспечения безопасности рабочих, осо-

бенно занимающихся штамповкой. Ключом к успеху при автоматизации многих операций загрузки с помощью промышленных роботов служит оснащение их двойными захватами. Если один робот должен последовательно обслуживать несколько установок, чтобы избежать дорогостоящего простоя оборудования в ожидании загрузки или разгрузки, часто требуется некоторым образом спланировать работы. Когда загружаемые детали нагреты, как, например, при ковке или литье под давлением, для применения при их загрузке роботов появляются особые причины.

Пожалуй, самой замечательной областью применения промышленных роботов является окраска распылением. Она, конечно же, требует использования самых совершенных типов роботов — роботов с контурным управлением. Было найдено, что согласованные, синхронные движения целой группы роботов для окраски имеют нечто общее с балетным представлением. Однако сбои в работе могут привести к возникновению повреждений и необходимости заново начинать цикл, что обходится недешево. Пример экономического анализа показал, что увеличение сменности повышает эффективность роботизированного производства, в частности автоматизированной окраски распылением.

Самой сложной задачей роботизации является автоматизация сборки. В этом случае робот обычно не может использовать свою способность работать в тяжелых условиях, ему приходится соревноваться с человеком в умелости, производительности и скорости. Человек обычно выигрывает это соревнование, однако некоторые роботы все же начинают завоевывать себе место при сборке двигателей автомобилей, и особенно при сборке электрического и электронного оборудования. Опыт, приобретенный инженерами при исследовании процесса роботизированной сборки, вероятно окажется полезным и для сборки вручную. Идеальные, абсолютно гибкие и универсальные роботизированные установки, не требующие специальной оснастки и приспособлений, пока еще не появились в промышленности. Знаменательно, однако, что такие установки находятся сейчас в стадии разработки в исследовательских лабораториях.

Упражнения и вопросы

9.1. Для каких технологических операций в настоящее время применяется наибольшее количество промышленных роботов? В какой отрасли промышленности работает наибольшее количество роботов?

9.2. Предположим, что новый робот для дуговой сварки обеспечивает увеличение времени горения дуги до 90 % при росте скорости процесса сварки на 25 %. Если три такие роботизированные установки могут работать одновременно с максимальной производительностью под надзором одного опытного сварщика, как повысится производительность при расчете на одного сварщика по сравнению с обычно применяемыми методами сварки?

9.3. Каковы главные побудительные причины использования промышленных роботов для загрузки и разгрузки штамповочных прессов?

9.4. Какие существуют особые причины использования роботов для загрузки при ковке и литье под давлением?

9.5. Для промышленного робота среднее время между отказами составляет 1000 ч. Каждый отказ приводит к простоям длительностью в среднем 16 ч. Каков коэффициент готовности робота? Если годовой фонд рабочего времени для этого робота составляет 4000 ч, сколько часов в год он работоспособен?

9.6. Пять роботов загружают и разгружают пять установок, обслуживаемых проходящим вдоль них непрерывным конвейером. Если технологический маршрут проходит последовательно через все пять установок, а длительность цикла обработки на каждой установке составляет 12 с, включая загрузку и разгрузку, какова будет идеальная производительность за одну восьмичасовую смену?

9.7. Для линии из пяти установок и пяти роботов, описанной в примере 9.6, предположим, что для каждого из роботов средняя наработка на отказ составляет 1000 ч и что отказ робота приводит к простоям средней продолжительностью 16 ч. Предположим также, что при отсутствии промежуточных складов между установками отказ любого из роботов вызовет остановку всей линии, которая будет продолжаться, пока робот не исправят. При той же длительности цикла, что и раньше, составляющей 12 с, каков будет выпуск продукции за восьмичасовую смену с учетом возможных отказов роботов?

9.8. Пять роботов загружают и разгружают пять установок, обслуживаемых общим конвейером. Хотя все пять установок осуществляют те же операции, они работают параллельно и независимо друг от друга. Если длительность цикла, включая загрузку и разгрузку, составляет 12 с, какова будет идеальная производительность за одну восьмичасовую смену?

9.9. Для производственного участка, описанного в примере 9.8, предположим, что у каждого из роботов среднее время между отказами составляет 1000 ч и что отказ робота приводит к простоям со средней продолжительностью 16 ч. При длительности цикла, составляющей 12 с, каков будет выпуск продукции за восьмичасовую смену с учетом возможных отказов роботов?

9.10. Один робот обслуживает три станка, каждый из которых обрабатывает заготовки последовательно, аналогично тому, как показано на рис. 9.7. Время, затрачиваемое роботом на обслуживание каждого из станков, складывается из времени на захват детали (0,1 мин), освобождение детали (0,1 мин) и соответствующего времени перемещения. Время перемещения захвата робота указано в табл. 9.1. Длительность обработки приведена ниже:

Время обработки на первом станке (позиция Б)	6 мин
Время обработки на втором станке (позиция В)	9 мин
Время обработки на третьем станке (позиция Г)	5 мин

Приняв коэффициент готовности робота 100 %, определите длительность цикла системы. Чему равна производительность системы?

9.11. Если среднее время между отказами для робота в упражнении 9.10 составляет 600 ч, а на устранение неисправности уходит 10 ч, чему окажется равной длительность цикла? Какова будет в этом случае производительность?

10. ПРОМЫШЛЕННЫЕ ЛОГИЧЕСКИЕ СИСТЕМЫ УПРАВЛЕНИЯ

Появление на современных предприятиях промышленных роботов, этих «механических» рабочих, дало поразительные результаты, однако, если внимательнее приглядеться к автоматизации, то можно обнаружить более скромную, но не менее важную роль, которую играют логические системы управления. Их основой являются программируемые контроллеры (ПК), детально рассмотренные в гл. 12.

Слово *робот* знакомо практически каждому, но вряд ли каждый может объяснить, что такое «программируемые контроллеры», «цифровая электроника» или «логические системы управления». Однако все эти термины относятся к основным элементам, которые повсеместно используются в автоматических системах, в том числе и во многих роботах. Данная глава имеет целью объяснить, что такое логическое управление и почему оно столь важно для автоматизации производства и робототехники.

В состав промышленного предприятия входят машины, процессы, люди, сырье и готовая продукция, причем все эти компоненты взаимодействуют в соответствии с планами, целями, инструкциями и режимами производства. Выполнению этих планов и целей препятствуют некачественное сырье, поломки машин, различные помехи для людей и процессов, бракованная продукция, простои и прочие неприятности, присущие реальному миру. Производственная система должна учитывать эти неприятности, принимая соответствующие решения и осуществляя корректирующие действия; при этом не имеет значения, кем принимаются решения — людьми или машинами. Кроме анализа этих неприятностей система, принимающая решения, должна задавать обычный и естественный порядок действий: когда необходимо выполнить следующую операцию и какую именно.

Решения, принимаемые на производстве, иногда делят на две большие категории: 1) решения по основным признакам и 2) решения по значениям переменных.

В сфере приемки продукции и контроля качества решения по основным признакам имеют вид «Принять/Отклонить» или «Годен/Не годен», тогда как решения по значениям переменных основаны на вопросах типа:

Рис. 10.1

Небольшая печь для предварительного разогрева
кузнечных заготовок

1. Насколько высока температура?
2. Каков вес?
3. Какова длина заготовки?

Контроль качества — это не единственная сфера, где решения принимаются по основным признакам и значениям переменных. Рассмотрим следующие вопросы, подразумевающие принятие решений людьми или машинами:

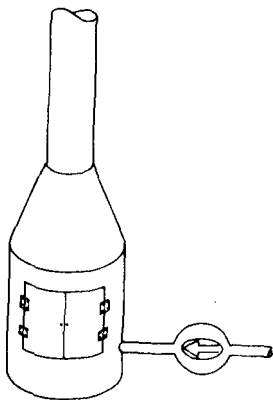
1. Действует ли подающий конвейер на линии разлива № 2?

2. Сколько имеется в наличии контейнеров, используемых на линии упаковки?

Если на вопрос можно ответить «да» или «нет», это значит, что он имеет отношение к основным признакам и открывает прямой путь к автоматизации с помощью логической системы управления.

Даже если вопрос имеет отношение к значениям переменных и ответ на него выражается числом или количеством вместо «да» или «нет», его можно свести к цепочке принятия решений да/нет и тем самым установить искомое значение. Таким образом, промышленная логическая система управления может быть использована непосредственно — в случае принятия решений по основным признакам, а также косвенно — в случае принятия решений по значениям переменных. Это будет проиллюстрировано ниже рядом примеров.

Простой пример поможет нам вывести из сферы абстракции понятие о логических системах управления. На рис. 10.1 схематически изображена небольшая печь для предварительного разогрева кузнечных заготовок. В печь подается природный газ; в случае исчезновения пламени необходимо вновь зажечь горелку, а если это не удастся, то подача горючего должна быть прекращена во избежание серьезной аварии. При грубом ручном управлении рабочий может контролировать пламя и изменять положение вентилей, регулирующих подачу топлива. Однако критерии эффективности и безопасности требуют, чтобы система управления автоматически контролировала состояние процесса и при необходимости оперативно вмешивалась в его ход. Разумеется, в ручных действиях необходимость все же остается хотя бы потому, что автоматическая система запускается и останавливается двухпозиционным переключателем. Систему можно усложнить путем применения таймера, автоматически регулирующего цикл зажигания. Это усложнение требует, однако, тщательного предварительного исследования процесса с целью установить все логические связи



между отдельными переменными. Для этого можно воспользоваться несколькими методами; мы рассмотрим их в данной главе и начнем с так называемых «таблиц истинности».

10.1. ТАБЛИЦЫ ИСТИННОСТИ

Таблица истинности — это матрица, устанавливающая связь между всеми возможными комбинациями логических переменных и соответствующими им значениями функции. Для иллюстрации мы воспользуемся еще одним примером, который даже проще, чем рассмотренная выше система управления печью.

Пульт управления с блокировкой. Предположим, что производственный процесс управляется с пульта, и мы хотим ограничить доступ к пулту путем блокировки, ключ к которой имеется только у определенных лиц. Для наибольшей простоты предположим, что на пульте имеется всего один орган управления — двухпозиционный переключатель (положения ВКЛ/ВЫКЛ), не считая замка блокировки (рис. 10.2). Подобный пульт с блокировкой представляет собой логическую систему управления с переменными дискретного типа, каждая из которых может принимать два и только два значения или состояния. Эти переменные и их состояния определим следующим образом.

Переменная S характеризует положение переключателя: $S = 0$, если переключатель находится в положении ВЫКЛ, $S = 1$, если переключатель находится в положении ВКЛ. Переменная L характеризует состояние блокировки: $L = 0$, если ключ отсутствует (блокировка включена), $L = 1$, если ключ присутствует (блокировка снята).

Переменная P отражает состояние процесса: $P = 0$, если процесс остановлен, $P = 1$, если процесс действует.

Цифры 0 и 1 обычно используются для представления двух состояний логических переменных. Иногда встречаются и иные обозначения, например: Истина/Ложь, ВКЛ/ВЫКЛ.



Рис. 10.2

Пульт с блокировкой: переключатель управляет процессом, только если блокировка снята

10.1. Таблица истинности

Входы			Выход P
S	L	P	
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

После определения переменных все их возможные комбинации сводятся в таблицу истинности. Входные переменные располагаются столбцами в левой части матрицы, а выходные — в правой.* Табл. 10.1 представляет собой таблицу истинности для примера с пультом с блокировкой.

В примере с пультом мы имеем три переменных: три входных и одну выходную. Это выглядит противоречием, поскольку три плюс один будет *четыре*, а не три. Но все объясняется тем, что переменная *P*, соответствующая состоянию процесса, является одновременно как входной, так и выходной. В этом есть свой смысл, так как, если ключ не вставлен в замок, то мы хотим, чтобы процесс оставался остановленным, если он уже был остановлен, или продолжал функционировать, если он действовал и до этого, каким бы ни было положение переключателя ВКЛ/ВЫКЛ. Иначе говоря, текущее состояние процесса *P* (вход) может оказывать влияние на его будущее состояние (выход).

Таблица истинности читается построчно. Например, третью строку в табл. 10.1 следует трактовать так:

ЕСЛИ переключатель ВКЛ/ВЫКЛ находится в положении ВЫКЛ

И ЕСЛИ ключ не вставлен в замок

И ЕСЛИ процесс остановлен

ТО процесс должен оставаться остановленным.

В общем случае таблицы истинности могут содержать любое число входных и выходных переменных, при этом каждая строка интерпретируется следующим образом:

ЕСЛИ (вход 1) И ЕСЛИ (вход 2) ... И ЕСЛИ (вход *n*)

ТО (выход 1) И (выход 2) ... (выход *m*),

где *n* — число входных, а *m* — число выходных переменных логической системы управления.

Строка 4 в табл. 10.1 на первый взгляд кажется парадоксом: как может переменная *P* в одной и той же строке иметь значения 1 и 0? Это кажущееся противоречие объясняется тем, что в одном случае переменная *P* выступает как входная, а в другом — как выходная. Словесная интерпретация четвертой строки табл. 10.1 выглядит так:

ЕСЛИ переключатель ВКЛ/ВЫКЛ находится в положении ВЫКЛ

И ЕСЛИ ключ находится в замке

И ЕСЛИ процесс действует

ТО процесс должен быть остановлен (немедленно).

Во избежание недоразумений значения входных переменных целесообразно рассматривать как имеющие место в момент времени, *предшествующий* соответствующим выходным переменным

* Иногда в таблице истинности переменные располагаются в виде строк, а не столбцов. При этом в верхней части матрицы помещают входные переменные, а в нижней — выходные.

в той же самой строке таблицы истинности. В гл. 12 мы увидим, что в реальных логических системах управления на преобразование входных переменных в выходные затрачивается очень малое, но все же конечное время. Иногда это тонкое различие между входным и выходным значениями одной и той же переменной подчеркивают обозначениями, скажем, P — для входа и P' — для выхода, или наоборот.

Последовательность строк (или столбцов) в таблице истинности является совершенно произвольной, и выбору этой последовательности не стоит придавать слишком серьезного значения. Иногда бывает полезно придерживаться естественной последовательности, характеризующей цепочку переходов системы из одного состояния в другое в процессе ее работы. Но подобная стратегия имеет один недостаток: легко проглядеть необычные состояния, в которых может оказаться система. Другая стратегия основана на использовании упорядоченной последовательности, исчерпывающей все возможные сочетания входных переменных. Такая стратегия использована при составлении табл. 10.1. Поскольку каждая переменная может принимать значения 0 или 1, очень удобно изображать каждую комбинацию входных переменных как число в двоичной системе счисления. При такой стратегии таблица истинности всегда начинается со строки, состоящей из нулей, и заканчивается строкой, состоящей из единиц. Именно так и выглядит табл. 10.1. Отметим, что в табл. 10.1 начальная строка, состоящая из нулей, включает в себя и элемент столбца выхода; то же самое можно сказать и о последней строке, состоящей из единиц. Однако это свойство имеет частный характер и обусловлено логикой работы пультa с блокировкой. В других практических приложениях это правило может и не выполняться.

Часто случается так, что таблица истинности должна была бы представлять собой длинный список входных переменных, но с учетом специфики исследуемого производственного процесса имеет смысл рассматривать лишь ограниченный набор комбинаций из значений этих переменных. В подобных случаях нецелесообразно пытаться составить все возможные комбинации значений входных переменных, используя двоичную систему счисления. Здесь следует отдать предпочтение описанной выше стратегии, при которой рассматривается естественная последовательность состояний системы.

Кнопочный выключатель. Большинство станков на производстве включаются и останавливаются с помощью пружинных кнопок, причем пуск и останов производятся разными кнопками (рис. 10.3). Это оправдано соображениями безопасности, так как кнопке СТОП можно дать преимущественное право остановки станка в экстренных случаях независимо от положения кнопки ПУСК. Логика работы кнопочного выключателя ПУСК/СТОП рассматривается здесь как еще один пример использования таблицы истинности. Входные и выходные переменные определим следующим образом.

Входы			Выход R'
G	S	R	
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

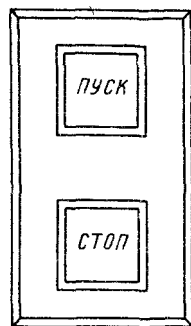


Рис. 10.3

Кнопочный выключатель: кнопка ПУСК включает систему, последняя работает до тех пор, пока не будет нажата кнопка СТОП. Кнопка СТОП имеет приоритет над кнопкой ПУСК

Переменная G характеризует положение кнопки ПУСК: $G = 1$, если кнопка ПУСК нажата, $G = 0$, если кнопка ПУСК отжата. Переменная S характеризует положение кнопки СТОП: $S = 1$, если кнопка СТОП нажата, $S = 0$, если кнопка СТОП отжата. Переменная R характеризует состояние станка: $R = 0$, если станок не работает, $R = 1$, если станок работает.

Табл. 10.2 представляет собой таблицу истинности для примера с кнопочным выключателем.

Сравнение табл. 10.2 с табл. 10.1 обнаруживает несколько интересных различий между примером с кнопочным выключателем и рассмотренным ранее примером с пультом с блокировкой. Прежде всего заметим, что в табл. 10.2 использовано обозначение R' , позволяющее отличить выходную переменную от входной переменной R . Отметим, что для обоих примеров таблицы истинности составлены по одному и тому же принципу, а именно путем упорядочения в двоичной системе всех комбинаций входных переменных. Однако обратим внимание на различия в значениях выходной переменной. Эти различия обусловлены логикой принципа действия пульта с блокировкой и кнопочного выключателя. Возможные варианты таблицы истинности ограничены лишь фантазией инженера по автоматике, работающего в сфере практического применения логических систем управления. И, наконец, мы обнаруживаем, что последняя строка в табл. 10.2 не вся состоит из единиц, как это было в табл. 10.1. Читателю необходимо осмыслить логику работы кнопочного выключателя, чтобы убедиться, что при входной комбинации 1—1—1 выход должен быть нулем, а не единицей.

10.2. БУЛЕВА АЛГЕБРА

Наряду с таблицами истинности для определения логических функций входных переменных можно воспользоваться алгебраическими выражениями. Алгебра логики называется также булевой

Оператор	Символ	Пример	Объяснение
И	$\cdot$	$A \cdot B$	Как A , так и B должны быть истинны, чтобы все выражение было истинно
ИЛИ	$+$	$A + B$	Если или A , или B , или обе истинны, то истинно и все выражение
НЕ	$-$	$\overline{A}$	Если A истинно, то выражение ложно; если A ложно, то выражение истинно

алгеброй (по имени английского математика и логика XIX в. Джорджа Буля), она похожа на обычную алгебру, но в то же время отличается от нее в основных положениях. Основы булевой алгебры излагаются здесь с минимумом теории, при этом преследуются две цели:

1. Дать возможность инженеру по автоматике в лаконичной форме описать логические соотношения между переменными в системе.

2. Продемонстрировать применение теорем и законов булевой алгебры для упрощения логических систем управления.

Основные операторы. Для образования булевых логических выражений необходимы всего лишь три основных оператора, представленных в табл. 10.3.

Может показаться несколько странным, что заимствованный из обычной алгебры знак плюс используется для логического ИЛИ, тогда как естественнее было бы использовать его для логического И. Кроме того, точка, которая в обычной алгебре является символом умножения, в булевой алгебре используется для обозначения логического И. В алгебре логики можно пользоваться терминами *множители* и *слагаемые* в их обычном смысле, если, конечно, представлять себе все отличия от обычной алгебры.

При интерпретации булевых логических выражений смысл содержащихся в них операторов становится более понятным, если каждый множитель представлять как *необходимое условие*, а каждое слагаемое — как допустимую *альтернативу* для получения результата.

Проиллюстрируем с помощью нескольких примеров смысл логических алгебраических операторов. Простейшим выражением алгебры логики является (10.1), в котором выходная (зависимая) переменная Y является функцией двух входных (независимых) переменных A и B :

$$Y = A \cdot B. \quad (10.1)$$

Это выражение говорит о том, что *обе* входные переменные, A и B , должны быть истинными (т. е. иметь значения, равные еди-

10.4

Входы		Выход Y
A	B	
0	0	0
0	1	0
1	0	0
1	1	1

10.5

Входы			Выход Y
A	B	C	
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

нице), для того чтобы выходная переменная также была истинной (что соответствует логической единице). Поскольку выражение содержит только одно слагаемое, альтернативы не существует: только определенная комбинация A и B может дать результат Y .

Выражение (10.2) создает альтернативу для получения выхода Y :

$$Y = (A \cdot B) + C. \quad (10.2)$$

Согласно этому выражению выход Y будет иметь значение, равное единице, если равна единице переменная C , даже когда остальные переменные, A и B , не равны единице. Выражение (10.2), которое читается как « A и B или C », представляет собой менее строгое условие формирования выходной переменной Y , нежели (10.1).

Оператор НЕ просто изменяет значение выражения на обратное, т. е. если оно равно нулю, то оператор НЕ делает его равным единице, и наоборот. Скобки в булевых выражениях имеют тот же смысл, что и в обычных алгебраических выражениях.

Связь с таблицей истинности. Булево логическое выражение полностью может быть представлено таблицей истинности. Все переменные, входящие в выражение, заносятся в таблицу в качестве входных переменных, а значение всего выражения — в качестве выходной переменной. Так, в булевых выражениях (10.1) и (10.2) переменная, стоящая слева от знака равенства, в таблице истинности должна играть роль выходной переменной. Табл. 10.4 и 10.5 представляют собой таблицы истинности для выражений (10.1) и (10.2) соответственно.

И наоборот, для любой таблицы истинности можно записать соответствующее булево логическое выражение. Возвращаясь к примеру с пультом с блокировкой, запишем булево логическое выражение, воспользовавшись табл. 10.1. Чтобы не спутать входную и выходную переменные, обозначенные в таблице одним символом P , присвоим выходной переменной обозначение P' . Из

Входы			Выходы					
A	B	C	ABC	A+B+C	$\overline{ABC}$	$\overline{A+B+C}$	$\overline{AB}$	$A(B+\overline{C})$
0	0	0	0	0	1	1	0	0
0	0	1	0	1	1	0	0	0
0	1	0	0	1	1	0	0	0
0	1	1	0	1	1	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	0	1	1	0	1	0
1	1	0	0	1	1	0	0	1
1	1	1	1	1	0	0	0	1

табл. 10.1 видно, что четыре значения выхода P' равны единице, а остальные являются нулями. Каждая из единиц соответствует одной возможной альтернативе образования этого значения; следовательно, в булевом выражении необходимо предусмотреть оператор ИЛИ, воздействующий на четыре слагаемых. Каждое из этих слагаемых должно представлять собой определенную комбинацию входных переменных, соответствующую данной строке таблицы истинности. Например, вторая строка табл. 10.1 говорит о том, что одна из возможностей образования логической единицы для выхода P' — это равенство нулю входных переменных S и L и равенство единице входной переменной P . Полностью булево выражение имеет следующий вид:

$$P' = (\bar{S} \cdot \bar{L} \cdot P) + (S \cdot \bar{L} \cdot P) + (S \cdot L \cdot \bar{P}) + (S \cdot L \cdot P). \quad (10.3a)$$

Как и в обычной алгебре, скобки и символы умножения часто опускают, тогда (10.3a) принимает вид

$$P' = \bar{S}\bar{L}P + S\bar{L}P + SL\bar{P} + SLP. \quad (10.3b)$$

Ниже будет показано, что выражение (10.3b) можно упростить, воспользовавшись теоремами булевой алгебры.

В качестве примера приводится таблица истинности (табл. 10.6) с тремя входами и шестью различными столбцами для выхода, соответствующими разным булевым выражениям.

Упрощение алгебраических выражений. В булевой алгебре имеются теоремы и законы, похожие на законы обычной алгебры, но в то же время отличающиеся от них в некоторых существенных моментах. Поводом для рассмотрения здесь этих законов и теорем является желание вооружить инженера по автоматике методами упрощения логических выражений и, следовательно, технических средств, необходимых для решения прикладных задач.

Чтобы показать, как применяются теоремы булевой алгебры, начнем с простого примера на преобразование правой части выражения (10.4):

$$X = A \cdot (B + \bar{B}). \quad (10.4)$$

Это выражение в словесной интерпретации означает:

Выход X будет равен 1, если

1. Вход A равен 1

И

2. Вход B равен либо 1 либо 0.

Совершенно очевидно, что если вдуматься в эти слова, то выражение (10.4) можно существенно упростить. Напомним, что логическая переменная может принимать только два значения: 0 или 1. Следовательно, B может быть либо единицей, либо нулем. Но выражение (10.4) содержит множитель $(B + \bar{B})$. Оператор ИЛИ (+) говорит нам о том, что B либо существует, либо не существует. Поскольку какое-то одно из этих взаимоисключающих событий всегда имеет место, то соответствующую часть выражения можно просто уничтожить без последствий для выхода X . Выражение (10.4) тогда приводится к виду

$$X = A. \quad (10.5)$$

Таким образом, если значение A равно единице, то и X будет единицей; если же A есть нуль, то и X будет нулем. Выражение (10.4) удалось упростить благодаря применению двух простейших теорем булевой алгебры:

1. Теорема включения: $B + \bar{B} = 1$.

2. Характеристическая теорема: $A \cdot 1 = A$.

На практике большинство логических выражений гораздо сложнее, нежели (10.4), но с помощью законов и теорем булевой алгебры многие из них можно значительно упростить.

Наиболее часто используемые законы и теоремы булевой алгебры сведены в табл. 10.7, которая может оказаться полезной при решении практических задач логического управления, описанных в данной книге. Как и в обычной алгебре или любой другой науке, основные аксиомы служат для доказательства теорем; те, в свою очередь, используются для доказательства других теорем и законов и т. д. Данная книга не ставит целью углубляться в математические доказательства, но чтобы проиллюстрировать метод, приведем лишь одно из них:

ДОКАЗАТЬ

Закон поглощения $X + XY = X$.

ДОКАЗАТЕЛЬСТВО

$$\begin{aligned} X + XY &= X \cdot 1 + X \cdot Y && \text{(На основании характеристической теоремы)} \\ &= X(1 + Y) && \text{(На основании закона дистрибутивности)} \\ &= X \cdot 1 && \text{(На основании характеристической теоремы)} \\ &= X && \text{(На основании характеристической теоремы)} \end{aligned}$$

Что и требовалось доказать.

Характеристические теоремы

1. $X \cdot 0 = 0$

2. $X \cdot 1 = 1$

3. $X + 0 = X$

4. $X + 1 = 1$

Закон коммутативности

1. $X + Y = Y + X$

2. $X \cdot Y = Y \cdot X$

Закон ассоциативности

1. $X + Y + Z = X + (Y + Z) = (X + Y) + Z$

2. $X \cdot Y \cdot Z = X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$

Закон дистрибутивности

1. $X \cdot Y + X \cdot Z = X (Y + Z)$

2. $(X + Y) (W + Z) = XW + XZ + YW + YZ$

Теоремы идемпотентности

1. $X \cdot X = X$

2. $X + X = X$

Теорема отрицания отрицания

$(\bar{X}) = X$

Теоремы включения

1. $X \cdot \bar{X} = 0$

2. $X + \bar{X} = 1$

Законы поглощения

1. $X + XY = X$

2. $X (X + Y) = X$

Теоремы отражения

1. $X + \bar{X}Y = X$

2. $X (\bar{X} + Y) = XY$

3. $XY + XYZ = XY + YZ$

Теоремы склеивания

1. $XY + X\bar{Y} = X$

2. $(X + Y) (X + \bar{Y}) = X$

Законы де Моргана

1. $\overline{XY} = \bar{X} + \bar{Y}$

2. $\overline{X + Y} = \bar{X}\bar{Y}$

Лучший способ отточить знания булевой алгебры — это попытаться доказать все теоремы и законы, представленные в табл. 10.7. Основные из них либо интуитивно очевидны, либо должны быть приняты как аксиомы. Сюда относятся законы коммутативности, ассоциативности и дистрибутивности, а также теоремы: характеристические, включения, идемпотентности и отрицания отрицания. Читатель, вероятно, уже успел заметить, что многие основные законы обычной алгебры имеют тот же смысл и в булевой алгебре (например, законы ассоциативности, коммутативности и дистрибутивности). Однако не следует слишком полагаться на знание обычной алгебры. Для примера рассмотрим детальнее четыре варианта характеристической теоремы из табл. 10.7. Первые три варианта вписываются в рамки обычной алгебры, но четвертый ($X + 1 = 1$) *противоречит* ее правилам. Инженер по автоматике, чтобы не попасть в ловушку, применяя правила обычной алгебры к логическим выражениям, прежде всего должен проверить, действует ли данное правило в алгебре логики.

Опуская доказательства теорем булевой алгебры, воспользуемся ими с целью упрощения логического выражения (10.36), полученного ранее для пультa с блокировкой. Это выражение образовано на основании соответствующей таблицы истинности, в которой выход формируется как функция трех входных переменных. Оно является неоправданно длинным и сложным; с помощью законов и теорем булевой алгебры это выражение значительно упрощается:

$$\begin{aligned} P' &= \bar{S}\bar{L}P + \bar{S}LP + SL\bar{P} + SL P = (\text{исходное выражение 10.36}) \\ &= \bar{L}P (\bar{S} + S) + SL (\bar{P} + P) = (\text{закон дистрибутивности}) \end{aligned}$$

Рис. 10.4

Схематическое изображение настольного автоматического пресса:

1 — пружинный возвратный механизм; 2 — плунжер

$$= \bar{L}P(1) + SL(1) = (\text{теорема включения})$$

$$= \bar{L}P + SL = (\text{характеристическая теорема})$$

Пример 10.1. Управление автоматическим прессом. Для иллюстрации положений данной главы рассмотрим упрощенный вариант задачи из области металлообработки — логическое управление автоматическим прессом. Пресс, схематически изображенный на рис. 10.4, представляет собой настольную модель для небольших работ; он приводится в действие сжатым воздухом или электромагнитом. Пресс запускается кнопкой ПУСК и для надежности останавливается одной из трех кнопок СТОП, удобно расположенных в различных местах станка. Если пресс включен, то он автоматически воздействует на плунжер всякий раз, когда концевой выключатель $KB1$ отклонен вверх. Пресс сохраняет давление на плунжер во время его хода вниз до тех пор, пока не будет отклонен вниз концевой выключатель $KB2$. В этот момент давление на плунжер прекращается, и он с помощью пружинного механизма возвращается в крайнее верхнее положение, когда $KB1$ вновь окажется отклонен вверх. Затем плунжер снова приводится в действие и так автоматически совершает движения вверх и вниз до тех пор, пока пресс не будет выключен одной из кнопок СТОП.

У описанного здесь автоматического пресса имеется всего один выход, который мы назовем «состояние плунжера». В то же время входов несколько, и все они отражены в табл. 10.8. Заметим, что переменная A (состояние плунжера) является входной, но в то же время и выходной переменной. Поэтому, как и в примере с пультом с блокировкой, мы будем использовать обозначение A для входной переменной и A' — для выходной.

Составление таблицы истинности, учитывающей все возможные комбинации переменных, перечисленных в табл. 10.8, было бы весьма трудоемкой процедурой. При наличии семи входов таблица истинности должна была бы состоять из $2^7 = 128$ строк! Но мы можем упростить задачу, сведя ее к трем небольшим подзадачам.

В первой подзадаче мы рассмотрим действие кнопок СТОП. Образует таблицу истинности (табл. 10.9), введя вспомогательную переменную S , значение которой отражает состояние трех переменных S_1 , S_2 и S_3 . Этой задаче соответствуют логические выражения:

$$\bar{S} = \bar{S}_1 \bar{S}_2 \bar{S}_3; \text{ на основании таблицы истинности}$$

$$\bar{S} = S_1 + S_2 + S_3; \text{ на основании законов де Моргана}$$

$$S = S_1 + S_2 + S_3. \quad (10.6)$$

10.8

Переменная	Обозначение	Значение
Положение кнопки ПУСК	G	$= 1$ (кнопка нажата) $= 0$ (кнопка отжата)
Положение кнопки СТОП ($i = 1, 2, 3$)	S_i	$= 1$ (кнопка нажата) $= 0$ (кнопка отжата)
Положение концевого выключателя ($i = 1, 2$)	L_i	$= 1$ (КВ отклонен) $= 0$ (КВ не отклонен)
Состояние плунжера	A	$= 1$ (плунжер действует) $= 0$ (плунжер не действует)

10.9

Входы			Выход S
S_1	S_2	S_3	
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

10.10

Входы			Выход R'
G	S	R	
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Заметим, что в данном примере проще начать с переменной $\bar{S}$, так как первое выражение для $\bar{S}$ содержит всего одно слагаемое; только одна строка таблицы истинности приводит к образованию $\bar{S}$.

В качестве следующего шага образуем таблицу истинности, с помощью которой можно было бы определить, работает или не работает пресс. Для этого введем еще одну переменную R : $R = 1$, если пресс работает, $R = 0$, если пресс не работает. R является как выходной, так и входной переменной, поэтому для выхода мы будем использовать обозначение R' . Табл. 10.10 есть таблица истинности, отражающая состояние пресса. Подчеркнем, что одна из *входных* переменных (S) в табл. 10.10 является *выходной* переменной в табл. 10.9. Табл. 10.10 поставим в соответствие логическое выражение:

$$\begin{aligned}
 R' &= \bar{G}\bar{S}R + \bar{G}\bar{S}\bar{R} + G\bar{S}R = \text{по таблице истинности} \\
 &= \bar{G}\bar{S}R + \bar{G}\bar{S}(\bar{R} + R) = \text{закон дистрибутивности} \\
 &= \bar{G}\bar{S}R + \bar{G}\bar{S}(1) = \text{теорема включения} \\
 &= \bar{G}\bar{S}R + \bar{G}\bar{S} = \text{характеристическая теорема} \\
 &= \bar{S}(\bar{G}R + G) = \text{законы коммутативности и дистрибутивности} \\
 &= \bar{S}(R + G) = \text{закон коммутативности и теорема отражения}
 \end{aligned} \tag{10.7}$$

И наконец, мы можем образовать еще одну таблицу истинности (табл. 10.11), которая включает в себя в качестве входной переменной выходную переменную R' из табл. 10.10 наряду с тремя оставшимися входными переменными рассматриваемой задачи. Поскольку теперь мы имеем четыре входных переменных, то табл. 10.11 состоит из $2^4 = 16$ строк. Логическое выражение, соответствующее табл. 10.11, получается следующим образом:

$$\begin{aligned}
 A' &= R'\bar{L}_1\bar{L}_2A + R'L_1\bar{L}_2\bar{A} + R'\bar{L}_1\bar{L}_2A = \text{по таблице истинности} \\
 &= R'\bar{L}_1\bar{L}_2A + R'L_1\bar{L}_2(\bar{A} + A) = \text{закон дистрибутивности} \\
 &= R'\bar{L}_1\bar{L}_2A + R'L_1\bar{L}_2(1) = \text{теорема включения} \\
 &= R'\bar{L}_1\bar{L}_2A + R'L_1\bar{L}_2 = \text{характеристическая теорема} \\
 &= R'\bar{L}_2(\bar{L}_1A + L_1) = \text{законы коммутативности и дистрибутивности} \\
 &= R'\bar{L}_2(A + L_1) = \text{закон коммутативности и теорема отражения}
 \end{aligned} \tag{10.8}$$

Теперь на основании всех трех таблиц истинности можно получить итоговое логическое выражение. Подставляя (10.7) в (10.8), имеем

$$A' = \bar{S}(R + G)\bar{L}_2(A + L_1).$$

Входы				Выход A'
R'	L_1	L_2	A	
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Используя (10.6), окончательно получим

$$A' = \overline{S_1 + S_2 + S_3} (R + G) \bar{L}_2 (A + L_1). \quad (10.9)$$

Выражение (10.9) полностью описывает логику функционирования автоматического пресса. В гл. 11 и 12 мы рассмотрим, как это логическое выражение может быть использовано в качестве модели для проектирования реальной системы управления прессом.

В ы в о д ы

В этой главе мы рассмотрели, из каких основных блоков образуются логические выражения и как эти выражения можно использовать в качестве моделей промышленных логических систем автоматического управления.

Наиболее полное представление о взаимосвязях между входными и выходными переменными дает таблица истинности. Необходимо определенная система, позволяющая отразить в таблице истинности все комбинации входных переменных. На примерах с пультом с блокировкой и кнопочным выключателем мы убедились, что при определении последующего логического состояния выходной переменной она одновременно может играть роль входной переменной.

Булева алгебра позволяет не только записывать логические выражения, но и приводить их к наиболее простому виду. Подобное упрощение открывает прямой путь к повышению экономичности и надежности промышленных логических систем управления.

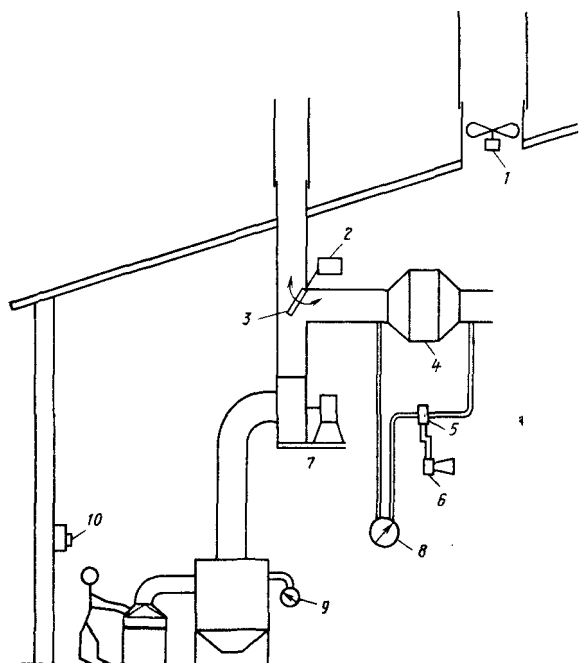


Рис. 10.5

Схема автоматической системы управления вентиляцией:

1 — вытяжка; 2 — привод заслонки; 3 — перепускной клапан; 4 — мощный фильтр; 5 — реле давления; 6 — сигнализация; 7 — устройство подачи воздуха; 8 — манометр; 9 — дифференциальный манометр на цеховом коллекторе; 10 — пульт управления

В гл. 11 мы рассмотрим, как на основе булевой алгебры создаются принципиальные схемы промышленных логических систем управления, что приблизит нас к реальным физическим представлениям об этих системах.

Упражнения и вопросы

10.1. Станок включается кнопкой ПУСК и выключается любой из двух кнопок СТОП. Составьте таблицу истинности и получите логическое выражение для выходной переменной «станок работает». С помощью булевой алгебры максимально упростите логическое выражение.

10.2. Запишите логическое выражение, соответствующее таблице истинности (табл. 10.2) для кнопочного выключателя. Минимизируйте это выражение с помощью булевой алгебры.

10.3. Какие из следующих булевых выражений истинны и какие ложны?

- $A + B = A + \bar{A}B$;
- $\bar{A}\bar{B} = \bar{A}\bar{B}$;
- $A = A + AB + ABX$;
- $X = X(Y + \bar{Y}) + X(Z + \bar{Z})$;
- $Y = Y(X + \bar{X}) + XZZ$.

10.4. Составьте таблицы истинности для следующих функций входных переменных A , B и C :

- $AB\bar{C}$;
- $(A + B)(B + C)$;

$$в) (A + B) + (B + C);$$

$$г) A + AB + ABC;$$

$$д) (A + \bar{A})(B + \bar{B})(C + \bar{C}).$$

10.5. Какое из следующих выражений удовлетворяет наиболее жесткому ограничению? Какое наиболее свободно от ограничений? Объясните Ваш выбор.

$$A + \bar{A}; A; A + B + C.$$

10.6. Спроектируйте логическую систему управления вентиляцией, при которой загрязненный воздух отсасывается, фильтруется и возвращается в помещение. Однако, если фильтр слишком засорен, что определяется по разности давлений на его входе и выходе, система управления включит аварийную сигнализацию, которая будет действовать до тех пор, пока не будет отключено питание. Даже если фильтр останется засоренным и включена сигнализация, вентилятор будет продолжать работать и перепускной канал фильтра будет открыт при условии, что загрязненность поступающего извне воздуха не превышает допустимого предела. Если же этот предел превышен, то вентилятор будет выключен (рис. 10.5).

10.7. Предположим, что требуется придумать устройство для мгновенного перемножения двух двухразрядных двоичных чисел A и B и получения их произведения P . Логическая система, решающая эту задачу, будет иметь четыре входа и четыре выхода:

$$\begin{array}{r} \times \quad A_1 A_2 \\ \quad B_1 B_2 \\ \hline P_1 P_2 P_3 P_4 \end{array}$$

Заметим, что произведение может содержать четыре разряда. Объясните, почему.

10.8. Составьте таблицу истинности для задачи 10.7.

10.9. Образуйте логические выражения для каждого из выходов в задаче 10.7. Методами булевой алгебры минимизируйте их.

11. ГРАФИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ

Промышленные логические системы управления, с которыми мы познакомились в предыдущей главе, удобнее изображать, проектировать и исследовать, пользуясь каким-либо из графических методов. Графическая интерпретация неоценима в тех случаях, когда логическая система становится довольно сложной — а это характерно для большинства прикладных задач автоматизации производства. Промышленные логические системы, как правило, не удастся описать только булевыми выражениями. Это затруднение объясняется тем, что в состав реальных систем обычно входят таймеры, запоминающие устройства, счетчики, элементы задержки; с этими устройствами мы познакомимся чуть позже. Данная глава имеет целью дать читателю представление об огромных возможностях логических систем при автоматизации производства. Для студентов и инженеров-практиков она полезна тем, что позволит им самостоятельно приступить к решению прикладных задач, с которыми они сталкиваются, путем создания логических систем управления. Они получают также представление о том, как спроектировать и построить простейшие промышленные роботы из типовых элементов автоматики.

11.1. ЛОГИЧЕСКИЕ СХЕМЫ

Первый графический метод естественным образом вытекает из рассмотренных в гл. 10 булевых выражений, которые интерпретируются в виде логических схем. Логическая схема составляется из элементов, являющихся аналогами операторов И, ИЛИ и НЕ. Например, на рис. 11.1 изображена логическая схема для пульта с блокировкой, рассмотренного в гл. 10. Для удобства восприятия этой схемы еще раз приведено соответствующее логическое выражение и раскрыт смысл логических переменных. Для многих не составит труда расшифровать обозначения элементов схемы на рис. 11.1, сравнив ее с логическим выражением.

На рис. 11.2 приведена расшифровка обозначений элементов логических схем. В дополнение к этим символам для изображения специальных элементов можно просто использовать квадрат с указанием внутри него функций элемента, например «Таймер» (TMR), «счетчик» и т. п. Кроме символов, принятых в данной книге, на рис. 11.2 приведены также иные обозначения, встречающиеся на практике.

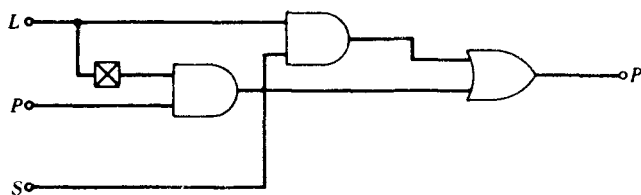


Рис. 11.1

Логическая схема для пультa с блокировкой:

$$P' = \bar{L}P + SL, \text{ где } L \text{ — состояние блокировки; } P \text{ — состояние процесса; } S \text{ — состояние переключателя}$$

Логический элемент	Алгебраический оператор	Основное обозначение элемента	Дополнительное обозначение
И	•		
ИЛИ	+		
НЕ	—		

Рис. 11.2

Обозначения элементов логических схем

Логическую схему удобнее всего представлять как некоторую электрическую цепь, входные и выходные сигналы которой суть напряжения. Проводя такую аналогию, однако, следует помнить, что эти напряжения могут принимать только два значения (уровня) в соответствии с двоичной природой логических систем.

На схемах, подобных рис. 11.1, принята условность, когда черной точкой указывается соединение линий связи, а отсутствие черной точки говорит о том, что пересекающиеся линии *не соединены* друг с другом. (Черная точка предполагает наличие пайки в месте соединения). По другой версии, все пересекающиеся линии считаются соединенными, а отсутствие соединения обозначается небольшой полукругностью (рис. 11.3). Вариант, представленный на рис. 11.1, является более предпочтительным.

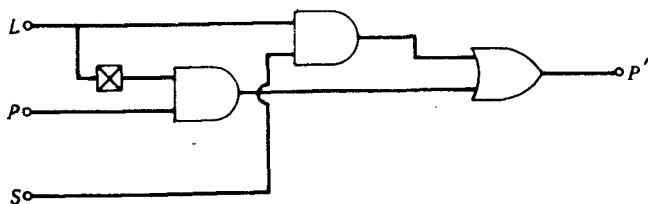


Рис. 11.3

Вариант изображения логической схемы для пультa с блокировкой

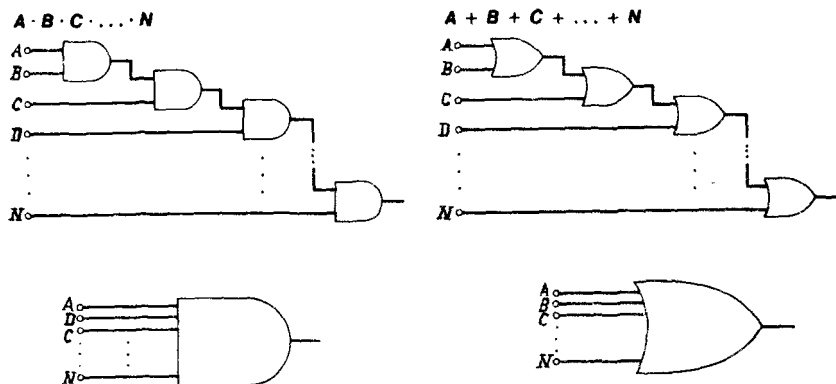


Рис. 11.4

Эквивалентная замена цепочки двухвходовых логических элементов одним n -входовым

Логический оператор И, связывающий несколько входных переменных, схематически можно изобразить с помощью цепочки двухвходовых элементов И. Однако ту же функцию можно реализовать, используя один n -входовый элемент И. В равной степени это относится и к элементам ИЛИ. Подобная эквивалентная замена показана на рис. 11.4.

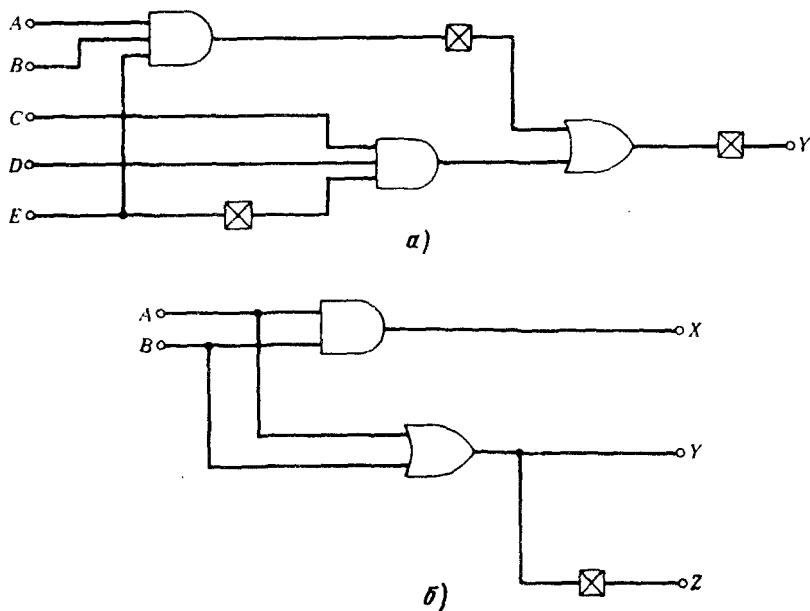


Рис. 11.5

Логические схемы, реализующие следующие булевы выражения:

$$a - Y = \overline{ABE} + \overline{CDE}; \quad б - X = AB; \quad Y = A + B; \quad Z = \overline{A + B}$$

В качестве еще одной иллюстрации использования логических схем для интерпретации булевых выражений на рис. 11.5 приведены две схемы и соответствующие им выражения. Заметим, что с помощью одной логической схемы на рис. 11.5, б реализованы несколько логических выражений.

11.2. ЛЕСТНИЧНЫЕ ЛОГИЧЕСКИЕ СХЕМЫ

Логические схемы, возможно, являются наиболее простой и естественной интерпретацией булевых логических выражений, но они не настолько хорошо знакомы специалистам-практикам. Для них более понятен язык *лестничных логических* (релейно-контактных) схем; этот язык, кстати, характерен для *программируемых контроллеров*, которые будут рассмотрены в гл. 12.

Преимущество лестничных логических схем заключается в том, что оператор И в них представляется *последовательным*, а оператор ИЛИ — *параллельным* соединением элементов (контактов). В простейшем примере на рис. 11.6 схема имеет только одну (или, может быть, две) ступень, но отдаленное сходство с лестницей все же налицо. Если читатель еще не догадался, то скажем, что эта схема соответствует уже знакомому нам примеру с пультом с блокировкой. Чтобы понять, чем отличается язык лестничных схем от языка обычных логических схем, достаточно сравнить рис. 11.6 с рис. 11.1.

Основными элементами лестничной логической схемы являются контакты (входы) и нагрузки (выходы). На рис. 11.7 представлены обозначения этих элементов и показано, как они изображаются в лестничных схемах. Левый поручень лестницы на схеме отождествляется с шиной электрической цепи, находящейся под напряжением (обычно 127 В переменного тока, но для большинства ло-

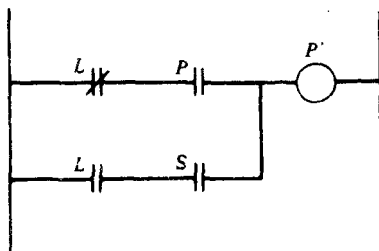
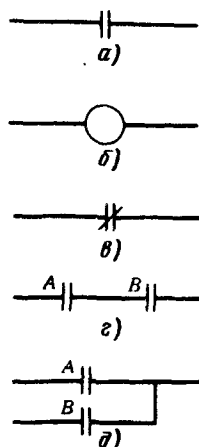


Рис. 11.6
Пример лестничной логической схемы

Рис. 11.7

Обозначения элементов лестничных логических схем:

а — входные контакты, эквивалентные действию переключателей, реле, фотоэлектрических датчиков, конечных выключателей и любых других устройств логических систем, формирующих сигнал типа включено/выключено. (Этот символ не следует путать с хорошо известным обозначением конденсаторов в электрических схемах); б — выходные нагрузки — двигатели, клапаны, устройства сигнализации, звонки, лампы, исполнительные механизмы или любая другая электрическая нагрузка, управляемая логической системой; в — логическая инверсия входа, соответствующая оператору НЕ; г — входные контакты, соединенные *последовательно*, соответствующие оператору И (реализуется выражение $A \cdot B$); д — входные контакты, соединенные *параллельно*, соответствующие оператору ИЛИ (реализуется выражение $A + B$).



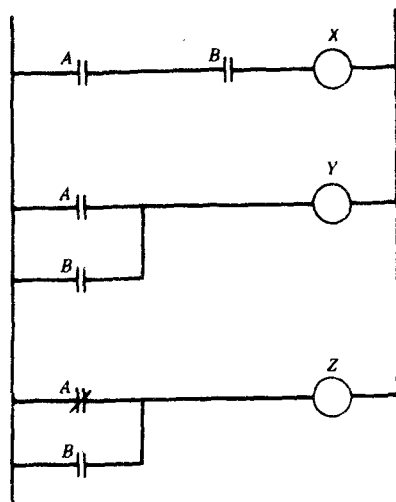


Рис. 11.8

Пример лестничной логической схемы

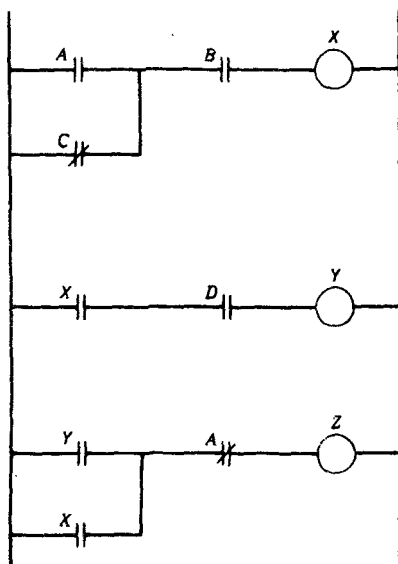


Рис. 11.9

Пример лестничной схемы, в которой состояние выходной нагрузки используется как входной контакт на другой ступеньке лестницы

гических схем это напряжение намного ниже). Правый поручень лестницы отождествляется с нейтральной шиной электрической цепи. Соответственно, контакты (переключатели) принято располагать в левой части схемы, а нагрузки — в правой части, непосредственно у нейтральной шины. Очевидно, каждая простая или составная ступенька лестницы обязательно должна содержать выходную нагрузку того или иного вида, соединенную с нейтральной шиной, иначе произойдет короткое замыкание. В сущности, ступенька без нагрузки будет действовать как прямое соединение шины питания и нейтральной шины, что приведет к срабатыванию плавкого предохранителя или тепловой защиты выключателя.

Наиболее трудным для понимания свойством лестничных схем является то, что входные контакты в них могут повторяться столько раз, сколько это необходимо. Это явление проиллюстрировано на рис. 11.8, где вход A присутствует в каждой ступеньке лестничной схемы. Эта схема интерпретирует следующие логические выражения:

$$\begin{aligned} X &= A \cdot B; \\ Y &= A + B; \\ Z &= \bar{A} + B. \end{aligned}$$

Еще одним интересным свойством лестничных схем является то, что состояние данной *выходной* нагрузки (включена/выключена)

может играть роль входной переменной логической системы, и, следовательно, оно может быть отражено в виде *входного* контакта на другой ступеньке лестничной схемы. Рис. 11.9, иллюстрирующий это свойство, приводит нас к следующим выражениям:

$$Z = (Y + X) \cdot \bar{A},$$

где $Y = X \cdot D$ и $X = (A + \bar{C}) \cdot B$.

В результате подстановки получим

$$Z = [(A + \bar{C}) \cdot B \cdot D + (A + \bar{C}) \cdot B] \cdot \bar{A}.$$

С помощью теорем булевой алгебры читатель без особого труда сможет упростить выражение для выходной переменной Z , что, в свою очередь, приведет к упрощению лестничной схемы. Предлагаем читателю в качестве упражнения решить эту задачу самостоятельно.

Важно понимать то, что во всех практических случаях логические выражения, соответствующие лестничной схеме, вычисляются *одновременно*. Порядок ступенек в лестнице не имеет значения, за исключением некоторых случаев применения программируемых контроллеров, как это будет показано в гл. 12. Однако здесь уместно игнорировать некоторые исключения и считать, что все вычисления производятся одновременно, т. е. все ступеньки лестницы как бы являются параллельными ветвями одной и той же электрической цепи.

Если теперь вернуться к самой первой рассмотренной нами лестничной схеме (рис. 11.6), то читатель уже достаточно подготовлен, чтобы узнать в ней логику пульта с блокировкой. Заметим, что для выхода здесь использовано обозначение P' в противоположность обозначению P для входа. Это сделано лишь для того, чтобы все обозначения соответствовали логической схеме на рис. 11.1, образованной с учетом правил, изложенных в гл. 10. Однако в лестничных схемах сама их символика достаточно ясно говорит о том, какие элементы являются входными и какие выходными, поэтому нет необходимости использовать «штрих» или другое обозначение переменной, чтобы отличать входы от выходов.

Принцип построения лестничных схем можно проиллюстрировать на примере с кнопочным выключателем. Мы начнем с простейшего выключателя, а затем последовательно будем добавлять элементы в лестничную схему. Рис. 11.10, а соответствует простому случаю, когда вход A включает выход Y . Как только вход A возвращается в состояние 0, выход выключается. Вряд ли нас удовлетворит такой кнопочный выключатель, потому что как только мы отпустим нажатую кнопку, выходная нагрузка окажется отключенной. На рис. 11.10, б добавлено условие ИЛИ, благодаря которому выход останется включенным, если он до этого уже был включен. Иными словами, первый вход A включает выходную нагрузку; после этого второй вход Y удерживает выход Y включен-

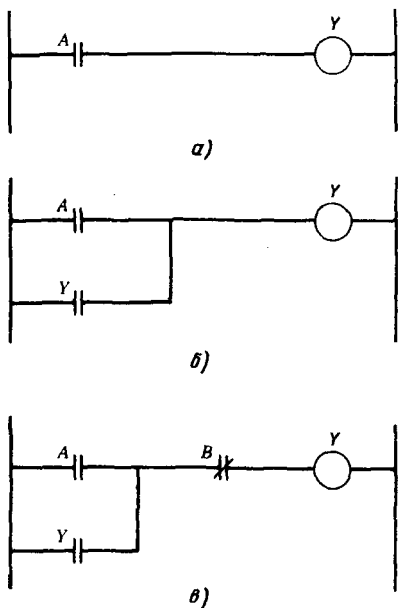


Рис. 11.10

Построение лестничной схемы для кнопочного выключателя:

а — входной контакт *A* включает нагрузку *Y*; *б* — входной контакт *A* включает нагрузку *Y*, вход *Y* удерживает ее включенной; *в* — входной контакт *A* включает нагрузку *Y*; вход *Y* удерживает нагрузку включенной до тех пор, пока вход *B* не выключит ее



Рис. 11.11

Логическая схема с устройством памяти, действующая как кнопочный выключатель

ным. Это уже лучше, но теперь возникает новая проблема: логическая схема, включив выход *Y*, уже никогда не сможет его выключить!

Как следует из рис. 11.10, б, выключить выход *Y* можно только одним способом, а именно сняв напряжение со всей системы (всей лестницы). Чтобы придать окончательный вид нашей системе с кнопочным выключателем, добавим еще один входной контакт *B* и получим схему рис. 11.10, в. Заметим, что этот входной контакт «обратного действия», т. е. выход будет находиться под напряжением, только если контакт *B* не включен. Если вход *A* отождествить с пружинной кнопкой ПУСК, а вход *B* — с пружинной кнопкой СТОП, то лестничная схема на рис. 11.10, в полностью будет соответствовать кнопочному выключателю, рассмотренному в гл. 10 (см. рис. 10.3).

Лестничную схему для кнопочного выключателя можно также назвать устройством памяти. В обычной логической схеме специальный элемент, называемый «памятью», изображается в виде квадрата, но чтобы не возникало никаких сомнений в его назначении, внутри квадрата делается надпись «ПАМЯТЬ». Существует множество модификаций элементов памяти, различающихся условиями их включения или выключения. Некоторые элементы памяти являются «сбрасываемыми», т. е. они возвращаются в состояние «выключено» всякий раз, когда система обесточивается. Другие, наоборот, являются «фиксирующими», т. е. они запоминают свое последнее состояние и возвращаются именно в него, когда вновь включается питание.

Рис. 11.11 иллюстрирует применение устройства памяти в логической схеме. Эта схема должна выполнять те же функции, что и кнопочный выключатель, иначе говоря — действовать так же,

как лестничная схема на рис. 11.10, в. Устройство памяти на рис. 11.11 можно классифицировать как элемент «с приоритетом нуля», т. е. если оба входа соответствуют логической единице, то выход будет нулем. Для большинства систем с кнопочными выключателями это является весьма желательным по соображениям безопасности. Можно создать такие устройства памяти, которые обладали бы приоритетом текущего состояния, если оба входа одновременно стремятся изменить это состояние (тем самым мы иллюстрируем множество модификаций элементов памяти). Устройство памяти на рис. 11.11 плюс ко всему должно быть сбрасываемым; в этом случае система останется выключенной, даже если питание сначала пропадет, а потом будет восстановлено. В большинстве промышленных систем при восстановлении питания гораздо безопаснее повторить процедуру запуска установки во избежание ее самопроизвольного включения.

11.3. ТАЙМЕРЫ

До сих пор, говоря о логических системах, мы предполагали, что любое изменение любой входной переменной мгновенно вызывает соответствующую реакцию на выходе. Но во многих практических случаях от логической системы требуется, чтобы выходная переменная давала реакцию на изменение входных переменных спустя некоторое заранее заданное время. Подобные случаи требуют применения таймеров или элементов задержки. Эти устройства чрезвычайно полезны в роботостроении и автоматизации производства; к сожалению, системы с таймерами намного труднее поддаются анализу и синтезу, чем обычные логические системы.

Булева алгебра не позволяет оперировать с таймерами и элементами задержки. Даже таблица истинности представляет известные трудности при попытках анализировать логическую систему с таймерами и элементами задержки. Поэтому таблицы истинности и булеву алгебру лучше всего использовать применительно к статическим логическим системам. Если же приходится иметь дело с динамическими системами, то здесь рекомендуется использовать аппарат логических схем или лестничных схем.

Символическое изображение таймера обычно представляет собой квадрат, а чтобы отличать его от счетчиков и других устройств специального назначения, внутри делается надпись «ТАЙМЕР». При этом квадрат (или прямоугольник) используется независимо от того, какая вычерчивается схема — обычная логическая или лестничная. Кроме того, в квадрате должны быть указаны период работы таймера или время выдержки. Большинство таймеров, используемых в промышленности, имеют настраиваемые параметры; пользователь устанавливает их с помощью отвертки, регулировочной ручки или другими механическими средствами. Параметры таймера устанавливаются во время настройки логической системы перед ее пуском в эксплуатацию.

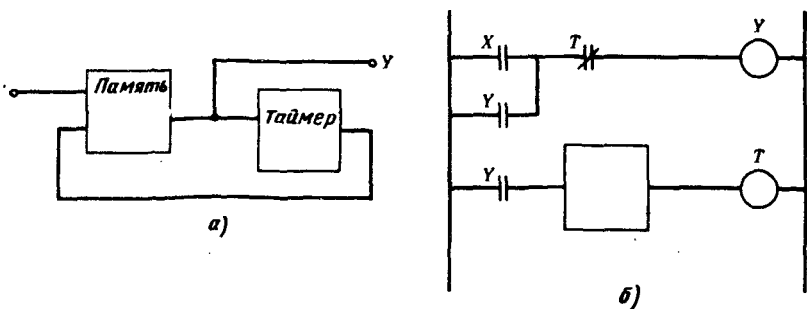


Рис. 11.12

Применение таймера в логической системе:

а — обычная логическая схема; б — лестничная логическая схема. Логика работы обеих схем одинакова

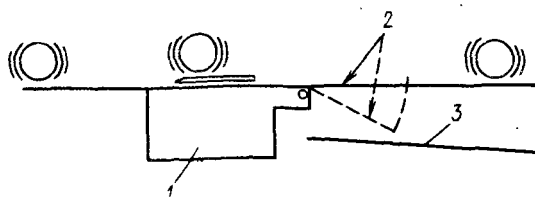


Рис. 11.13

Пункт автоматического взвешивания на конвейере:

1 — пункт автоматического взвешивания; 2 — откидная крышка; 3 — отводной лоток

Существует много модификаций промышленных таймеров, поэтому, чтобы правильно эксплуатировать каждый образец, необходимо предварительно познакомиться с описанием изделия. На рис. 11.12 показано применение типового промышленного таймера в обычной логической и в лестничной схемах, выполняющих одни и те же функции. Рисунок приведен с целью проиллюстрировать применение таймера и дать читателю еще одну возможность сравнить между собой две формы представления логических схем — обычную и лестничную. Рис. 11.12 представляет интерес не только с чисто теоретической точки зрения; обе схемы находят широкое применение во многих задачах автоматизации производства. Рассмотрим, например, изображенный на рис. 11.13 конвейер, оснащенный пунктом автоматического взвешивания. Если вес изделия, проходящего через этот пункт, превышает норму, то мгновенно открывается откидная крышка, которая остается в этом положении в течение 4 с — времени, достаточном для того, чтобы данное изделие попало на отводной лоток. Для правильной работы данной системы необходимо, чтобы изделия двигались по конвейеру друг за другом с интервалами не менее 5 с. Без сомнения, читатель может предложить и другие примеры промышленного применения логической системы, изображенной на рис. 11.12.

11.4. ВРЕМЕННЫЕ ДИАГРАММЫ

В качестве компенсации за невозможность применения таблиц истинности и булевой алгебры к системам, содержащим таймеры, предлагается еще один графический метод — *метод временных*

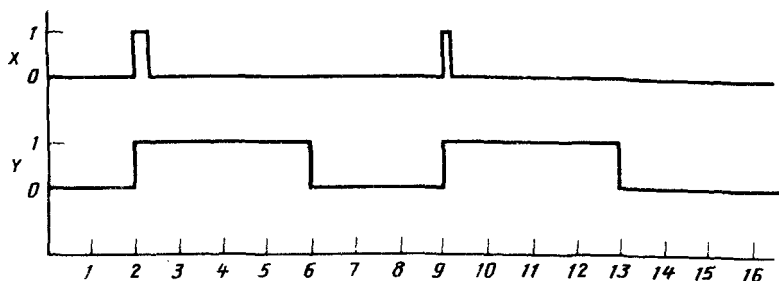


Рис. 11.14

Временная диаграмма логической системы управления конвейером с пунктом автоматического взвешивания

диаграмм. Этот метод пригоден для всех типов логических систем. С помощью временной диаграммы специалист по автоматике может показать переход любой из входных переменных из одного состояния в другое и проследить, как это отражается на выходных переменных. Временная диаграмма является также эффективным средством точного описания поведения различных устройств памяти и таймеров.

Временная диаграмма — это просто семейство графиков, каждый из которых соответствует логической переменной. На графиках по горизонтальной оси откладывается время, а по вертикальной — логическое состояние переменной, т. е. 0 или 1. Графики располагаются так, чтобы все оси времени были синхронизированы; таким образом, вертикальная линия, проведенная в любом месте диаграммы, соответствует одному и тому же моменту времени, и мы имеем возможность оценить значения всех входных и выходных переменных в данный момент.

На рис. 11.14 представлена временная диаграмма работы конвейера с пунктом автоматического взвешивания (см. рис. 11.13 и соответствующие логические схемы на рис. 11.12). Заметим, что каждая из двух логических переменных на графиках представлена только двумя состояниями: 0 и 1. В реальной системе этим состояниям могут соответствовать определенные значения напряжения, давления сжатого воздуха, высоты или какой-либо другой физической величины. Заметим также, что изменение состояния *входной* переменной X задает произвольно тот, кто изображает временную диаграмму. В данном случае в начале диаграммы выбран интервал покоя 2 с, а затем предполагается, что изделие с избыточным весом проходит через пункт взвешивания в моменты времени 2 и 9 с. График изменения *выходной* переменной Y , напротив, не является независимым, он полностью определяется структурой логической системы и, разумеется, характером изменения входной переменной.

С помощью временной диаграммы можно также описать работу обычных логических систем, не содержащих таймеров. Например,

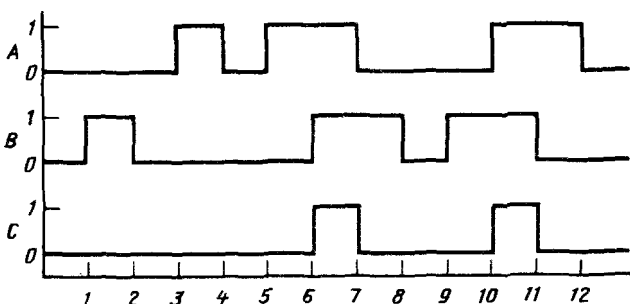


Рис. 11.15

Временная диаграмма логической системы, выполняющей операцию И

на рис. 11.15 изображена система, выполняющая логическую операцию И. Как и на всех временных диаграммах, характер изменения входных переменных задается произвольно. На рис. 11.15 графики этих переменных изображены так, чтобы продемонстрировать все их возможные принципиально отличные комбинации. График выходной переменной С при этом полностью описывает работу элемента И.

Хотя составление временной диаграммы обычно отнимает больше времени по сравнению с другими методами описания логических систем, она предоставляет неоценимую возможность изображения всех состояний системы. Это особенно важно для систем с таймерами и чрезвычайно полезно для сравнения особенностей различных типов устройств памяти, которые на первый взгляд почти идентичны. На рис. 11.16 с помощью соответствующих временных диаграмм проиллюстрирована работа трех различных типов устройств памяти.

Пример 11.1. Система автоматической вентиляции. Возможность использования таймеров значительно расширяет область применения логических систем в робототехнике и автоматизации производства. В данном примере мы покажем, как с помощью временной диаграммы проектируется простейшая промышленная автоматическая система. Многие станки (или технологические процессы) нуждаются в вентиляции во время их работы, а также в течение некоторого времени (например, 3 мин) после остановки с целью полного удаления отходов. Подобная ситуация является идеальным примером для иллюстрации методов конструирования логических систем, рассмотренных в данной главе.

Необходимость продолжения вентиляции в течение определенного времени после выключения станка требует введения в структуру логической системы таймера, а это фактически лишает проектировщика возможности использовать таблицы истинности и булеву алгебру. Поэтому не остается ничего другого, как применить графические методы, описанные в данной главе: временные диаграммы, логические схемы, лестничные схемы. Разумным первым шагом является использование временной диаграммы для описания зависимости между

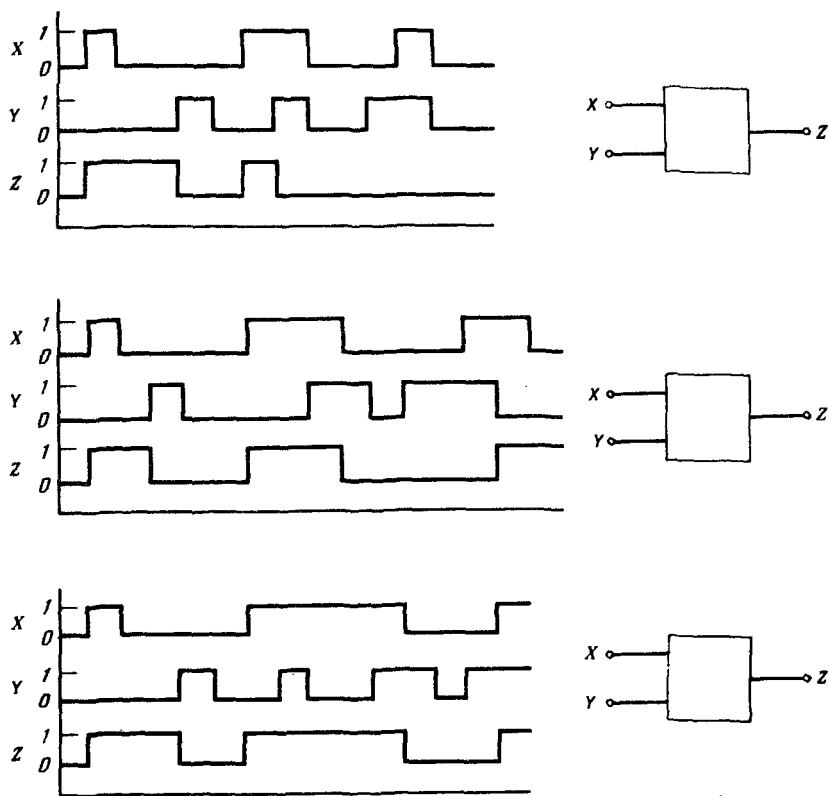


Рис. 11.16

Три типа устройств памяти на триггерах. Временные диаграммы иллюстрируют их различия:

A — память с приоритетом нуля; *B* — память с приоритетом текущего состояния; *C* — память с приоритетом единицы

входными и выходными переменными. В нашем примере система автоматической вентиляции имеет только один вход — состояние двухпозиционного выключателя системы. Этот выключатель *одновременно* запускает станок и вентилятор, и их состояние мы рассматриваем в качестве выходов системы. Однако после выключения системы вентилятор в течение 3 мин продолжает работать. Итак, определим логические переменные:

Вход

$A = 1$, если система включена,

$A = 0$, если система выключена.

Выходы

$B = 1$, если станок работает,

$B = 0$, если станок остановлен;

$D = 1$, если вентилятор включен,

$D = 0$, если вентилятор выключен.

Задав произвольным, но нетривиальным графиком для переменной *A*, изобразим временную диаграмму (рис. 11.17), чтобы отразить желаемую реакцию двух выходных переменных на различные последовательности состояний, принимаемых входной переменной. В случае входной последовательности, изображенной на рис. 11.17, предполагается, что система включается в момент

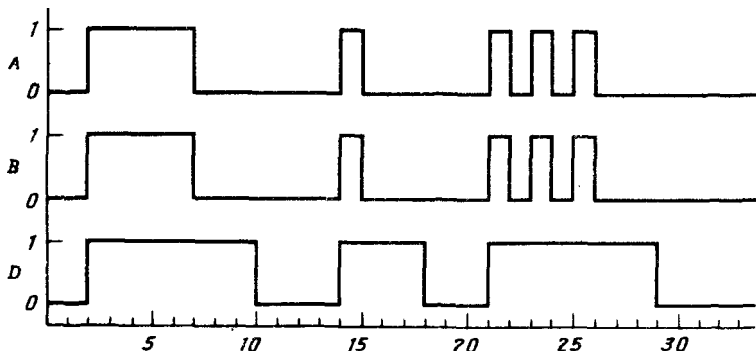


Рис. 11.17

Временная диаграмма для системы автоматической вентиляции

$t = 2$ мин. Выходная переменная B (станок) изменяется так же, как A , однако выходная переменная D (вентилятор) сохраняет значение 1 еще в течение 3 мин (до $t = 10$ мин) после выключения системы при $t = 7$ мин. С помощью короткого импульса при $t = 14$ мин показано, что продолжительность работы вентилятора после выключения системы постоянная и равна 3 мин независимо от времени пребывания переменной A в состоянии 1. С помощью последовательности коротких импульсов, начинающейся в момент $t = 21$ мин, показано, что вентилятор остается включенным в течение всего этого периода и продолжает работать до $t = 29$ мин после того, как система окончательно выключается в момент $t = 26$ мин. Таким образом, временная диаграмма необходима для того, чтобы на этапе проектирования *определить*, что же должна делать логическая система. Позже, на этапе создания и *отладки* системы, временная диаграмма может вновь потребоваться для проверки правильности работы логической системы.

После того как с помощью временной диаграммы составлено представление о поведении системы, можно перейти к синтезу логической схемы или лестничной схемы, реализующих данный алгоритм действий. Начнем с логической схемы. Не имея таблицы истинности и не пользуясь булевой алгеброй, тем не менее нам необходимо исследовать поведение входных и выходных переменных и установить логические связи между ними. Лучше всего начать с выходных переменных и двигаться в обратном направлении. Так как выходная переменная B просто должна отслеживать входную переменную A , их можно соединить напрямую. Что касается выходной переменной D (вентилятор), то ее состояние определяется *двумя альтернативами*: либо входом A , либо еще включенным таймером после того, как A сбрасывается на нуль. Слово «альтернативы» дает проектировщику ключ к использованию логического элемента ИЛИ. Таким образом появляется фрагмент логической схемы, который показан на рис. 11.18. Обратим внимание, что единственный вход изображен в левой части схемы, а два выхода — в правой.

Продолжая рассуждения, попытаемся задействовать входы элемента ИЛИ. С первым входом все ясно: одна из альтернатив иметь вентилятор включенным — это просто включить систему (входная переменная A). На другой вход должен прийти сигнал от таймера, но заметим, что по условию задачи вентилятор должен оставаться включенным, пока время, задаваемое таймером, еще не истекло. Это приводит нас к идее использовать элемент НЕ между выходом таймера и входом элемента ИЛИ. В результате этих рассуждений логическая схема постепенно усложняется и на данном этапе принимает вид, изображенный на рис. 11.19.



Рис. 11.18

Фрагмент логической схемы системы вентиляции

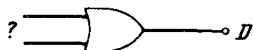
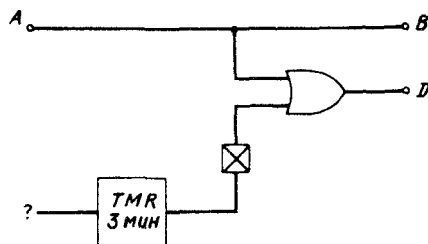


Рис. 11.19

Второй этап построения логической схемы для системы автоматической вентиляции



Остается решить только один вопрос: какой логический вход использовать для запуска таймера? Будем считать, что в качестве таймера выбрано устройство, принцип действия которого иллюстрируется временной диаграммой на рис. 11.20. Заметим, что таймер не является кумулятивным; каждый раз, когда вход X возвращается на уровень логического нуля, показания таймера сбрасываются и он должен начинать отсчет 3 мин снова с нуля. Так, если вход X периодически принимает значение 0 с интервалами менее 3 мин, то выход Y никогда не будет равен единице. Нам же нужно совсем обратное. Если система периодически выключается на время, меньшее чем 3 мин, вентиляция *не должна* прекращаться. Именно поэтому мы и поставили элемент ИЛИ на входе таймера. Элемент НЕ на выходе таймера нам нужен потому, что мы хотим, чтобы отсчет интервала 3 мин начинался только тогда, когда система выключается. Окончательный вид логической схемы представлен на рис. 11.21.

Для системы автоматической вентиляции можно разработать также лестничную логическую схему. Это можно сделать либо путем анализа логических связей между переменными (т. е. как и ранее при построении обычной логической схемы), либо путем поэтапного преобразования логической схемы. Мы воспользуемся вторым методом, предыдущая же схема будет служить нам для проверки. Лестничная схема должна содержать по крайней мере две ступеньки, соответствующие двум выходам: B (станок) и D (вентилятор). Входной выключатель A по необходимости будет повторяться в схеме, чтобы каждая ступенька правильно отражала логические связи. Мы введем также третью ступеньку, учитывающую логику работы таймера. Окончательный вид лестничной схемы представлен на рис. 11.22; читателю предоставляется возможность самостоятельно разобраться в работе данной схемы и убедиться, что по выполняемым функциям она полностью идентична логической схеме на рис. 11.21.

В завершение данного примера сделаем несколько замечаний. Существует большая свобода в выборе конфигурации как обычной логической схемы, так и лестничной схемы. Следует помнить, что

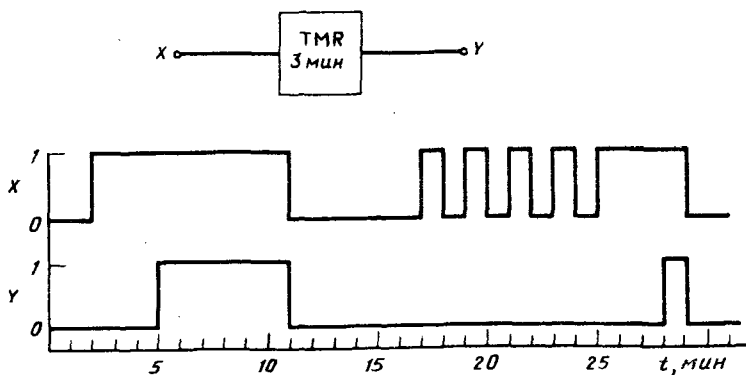


Рис. 11.20

Простой таймер с пояснением его принципа действия

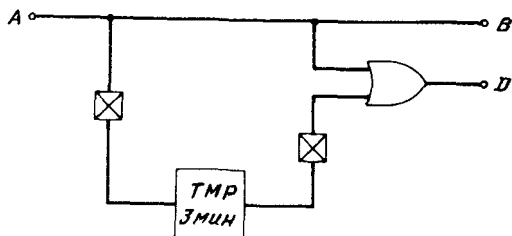


Рис. 11.21
Окончательный вид логической схемы для системы автоматической вентиляции

можно получить множество правильных решений. Что касается последовательности сту-

С целью закрепления графических методов построения логических систем рассмотрим еще один, несколько более сложный пример.

Пример 11.2. Управление печью. На рис. 10.1 показана небольшая печь для предварительного разогрева кузнечных заготовок перед их подачей под штамповочный молот. В печь подается природный газ, и если пламя гаснет, то, разумеется, горелку надо вновь зажечь. В противном случае, если пламя гаснет, а клапан остается открытым, ситуация представляет потенциальную опасность. До появления автоматических систем рабочий должен был наблюдать за пламенем и в случае его исчезновения либо вновь зажигать горелку, либо перекрывая подачу газа. Современные системы автоматически контролируют состояние пламени и выполняют те же действия, но без участия человека. Конечно, надо располагать какими-то средствами, с помощью которых при необходимости можно было бы отключить автоматическую систему. Ясно, что различные данные о состоянии такой системы и предпринимаемые действия имеют двоячную природу; поэтому вполне уместно поставить вопрос о создании промышленной логической системы управления. Читатель, вероятно, легко будет заметить, что в данной

системе должен присутствовать хотя бы один таймер, чтобы осуществлять зажигание при включении системы в работу и выключать систему ради безопасности, если пламя не зажигается или не восстанавливается. Охарактеризуем логическую систему управления такой печью с помощью временной диаграммы, а также изобразим соответствующую логическую схему и лестничную схему.

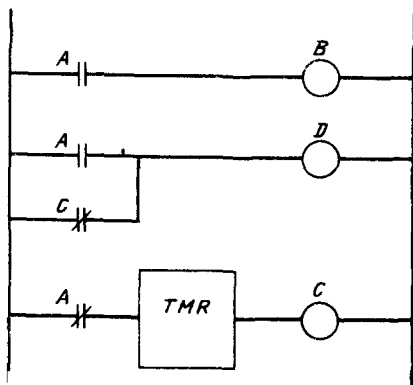


Рис. 11.22
Лестничная логическая схема для системы автоматической вентиляции

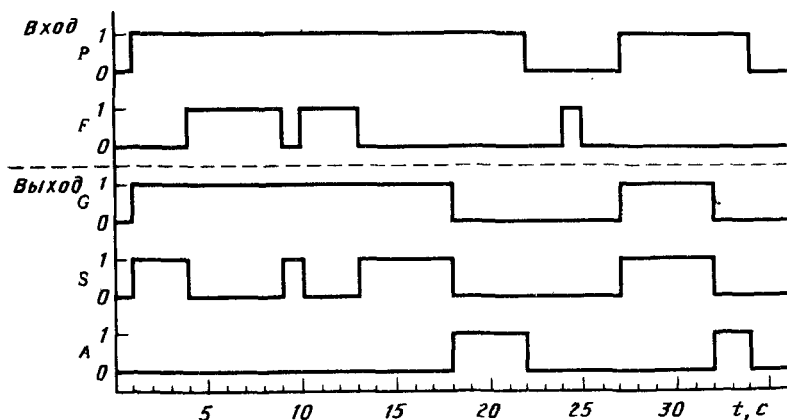


Рис. 11.23

Временная диаграмма для логической системы управления печью

Определим сначала необходимые нам входные и выходные переменные.

ВХОД

Электропитание: $P = 1$, если питание включено;
 $P = 0$, если питание отключено.

Пламя: $F = 1$, если пламя существует,
 $F = 0$, если пламя отсутствует.

ВЫХОД

Топливо: $G = 1$, если вентиль на трубопроводе открыт,
 $G = 0$, если вентиль закрыт.

Зажигание: $S = 1$, если система искрового зажигания включена,
 $S = 0$, если система искрового зажигания выключена.

Сигнальное устройство: $A = 1$, если устройство мигает,
 $A = 0$, если устройство не дает сигнала об опасности.

Можно попытаться учесть многие другие входные и выходные переменные, такие, как «ручная блокировка автоматики», «давление газа», но это значительно усложнит автоматическую систему. Поэтому мы рассмотрим простую логическую систему, ограничившись пятью наиболее важными переменными, указанными выше.

При более внимательном рассмотрении пяти входных и выходных переменных может возникнуть вопрос: почему «зажигание» считается выходной, а «пламя» — входной переменной, а не наоборот? В самом деле, разве не зажигание вызывает появление пламени? Но давайте не забывать о том, что мы проектируем логическую систему управления и слова «вход» и «выход» следует понимать как относящиеся именно к этой системе, а не к объекту управления, которым является печь. Зажигание является автоматическим ответом (выходом) логической системы управления на состояние входа (отсутствии пламени). Наличие пламени может быть установлено с помощью фотоэлектрического датчика или датчика температуры. Топливо также является выходной переменной, так как логическая система определяет, следует ли открыть или закрыть вентиль на трубопроводе, по которому подается газ. Электропитание является входной переменной, потому что оно необходимо для всех выходных переменных.

Таймер нужен для того, чтобы даже при отсутствии пламени в течение определенного времени продолжалась подача топлива и можно было бы осуществить

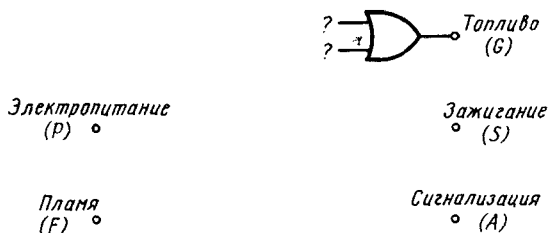


Рис. 11.24

Первый этап построения логической схемы для системы управления печью; здесь отражены две возможности, при которых осуществляется подача топлива

зажигание, но если пламя не появляется, скажем, через 5 с, то подачу топлива надо прекратить и включить аварийную сигнализацию.

Как и в примере 11.1, по-видимому, лучше всего начать с составления временной диаграммы, как дающей наиболее наглядное представление о желаемом поведении логической системы управления печью. Один из возможных вариантов такой диаграммы изображен на рис. 11.23. Рассмотрим последовательно все моменты времени, в которые происходят существенные изменения состояния системы. В момент $t = 1$ с включается электропитание и одновременно начинается подача топлива и работа системы зажигания. Спустя 3 с, при $t = 4$ с, появляется пламя, что является поводом для выключения зажигания; подача топлива при этом не прекращается. Система работает нормально до $t = 9$ с, и в этот момент мы имитируем кратковременное исчезновение пламени, чтобы определить соответствующую реакцию со стороны системы зажигания. При $t = 9$ с возобновляется зажигание, при $t = 10$ с восстанавливается пламя, поэтому зажигание выключается. В момент $t = 13$ с пламя исчезает на продолжительный период. Мгновенно начинает работать система зажигания, но пламя не восстанавливается. Наконец, в момент $t = 18$ с инициируются действия, соответствующие аварийной ситуации, а именно: прекращается подача топлива, выключается система зажигания и начинает работать сигнализация. Если электропитание включено, то система будет оставаться в таком состоянии неопределенно долго. В момент $t = 22$ с мы отражаем на диаграмме действие оператора, отключающего электропитание. Это приводит к выключению устройства сигнализации. В итоге система полностью выводится из рабочего режима и будет оставаться в таком состоянии до тех пор, пока вновь не появится электропитание. В момент $t = 24$ с мы имитируем появление ложного одисекундного сигнала от датчика пламени. Теоретически этого не должно произойти, так как подача топлива перекрыта, но нам необходимо убедиться в том, что логическая система способна игнорировать неожиданные помехи (такие, как ложные сигналы от датчика или вспышки света от посторонних источников). Соответственно в системе ничего и не происходит. В момент $t = 27$ с мы вновь включаем электропитание и возвращаемся к тому же состоянию системы, с которого начали анализ при $t = 1$ с. Однако на этот раз пламя не зажигается, и диаграмма показывает нам, что процедура аварийной остановки при пуске системы протекает так же, как и при случайном исчезновении пламени в процессе нормальной работы. В момент $t = 34$ с происходит то же, что и при $t = 22$ с. Таким образом, временная диаграмма демонстрирует нам все возможные последовательности событий. Она позволяет нам четко установить желаемое поведение логической системы при различных обстоятельствах. Позже, на этапе отладки созданной логической системы управления, временную диаграмму можно использовать для формирования тестовой процедуры, чтобы посмотреть, работает ли система так, как это предписано временной диаграммой.

Следующий этап состоит в построении логической схемы или лестничной схемы, реализующих логику действий, определяемую временной диаграммой. Это может оказаться непростой задачей, если не прибегать к помощи таблиц истинности. Напомним, что наличие таймера в системе управления печью делает затруднительным или даже вовсе невозможным использование таблицы истинности. Как и в примере 11.1, попробуем сначала построить логическую схему, причем начнем опять-таки с выходов и будем двигаться в обратном направлении.

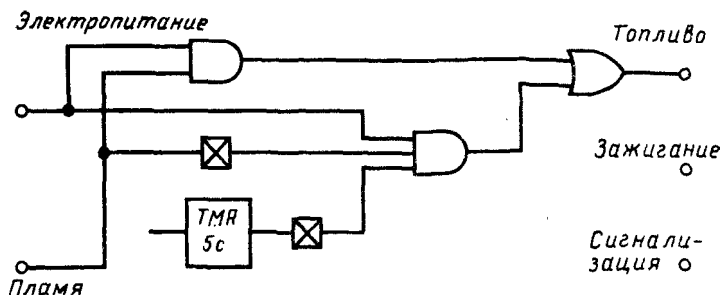


Рис. 11.25

Второй этап построения логической схемы для системы управления печью

Система управления печью имеет три выхода: топливо G , зажигание S и сигнализацию A . Существуют две возможные ситуации, при которых должна осуществляться подача топлива:

1. Электропитание включено, пламя горит и система работает нормально.
2. Электропитание включено, зажигание пытается восстановить пламя, но безопасный интервал времени (5 с) еще не истек. Эти две возможности предполагают использование логического элемента ИЛИ, выходом которого должна являться переменная «топливо». Соответствующий фрагмент логической схемы представлен на рис. 11.24.

Продолжая построение логической схемы, мы видим, что одним из условий подачи топлива является то, что одновременно должны быть в наличии электропитание и пламя. Слово «одновременно» предполагает использование логического элемента И, к которому подводятся входные переменные P и F . Другая альтернатива осуществления подачи топлива связана с выполнением трех условий:

1. Электропитание включено.
2. Зажигание пытается восстановить пламя (но пламени еще нет).
3. Безопасный лимит времени еще не исчерпан.

Одновременное выполнение этих условий предполагает использование в схеме еще одного элемента И, на этот раз трехходового, к которому подводятся входная переменная P , инвертированная (пламени *нет*) переменная F и сигнал с выхода 5-секундного таймера. На этом этапе логическая схема принимает вид, изображенный на рис. 11.25.

Продолжая далее наше движение по схеме, задействуем вход таймера. Будем считать, что выбран таймер того же типа, что и в примере на построение системы автоматической вентиляции, только на этот раз время выдержки вместо 3 мин составляет 5 с. Вспомнив принцип действия такого таймера, отметим, что если его входной сигнал периодически сбрасывается на нуль с интервалами менее 5 с, то выходной сигнал никогда не сможет принять значение логической единицы. Это именно то, что нам нужно в данном примере. Если пламя всегда появляется

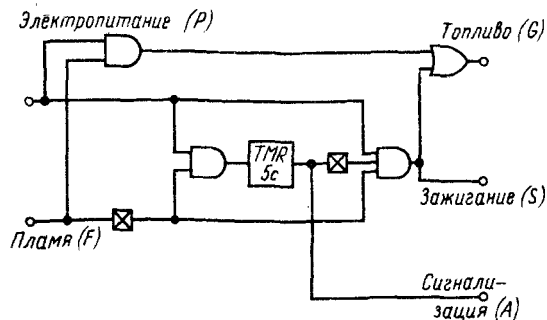


Рис. 11.26

Окончательный вид логической схемы для системы управления печью

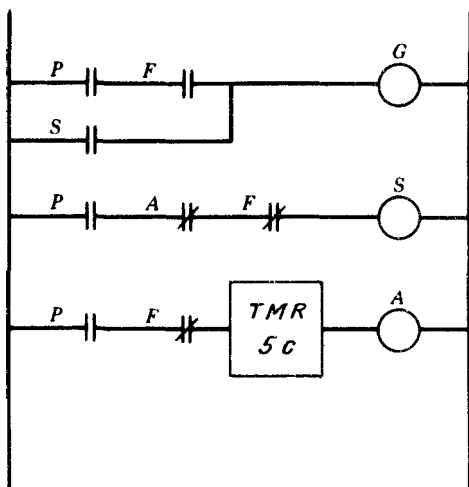


Рис. 11.27
Лестничная логическая схема для системы управления печью

раньше, чем истекнут 5 с, то нет необходимости включать аварийную сигнализацию. Должен быть выдержан определенный промежуток времени, в течение которого электропитание остается включенным, а пламя отсутствует. Это предполагает использование двухвходового элемента И, на который подаются сигналы P и инвертированный (НЕ) F .

Чтобы завершить построение логической схемы, нам нужно дополнить ее связями, ведущими к выходам S и A . Это сделать намного проще, чем для выхода G , уже учтены в схеме, и нет необходимости их повторять. Например, выход S (зажигание) должен быть идентичен второму входу элемента ИЛИ, формирующего переменную G (топливо). Выход A (сигнализация) есть не что иное, как выход таймера. Окончательный вид логической схемы показан на рис. 11.26. Эту схему можно реализовать с помощью стандартных электрических логических элементов, а затем протестировать ее на предмет соответствия временной диаграмме на рис. 11.23.

В завершение построим лестничную схему для системы управления печью. В схеме нам потребуются по крайней мере три ступеньки, соответствующие трем выходам: G (топливо), S (зажигание) и A (сигнализация). Входы P (электропитание) и F (пламя) будут представлены в схеме контактами и будут повторяться везде, где это необходимо. Полная лестничная схема изображена на рис. 11.27.

Мы рассмотрели пример на построение системы управления печью для предварительного разогрева кузнечных заготовок. Оглядываясь назад, можно было бы усомниться в ценности этого примера, как поверхностного, не учитывающего такие реальные факторы, как продувки перед повторными запусками, случайные пульсации входных сигналов и неисправности клапанов или исполнительных механизмов. Но ведь прежде чем победить, мы сначала идем шагом, поэтому и данный упрощенный пример был выбран для того, чтобы, игнорируя многие реальные факторы, проиллюстрировать лишь принципы построения логических систем управления.

В гл. 12 будет показано, с помощью каких серийных технических средств воплощается в жизнь идея автоматизации производства на основе логических систем управления. Чтобы заставить реальную систему работать, от проектировщика сейчас не требуется быть магом и волшебником в области электроники или электротехники. Все, что он должен сделать, — это отчетливо представить себе функции автоматической системы, отразить логику ее работы в виде обычной логической схемы или лестничной схемы, а затем запрограммировать некоторое техническое устройство,

реализующее эту логику. В гл. 10 и 11 были изложены основы аналитических методов проектирования, а гл. 12 будет посвящена реализации этих методов с помощью современных технических средств.

Выводы

В данной главе показано, с помощью каких изобразительных средств, дополняющих словесное описание, таблицы истинности и логические выражения, можно охарактеризовать работу логических систем управления. Некоторые промышленные логические системы, особенно системы, содержащие таймеры и элементы задержки, не поддаются описанию с помощью логических высказываний или выражений; здесь необходима уже та или иная графическая интерпретация.

Временная диаграмма является прекрасным методом описания того, что должна делать логическая система, однако она не дает ответа на вопрос о том, как эта система должна реализовывать свои функции. Поэтому временная диаграмма полезна на этапе формирования представлений о логике работы системы, а в дальнейшем она может быть использована в качестве эталона при тестовых испытаниях.

Логические схемы и лестничные схемы суть два метода воплощения того, как будет реализовываться логика действий автоматической системы. Обе они наводят на мысль об использовании электрических цепей, однако следует помнить, что это лишь один из нескольких возможных способов реализации логических действий.

Графические методы описания промышленных логических систем управления были проиллюстрированы двумя простыми примерами. Хотя эти примеры были несколько идеализированы, и реальные системы всегда намного сложнее, известные упрощения позволили сконцентрировать внимание на основных технических приемах изображения и анализа логических систем управления.

Упражнения и вопросы

11.1. Какие недостатки таблиц истинности и булевой алгебры заставляют прибегать к использованию логических схем и лестничных схем?

11.2. Какова разница между *таймером* и *элементом задержки*? Являются ли они взаимозаменяемыми?

11.3. На рис. 11.28 изображен двухвходовый кумулятивный таймер, у которого один вход является разрешающим, а второй действует как обычный вход таймера. Таким образом, вход таймера можно заблокировать, при этом таймер запомнит время, прошедшее с момента его включения, и будет хранить эти данные. Каждый раз, когда сигнал на разрешающем входе сбрасывается на нуль, показания таймера обнуляются, и он вновь готов к запуску,

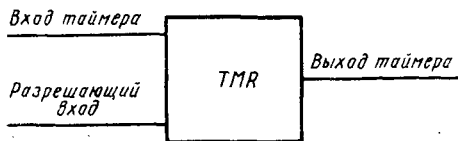


Рис. 11.28

Двухвходовый кумулятивный таймер

как только разрешающий сигнал становится равным единице. Изобразите временную диаграмму, иллюстрирующую принцип действия такого таймера.

11.4. Объясните, с помощью каких соединений двухходовый таймер (см. рис. 11.28) можно превратить в обычный таймер, действующий в соответствии с рис. 11.20.

11.5. Перерисуйте лестничную схему (см. рис. 11.27) для системы управления печью, используя вместо обычного таймера двухходовый кумулятивный таймер.

11.6. Перерисуйте лестничную схему для системы автоматической вентиляции (см. рис. 11.22), включив в нее вместо обычного таймера двухходовый таймер.

11.7. Разработайте логическую схему для автоматической системы управления вентиляцией, описанной в упражнении 10.6.

11.8. Изобразите лестничную логическую схему для системы из упражнения 10.6.

11.9. Изобразите логическую схему для устройства автоматического перемножения двоичных чисел из упражнения 10.7.

11.10. Разработайте лестничную схему для логической системы, выполняющей те же действия, что и устройство автоматического перемножения двоичных чисел из упражнения 10.7.

11.11. При изготовлении светокопий применяется аммиак, который после остановки копировальной машины должен быть удален из нее во избежание загрязнений окружающей атмосферы. В такой установке в обязанности оператора входит включение вытяжного вентилятора на 2 мин, что необходимо для полного удаления аммиака. Во время нормальной работы установки вентилятор *не должен* работать, так как аммиак нужен в процессе копирования. Как бы Вы автоматизировали такую систему? Чем эта система отличается от рассмотренной ранее в данной главе системы автоматической вентиляции. Продемонстрируйте это при помощи временной диаграммы.

11.12. Изобразите логическую схему для автоматической системы, предложенной Вами для светокопировальной установки из упражнения 11.11.

11.13. Разработайте лестничную схему для автоматической системы, обслуживающей светокопировальную установку из упражнения 11.11.

11.14. Сравните логическую схему на рис. 11.5 с лестничной схемой на рис. 11.8. Обратите внимание, что реализуемая ими логика не совсем идентична. Видоизмените схему на рис. 11.5 так, чтобы ее логика соответствовала схеме на рис. 11.8. Видоизмените схему на рис. 11.8 так, чтобы ее логика соответствовала схеме на рис. 11.5.

11.15. Робот может быть оснащен блокировкой, которая отключает электропитание, если дверь для доступа в рабочую зону открыта. Спроектируйте логическую систему, которая не только отключала бы электропитание робота, но и приводила в действие звуковую сигнализацию, если открыта любая из пяти дверей ограждения. Звуковая сигнализация должна действовать в течение 30 с, но электропитание робота должно оставаться включенным до тех пор, пока не будут закрыты все двери ограждения и нажата кнопка запуска системы.

12. ПРОГРАММИРУЕМЫЕ КОНТРОЛЛЕРЫ

Настоящая глава является заключительной по теме «Системы управления промышленным оборудованием». Эта тема нашла отражение в гл. 10—12. В гл. 10 дан анализ логических систем управления промышленным оборудованием. В гл. 11 рассмотрены способы программирования. В настоящей главе раскрыта процедура реализации систем управления с использованием одного из наиболее простых устройств, предназначенных для ускорения автоматизации производства. Эти устройства отличаются очень высокой гибкостью и могут применяться для автоматизации практически любого производственного процесса. Такие устройства, называемые программируемыми контроллерами (ПК), обеспечивают контроль безопасности работы оборудования и потребления электроэнергии, а также управляют станками и автоматическими производственными линиями.

Представляет интерес взаимосвязь между ПК и промышленными роботами. Относительно несложно осуществлять с помощью индикации контроль за работой системы управления, которая выдает управляющие сигналы на пуск робота. Формирование таких сигналов осуществляется на основе данных, поступающих на входы системы от соответствующих датчиков, контролирующих технологический процесс. Возможен визуальный контроль по индикации выходного сигнала от робота к ПК. Этот сигнал является командой на пуск станка. Однако, кроме приведенных примеров, обычные ПК находят широкое применение в качестве устройств управления роботами. Отдельные элементы движения рабочих органов робота, например, ВЫЙТИ В ПОЗИЦИЮ — ЗАХВАТИТЬ — ВЕРНУТЬСЯ В ИСХОДНУЮ ПОЗИЦИЮ — ПОВЕРНУТЬ — ВЫЙТИ В ПОЗИЦИЮ — ОСВОБОДИТЬ — ВЕРНУТЬСЯ В ИСХОДНУЮ ПОЗИЦИЮ и т. п., находятся под непосредственным контролем универсальных ПК, которые имеются в продаже. Таким образом, представляется возможность создания промышленного робота (особенно с ограниченным числом степеней подвижности) с использованием пневмоцилиндров, двигателей, приводов и конечных выключателей, а также ПК в качестве системы управления.

12.1. ЧТО ПРЕДСТАВЛЯЕТ СОБОЙ ПК?

Аббревиатура ПК может означать несколько понятий даже для людей, занятых в одной и той же области деятельности, не говоря уже о ее значениях в других областях. В начале 70-х гг. был разработан программируемый калькулятор — малогабаритное вычислительное устройство, которое могло запоминать некоторую последовательность арифметических команд. В начале 80-х гг. фирмой ИБМ стал широко применяться термин *персональный компьютер*, который относится к группе малогабаритных ЭВМ, отличающихся невысокой стоимостью; большинство пользователей и производителей стали называть такие недорогие ЭВМ *домашними компьютерами*. Тем не менее ни персональный компьютер, ни программируемый калькулятор не являются типом ПК, целиком предназначенным для управления роботами и автоматиза-

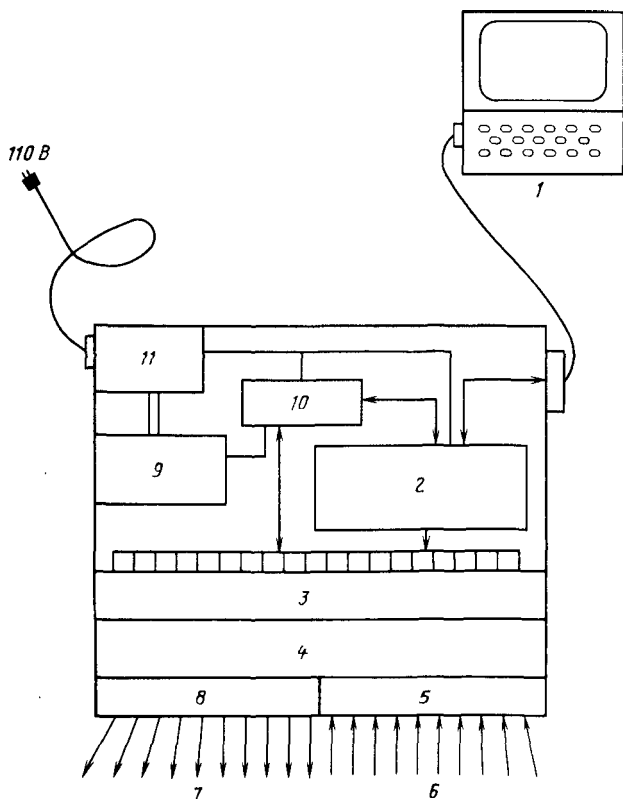


Рис. 12.1
Программируемый контроллер:

1 — переносной программатор; 2 — центральный процессор; 3 — интерфейс; 4 — контакты входов-выходов; 5 — входы; 6 — от процессора; 7 — к технологическому оборудованию; 8 — выходы; 9 — аварийное батарейное питание; 10 — память; 11 — источник питания

ции производства. Именно ПК предоставил широкие возможности для автоматизации заводского оборудования, особенно при реализации систем управления промышленным оборудованием и автоматическими линиями.

На рис. 12.1 в виде схемы представлены основные элементы типичного ПК. Питание ПК осуществляется от обычной сети, хотя ПК может обеспечить питание схем управления большой мощности при напряжении 440 В и выше. Совместно с ПК применяется съемный модуль-программатор. Один и тот же программатор может быть переставлен с одного ПК на другой. После подключения к одному из ПК программатор может быть подключен к другим ПК или отправлен на хранение. Интерфейсные устройства и устройства ввода-вывода данных, входящие в состав ПК, как показано на рис. 12.1, могут представлять собой дополнительные съемные модули.

Сравнение ПК с ЭВМ. Компоненты блока центрального процессора ПК, такие, как память и ЦП (центральный процессор), позволяют предположить, что ПК представляет собой разновидность ЭВМ. Действительно, и программируемый калькулятор, и персональный компьютер, и программируемый контроллер включают в качестве базового элемента ЭВМ, построенную на микропроцессорной основе. Все же лучше не рассматривать ПК в качестве ЭВМ. Почему? ЭВМ с высокой скоростью выполняет последовательность операций. Отдельные команды в этой последовательности выполняются за микросекунды и даже за доли микросекунд. Однако для выполнения полной программы, состоящей из последовательности таких команд, могут потребоваться секунды, минуты или даже часы. При этом время выполнения зависит от сложности программы. Однако подобное функционирование совершенно не подходит к типовому ПК. На выполнение полной программы ПК тратятся доли секунды, а отдельные шаги программы отрабатываются не быстрее, чем в обычных ЭВМ. Поэтому программа должна быть короче и проще. Сокращение и упрощение программы не вызывает проблем, поскольку основной задачей ПК является принятие логических решений типа да-нет, а не манипуляция большой последовательностью сложных арифметических вычислений над огромным количеством данных. Исполнение всей последовательности команд ПК происходит столь быстро, что обычно процесс такой реализации рассматривается как одномоментный. Таким образом, ПК непрерывно и практически без задержки анализирует данные с датчиков, контролирующих технологический процесс, формирует логические решения и передает выходные данные в виде управляющих сигналов на органы управления тем же процессом.

Другими различиями между обычными ЭВМ и ПК являются управление и представление исполнительных программ. В обычной ЭВМ загрузка программы производится по команде, затем осуществляется исполнение этой программы 1—2 раза, после чего

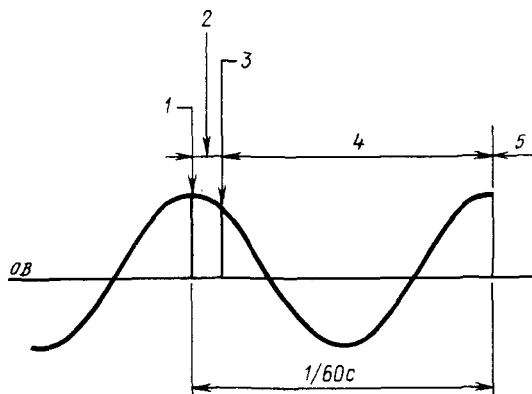


Рис. 12.2

Пример цикла обработки данных ПК. Время обработки логических данных весьма мало по сравнению с полным периодом переменного тока:

1 — опрос входов при максимальном напряжении; 2 — обработка всех логических данных; 3 — установка в соответствующее состояние всех выходов; 4 — время ожидания (до начала следующего цикла); 5 — следующий цикл

загружается следующая программа, производится ее исполнение и т. д. Значительная часть машинного времени используется на компилирование новых программ, введенных в ЭВМ различными пользователями. Ежедневно ЭВМ осуществляет реализацию различных комплектов программ в соответствии с запросами пользователей. Таким образом, обычная ЭВМ представляет собой универсальное устройство. Наоборот, ПК представляет собой специализированное устройство. Как правило, он может исполнять свою микропрограмму непрерывно *сотни миллионов* раз до прерывания, после которого вводится новая программа. ПК может перепрограммироваться ежечасно, ежедневно, в зависимости от необходимости переналадки. Однако ПК может быть запрограммирован всего лишь один раз на весь срок службы. Это отличие ПК от ЭВМ по режиму работы достаточно четко видно из рис. 12.1. Не предусмотрено подключение к ПК печатающих устройств, электрических пишущих машинок, устройств считывания с перфоленты, дисководов с гибкими магнитными дисками, дисплеев или пультов с универсальной клавиатурой. Даже программатор подключается к ПК нестационарно, но он может оснащаться клавиатурой или дисплеем. Однако эти устройства при работе ПК не используются. Далее будет показано, что ПК может оснащаться указанными периферийными устройствами при сборе данных и для передачи отчетов. Однако оснащение ПК пультом с универсальной клавиатурой и дисплеем позволяет получить устройство, занимающее промежуточное положение между ЭВМ и ПК.

Цикл работы ПК. Исполнение последовательности команд программы в ПК осуществляется непрерывно с высоким быстродействием. Начало исполнения следующих команд синхронизировано с периодом переменного тока сети питания ПК. Начало каждого цикла синхронизировано с максимумом напряжения питания, как это можно видеть из рис. 12.2. Длительность исполнения логической части программы мала по сравнению с длительностью всего цикла. Длительность исполнения программы (за-

штрихованная область на рис. 12.2) представлена не в масштабе, поскольку истинная длительность будет изменяться в зависимости от количества ступеней логической схемы, числа выходов и входов; она определяется уровнем сложности автоматизированного объекта. Длительность всего цикла (длительность периода переменного тока) остается постоянной и определяется частотой сети питания, которая в США составляет 60 с^{-1} , или 60 Гц. В некоторых моделях ПК длительность исполнительного цикла может определяться не частотой сети питания, а частотой другого источника. Однако суть заключается в том, что частота исполнительного цикла ПК обычно остается неизменной.

Если новая информация от датчика, контролирующего технологический процесс, поступит в ПК в течение исполнительного цикла, возможно искажение исходной информации, которая окажется несовместимой с запрограммированной логикой. Если программируемая логическая схема состоит больше чем из одной ступени, например содержит тот же вход, и некоторые из ступеней выполняются до изменения состояния входов, а некоторые после такого изменения, то логические выходы окажутся несовместимыми. Однако в действительности здесь, как правило, проблемы не возникает. В ПК существует возможность коррекции логической информации через $1/60\text{ с}$ в следующем цикле с помощью программируемой логики. В большинстве применяемых в производстве ПК никогда не возникает заметных отклонений во время реализации программы свыше миллионов циклов. Тем не менее можно разработать логическую схему, в которой ступени окажутся в сильной взаимосвязи. В результате разработанная система управления промышленным оборудованием может совершать очевидные логические ошибки, даже ошибки, приводящие к аварийному состоянию. Таких ситуаций можно избежать или их отрицательный эффект может быть минимизирован путем исключения соответствующих взаимосвязей и путем тщательной подготовки управляющей программы для ПК. Концепции программирования ПК, изложенные далее, помогут прояснить эту проблему.

Управление с обратной связью и управление без обратной связи. Существуют два способа управления автоматизированным технологическим процессом с помощью ПК. Способ, основанный на управлении с обратной связью, в котором предусматривается контроль с помощью датчиков некоторых параметров технологического процесса. От датчика выдается сигнал об окончании выполнения заданной команды управления автоматизированным процессом. Это является началом исполнения следующей команды. Второй способ предусматривает выполнение последовательности команд без обратной связи. При этом используются таймеры, входящие в состав ПК. Таймером определяются окончание действия одной команды и ввод в действие следующей в соответствии с запрограммированной последовательностью.

Принципиальное различие между двумя способами заключается в том, что первый способ управления предполагает использование обратной связи, в то время как второй (концепция второго типа управления) предусматривает управление без обратной связи. Управление с обратной связью имеет преимущество, заключающееся в обеспечении действительного завершения исполнения каждой команды, продолжительность исполнения которой можно варьировать. Другими словами, для исполнения соответствующей команды должно быть зарезервировано максимальное время, достаточное для исполнения этой команды. При управлении без обратной связи неизбежны затраты времени на холостой ход ПК, если фактическая продолжительность исполнения команды менее ожидаемой.

Вследствие очевидных преимуществ управления с обратной связью может возникнуть соображение, что управление без обратной связи не следует использовать. Однако управление без обратной связи имеет преимущество, заключающееся в простоте, и более существенное преимущество, состоящее в отсутствии дорогостоящих датчиков для контроля исполнения каждой команды. Некоторые ПК включают специальные устройства, например таймеры, моделирующие многоконтурные кулачки. Такие таймеры представляют собой идеальное средство для управления последовательностью команд в функции времени. Рассмотрим более детально некоторые из этих специальных устройств.

12.2. СПЕЦИАЛЬНЫЕ УСТРОЙСТВА ПК

К основным функциям ПК относятся сканирование, в том числе повторное, логических входов с высокой скоростью, оценка логических соотношений с целью определения состояний выходов в соответствии с логикой, запрограммированной на основе лестничной (ступенчатой) схемы, установления соответствующих состояний выходов в соответствии с запрограммированной логикой. В ряде случаев ПК может выполнять указанные функции без использования таймеров, счетчиков, аналого-цифровых преобразователей и других специальных устройств. Если такие устройства требуются, они могут быть внешними по отношению к ПК и могут подключаться к ПК через его выходы и входы. Для подключения указанных устройств производителем ПК разрабатывается программное обеспечение, которое записывается в постоянное запоминающее устройство ПК. Пользователь по желанию может подключить эти устройства к ПК.

Одним из специальных устройств ПК является таймер. Таймеры могут производить отсчет времени в прямом и обратном направлениях, осуществлять подсчет времени за один или несколько циклов функционирования контролируемого оборудования. Поскольку таймеры могут быть внешними устройствами ПК, они могут также применяться в обычных релейных схемах управле-

ния. В таком качестве таймеры рассматривались в гл. 10. Все характеристики таймеров, рассмотренные в гл. 10, также сохраняются для таймеров, встроенных в ПК.

Устройством, тесно связанным с таймером, является счетчик. Фактически таймер представляет собой счетчик импульсов, формируемых генератором. Счетчики рассмотрены в гл. 2 как независимые устройства. Встроенные в ПК счетчики имеют некоторые особенности.

Счетчики. Управление производством, контроль и управление материалами предусматривают необходимость подсчета дискретных компонентов. При выполнении счета в автоматическом режиме появляется множество преимуществ, помимо снижения затрат на живой труд. Автоматический счет является более точным, поскольку на операторов, как правило, одновременно возлагаются более ответственные задачи. Поэтому операторы забывают производить соответствующие подсчеты. Во многих случаях подсчеты представляют собой основу для оплаты. Поэтому при счете вручную существуют благоприятные возможности для обмана и конфликтов на почве обмана. Наивно полностью полагаться на честность всех рабочих. Даже относительно честные рабочие могут заявить, что они случайно забыли подсчитать, и таким образом оправдать искажение цифры при подсчете продукции.

Следует различать подсчет компонентов, находящихся в статическом состоянии, например на складе, и подсчет продукции, находящейся в динамическом состоянии, т. е. в процессе производства. Оператор, контролирующий состояние оборотных средств, может произвести подсчет объема продукции, произведет выборку и осуществит оценочный подсчет оборотных средств с некоторым преимуществом по сравнению со счетными машинами. Это преимущество сохранится даже в случае, если контролер будет частично использовать вспомогательные средства автоматического подсчета изделий. Однако подсчет продукции, находящейся в процессе производства, является одной из тех задач, которые легко поддаются автоматизации и являются дорогостоящими при ручном труде.

До сих пор нами рассматривались автоматические счетчики, которые автоматизируют счетный труд в соответствующих случаях, когда это является выгодным. Однако встроенные в ПК счетчики, как и внешние полупроводниковые счетчики, позволяют производить подсчет электрических импульсов на частотах, значительно превышающих человеческие возможности. При этом открываются возможности для автоматизации соответствующих процессов, которая в противном случае вряд ли могла быть осуществима. Даже логические таймеры, рассмотренные в гл. 11, представляют собой обычные счетчики импульсов, формируемых генератором, т. е. электрических импульсов, поступающих от генератора через точно заданные временные интервалы. Полупроводниковые устройства, обеспечивающие формирование и счет импульсов, поступающих с высокой частотой, привели к созданию импульсного датчика

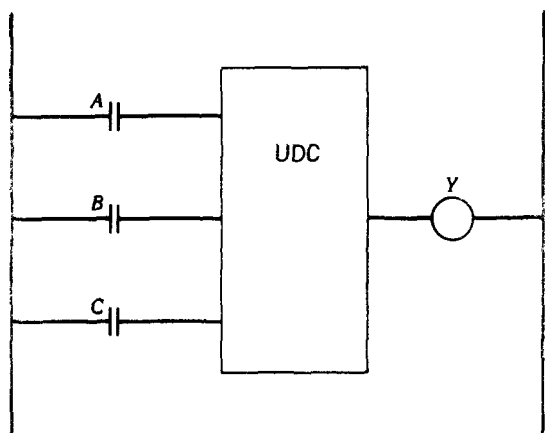


Рис. 12.3
Реверсивный счетчик с
ограничительной уста-
вкой 4 с

положения и других устройств, используемых в промышленных роботах и автоматизированных системах производства.

Максимальная частота импульсов является основным параметром счетчиков, работающих с импульсными датчиками положения. ПК обычно не используются для выполнения таких функций. Встроенный счетчик типового ПК используется для подсчета импульсов, поступающих от датчика, контролирующего технологический процесс. По достижении содержимым счетчика значения запрограммированной установки формируется выходной сигнал. Этот сигнал, поступающий с выхода счетчика, может управлять по программе состоянием «Включено» и «Выключено» выходов ПК, т. е. может использоваться для управления технологическим процессом или для внутреннего управления в качестве «релейного контакта» или другого элемента лестничной логической схемы. Счетчики ПК осуществляют счет в прямом или обратном направлении. При этом один вход счетчика используется для счета импульсов в прямом направлении, другой — в обратном. Такие счетчики называются реверсивными.

На рис. 12.3 показан реверсивный счетчик на лестничной логической схеме. На этом счетчике установлена уставка 4. Вход *A* соответствует счету в прямом направлении. Вход *B* соответствует счету в обратном направлении. Вход *C* соответствует сбросу счетчика на нуль. При сбросе вход *C* должен быть разомкнут. Реверсивный счетчик на рис. 12.3 имеет один выход *Y*. Наличие сигнала на этом выходе показывает, что содержимое счетчика при счете в прямом направлении достигает или превышает значение заданной установки. При счете в обратном направлении или сбросе наличие сигнала на том же выходе счетчика говорит о достижении его содержимым нулевого состояния или отрицательного числа. Другие счетчики могут располагать двумя выходами, что позволяет различить достижение содержимым счетчика заданной установки в прямом или в обратном направлении.

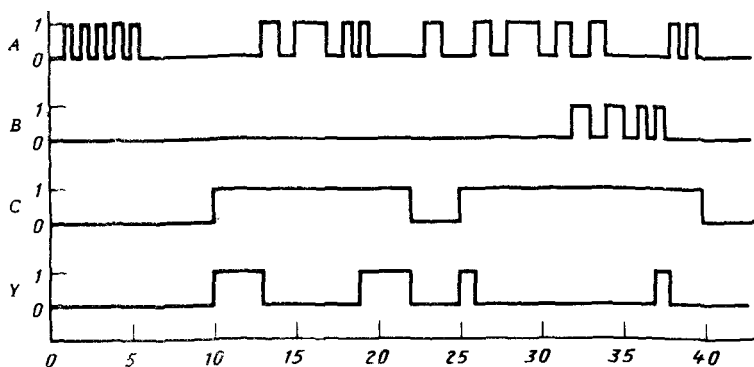


Рис. 12.4

Временная диаграмма для реверсивного счетчика на рис. 12.3

Действие реверсивного счетчика (рис. 12.3) можно пояснить с помощью временной диаграммы, представленной на рис. 12.4. Первые пять импульсов, поступающих с высокой частотой, не оказывают влияния на состояние выхода Y , поскольку сигнал на входе C отсутствует. Как только появляется сигнал на входе C , на выходе Y также возникает сигнал, соответствующий нулевому состоянию счетчика. При появлении первого импульса на входе A сигнал на выходе Y равен нулю. С передним фронтом четвертого

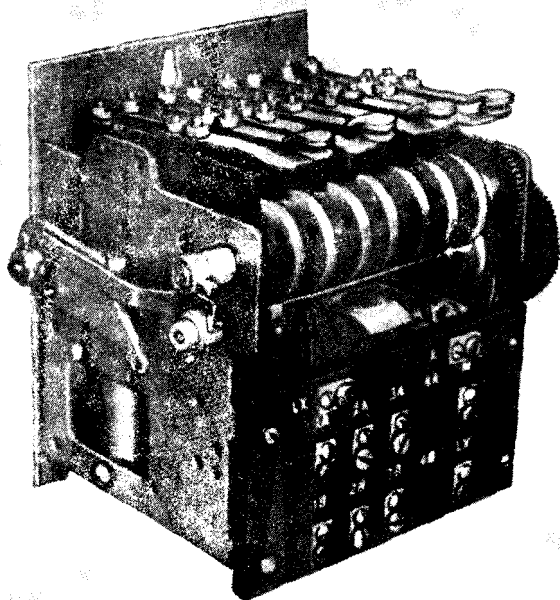


Рис. 12.5

Кулачковый или барабанный таймер

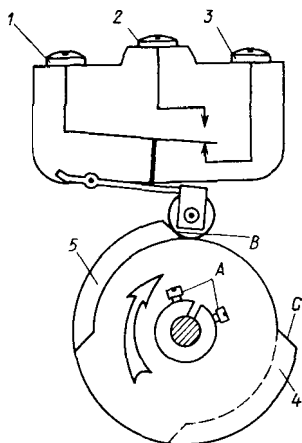


Рис. 12.6

Схема работы кулачка в составе барабанного таймера. Регулировочные винты *A* предназначены для регулировки положения упора *B* (положение «Включено» для замыкающего контакта) и положения упора *C* (положение «Включено» для размыкающего контакта):

1 — общий зажим; 2 — зажим замыкающего контакта; 3 — зажим размыкающего контакта; 4 — левая часть кулачка; 5 — правая часть кулачка

импульса на входе *A* появляется сигнал на выходе *Y*. Длительность входных импульсов на рис. 12.4 показана не в реальном масштабе. В момент, соответствующий значению 22, производится повторный сброс сигнала на выходе, поскольку снимается сигнал со входа *C*. На входе *C* сигнал отсутствует, поэтому импульсный сигнал на входе *A* в момент времени 23 не воспринимается. В момент времени 25 на входе *C* вновь появляется импульсный сигнал, что приводит к появлению соответствующего сигнала на выходе *Y* до тех пор, пока на входе *A* держится импульсный сигнал. Затем на входах «плюс» (вход *A*) и «минус» (вход *B*) почти одновременно появляются импульсные сигналы. Содержимое счетчика достигает первой уставки («Нуль»), и на выходе *Y* опять устанавливается сигнал. В момент времени 40 со входа *C* опять снимается сигнал, вновь сбрасывая таймер.

Таймеры барабанного типа. Таймерное устройство последовательного действия, контроллер без обратной связи, представляет собой таймер кулачкового типа, показанный на рис. 12.5. Вращающийся вал выполняет функцию часового вала. По длине кулачка находятся контакты конечного выключателя, который срабатывает при контакте с кулачком в соответствии с настройкой по программе. Такое устройство часто называется таймером барабанного типа. Программирование таймеров барабанного типа сводится к обычной наглядной механической настройке и использованию регулировочных винтов, как показано на рис. 12.6. В ПК в действительности отсутствуют какие бы то ни было механические кулачки или барабан, но используются встроенные таймеры, моделирующие работу механических таймеров барабанного типа. Такие устройства-таймеры позволяют быстрее выполнять программирование. Встроенный в ПК таймер выполняется электронными средствами, моделирование работы механического таймера осуществляется с помощью резидентного программного обеспечения непосредственно в ПК. Пользователю предоставляется доступ к модели, реализованной программным путем. При этом пользователь идентифицирует конечные выключатели, которые срабатывают от нажатия каждым «кулачком», частоту вращения вала модели или барабана (частота может быть задана переменным

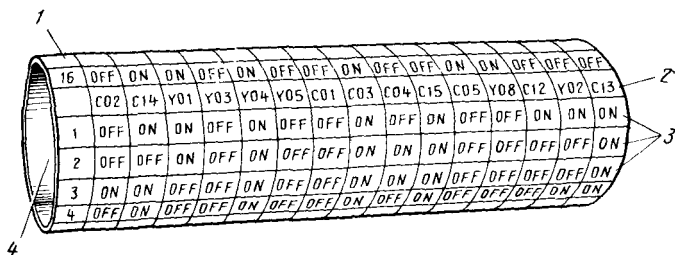


Рис. 12.7
Модель барабанного таймера в ПК:

1 — шаг; 2 — выходы; 3 — маски; 4 — предварительная настройка (исходное положение)

параметром) и конфигурацию (маску) состояний (включено-выключено) выходов для каждого отдельного шага смоделированного вращения барабана.

Дискретные шаги могут различаться по длительности (во времени), поскольку может быть задано количество импульсов на один шаг. Импульсы имеют постоянную взаимосвязь с внутренней частотой сканирования ПК или с циклом работы ПК. Например, если продолжительность цикла ПК синхронизирована с частотой 60 Гц, то длительность самого малого шага барабанного таймера ПК будет соответствовать частоте 1/60 с или 16,67 мс. Чтобы барабанный таймер выдавал сигналы через такой короткий промежуток времени, следует определить функцию «количество импульсов на шаг» как 1. Задание 60 импульсов/шаг будет соответствовать продолжительности шага барабанного таймера 1 с, а задание 3600 импульсов/шаг будет соответствовать продолжительности шага 1 мин. Таким образом, изменяя значение количества импульсов на шаг, можно смоделировать различные значения частоты вращения барабана.

Физическая зависимость между шагами, количеством импульсов на шаг и маской состояния выходов показана на рис. 12.7. Если бы изображенный на рис. 12.7 барабан в действительности был механическим, а не моделью, то он вращался бы против часовой стрелки, если смотреть с левой стороны барабана. Идентификация выходов, которые выбирает программист для каждого кулачка, производится в соответствующем ряду барабана, который представляет собой полый цилиндр, с левой стороны которого отсутствуют маски выходов. Каждый из шестнадцати шагов, расположенных по периферии барабана, представляет различную маску для пятнадцати колец с кулачками. При первом шаге подаются сигналы на входы C14, Y1, Y4, C3, C15, C12, Y2 и C13. Программа продолжает выполняться при повороте барабана вплоть до шестнадцатого шага. Во время отработки программы подаются сигналы на выходы C14, Y1, Y4, C3, C5 и Y8.

Перед программированием барабанного таймера ПК целесообразно составить технологическую карту, в которой указы-

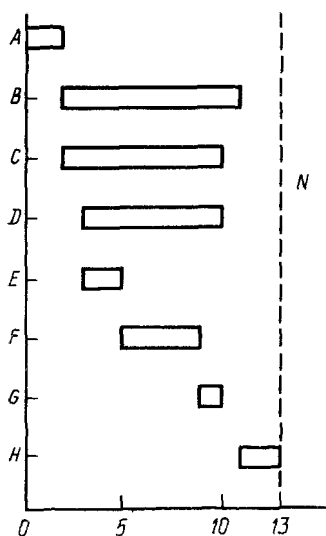


Рис. 12.8

Диаграмма работы таймера в примере 12.1:

A — загрузка с помощью робота; B — зажим; C — включение охлаждения; D — включение шпинделя; E — подвод инструмента; F — цикл работы станка; G — отвод инструмента; H — разгрузка с помощью робота; N — следующий цикл

ваются начальный шаг и функции, выполняемые на каждом шаге. Поскольку в барабанных таймерах используется метод последовательного программирования, начало каждого цикла определяется и перепрограммируется без сигнала обратной связи об окончании предыдущего шага. После определения начала и конца каждой операции составляется технологическая карта, которая

представляет собой матрицу, определяющую маски выходов на барабанном таймере. В начале каждого нового шага при повороте барабана изменяется в соответствии с программой состояние одного или нескольких выходов (с «Включено» на «Выключено» или наоборот). На одном и том же этапе одновременно может быть изменено состояние нескольких выходов. В примере 12.1 рассмотрен цикл, состоящий из нескольких таких шагов.

Пример 12.1. Автоматический цикл станка с цикловым управлением в функции времени. Рассмотрим особенности работы обычного станка в автоматическом цикле с управлением от ПК, оснащенного программируемым таймером барабанного типа. Загрузка-разгрузка станка осуществляется с помощью промышленного робота, который функционирует в соответствии с запрограммированными

12.1. Матрица барабанного таймера для примера 12.1

Количество опросов барабана № 1 (в автоматическом цикле станка) $n = 60$

Шаг	Количество опросов/шаг	Выходы							
		A	B	C	D	E	F	G	H
1	2	1	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	0	0
3	2	0	1	1	1	1	0	0	0
4	4	0	1	1	1	0	1	0	0
5	1	0	1	1	1	0	0	1	0
6	1	0	1	0	0	0	0	0	0
7	2	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0

циклами загрузки и разгрузки 2 с. Различные функции выполняются в соответствии с запрограммированной последовательностью, и реализация некоторых из этих функций совмещается во времени. Диаграмма полного цикла, включающего различные функции, представлена на рис. 12.8. Программист-оператор выбирает соответствующий выход ПК, чтобы активизировать каждую функцию (выходы обозначены буквами от *A* до *H*). Затем эти выходы идентифицируются с помощью соответствующих колец кулачков барабанного таймера. Для определения шагов барабанного таймера и масок выходов для каждого шага используется матрица, показанная в табл. 12.1. Допускается использование меньшего числа шагов, предусмотренных в таймере. Обе указанные возможности проиллюстрированы на матрице в табл. 12.1, в которой определены лишь восемь выходов и семь (из десяти возможных) шагов. Неиспользованным шагам 8, 9 и 10 присвоено холостое состояние путем задания нулей в графе «Количество импульсов/шаг». Задание 0 импульсов/шаг обуславливает практически мгновенную обработку этого шага, и, следовательно, этот шаг пропускается. Определение семи заданных шагов производится с помощью контроля начала и конца каждой функции, указанной на диаграмме, приведенной на рис. 12.8. В каждый момент времени, когда происходит какое-либо событие, т. е. начинается или заканчивается выполнение одной или нескольких функций, должен быть идентифицирован новый шаг.

В примере 12.1 количество циклов опросов составляет 60, что предполагает использование в ПК цикла частотой 60 Гц и длительностью 1 с между соседними импульсами. Эти параметры являются оптимальными для примера 12.1, поскольку длительность реализации соответствующих функций измеряется в секундах. Таким образом, числа, указанные в графе «Количество импульсов/шаг», имеют размерность секунды. Маски выходов представляются на языке алгебраической логики с использованием «0» и «1»: «0» соответствует состоянию «Выключено», «1» — «Включено». Переход от шага к шагу осуществляется в непрерывном режиме, обеспечивая на выходах состояния «Включено» при переходе от одного шага к другому. Например, выход *B* остается в состоянии «Включено» при обработке шагов от двух до шести, несмотря на то, что состояния других выходов в это время подвергается изменениям.

Рассмотрение примера 12.1, а также особенностей счетчиков, таймеров и барабанных таймеров должно дать читателю представление об огромных возможностях использования ПК в области робототехники и автоматизации производства. Теперь обратимся к вопросу программирования ПК и рассмотрим, насколько просто эти возможности реализуются на практике.

12.3. ПРОГРАММИРОВАНИЕ ПК

Методы программирования ПК являются различными для каждого производителя. Однако основная концепция, предусматривающая при программировании ПК использование логической ступенчатой схемы, по-видимому, применяется во всех отраслях промышленности. В большинстве случаев итеративный цикл в ПК не принимается во внимание программистом, который считает, что логическая система управления обеспечивает одновременно выполнение функций ступеней логической схемы.

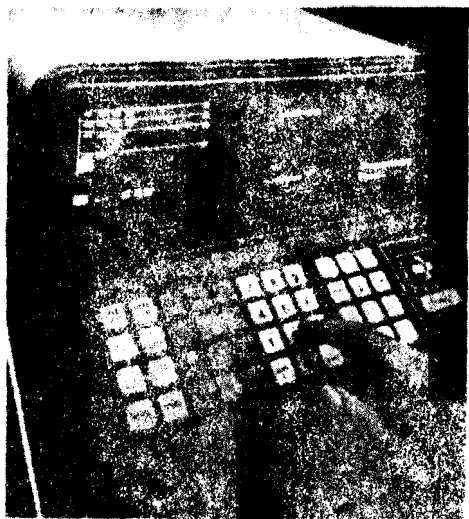


Рис. 12.9

Клавиатура и дисплей программатора ПК

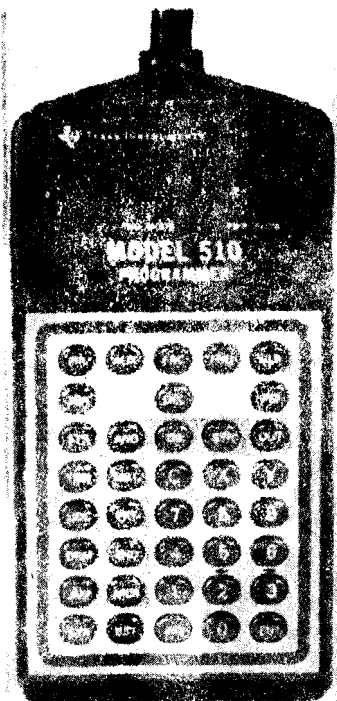


Рис. 12.10

Малогабаритный переносной программатор для ПК

Некоторые производители оснащают ПК графическим дисплеем с экраном, на котором воспроизводится лестничная схема. Последовательное воспроизведение такой схемы осуществляется программистом с помощью клавиатуры. На рис. 12.9 показан пример широко распространенной модели программатора ПК с клавиатурой и дисплеем, на экране которого изображена реальная логическая схема. Следует отметить, что на кнопках клавиатуры нанесены символы контактов, которые интерпретируются как показано в гл. 11. В гамму программаторов ПК входит малогабаритный ручной, выполненный в виде подвесного пульта, программатор, общий вид которого показан на рис. 12.10. На каждой кнопке этого программатора нанесена аббревиатура из двух-трех букв, соответствующая mnemonicским обозначениям функций и кодов, используемых оператором при программировании ПК. В верхней части подвесного пульта программатора находится указатель, выполненный на светодиодах, с помощью которых индицируются команды ПК, хранящиеся в оперативной памяти.

На лицевой стороне программатора, показанного на рис. 12.10, можно видеть кнопки, соответствующие логическим операторам И, ИЛИ и НЕ. Эти кнопки применяются для формирования логики данной ступени логической схемы. Начало программирования ступени схемы устанавливается с помощью кнопки *STR*, которая

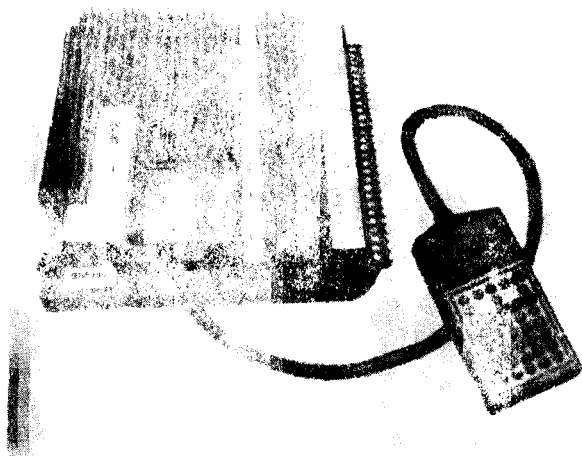


Рис. 12.11

ПК типа TI-510 с подключаемым малогабаритным программатором

соответствует функции START или STORE. После нажатия этой кнопки вводится изображение первого входного контакта программируемой ступени схемы. Входные контакты обозначаются переменными X , выходы — переменными Y , промежуточные реле управления — переменными C . Под «реле управления», которые обеспечивают управление входными контактами в лестничной логической схеме, понимаются выходы ступени этой схемы. Однако «реле управления» не используются в качестве выходов ПК для связи с внешними устройствами. Программирование каждой ступени логической схемы заканчивается командой OUT, за которой следует переменная Y или C .

ПК TI-510 фирмы «Тексас Инструментс» (США) представляет собой систему, которая оснащается программатором, показанным на рис. 12.10. ПК TI-510 является широко распространенным недорогим ПК, пригодным для использования в лабораториях

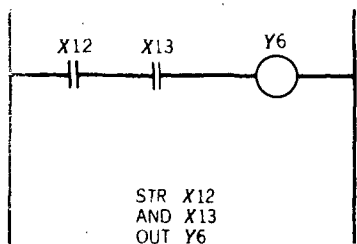


Рис. 12.12

Программа ПК для реализации схемы И

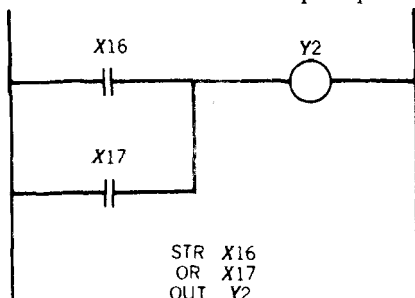
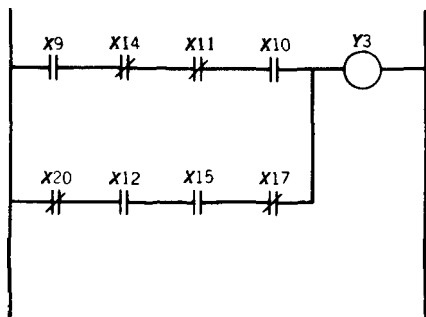


Рис. 12.13

Программа ПК для реализации схемы ИЛИ



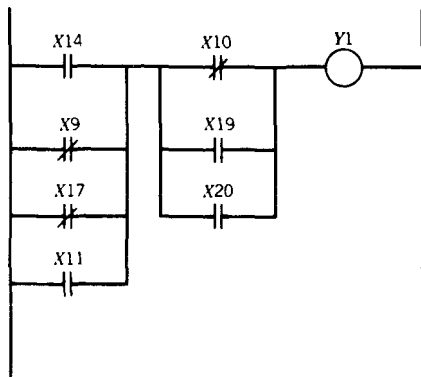
```

STR X9
AND NDT X14
AND NDT X11
AND X10
STR NDT X20
AND X12
AND X15
AND NDT X17
DR STR
DUT Y3

```

Рис. 12.14

Лестничная схема, состоящая из двух параллельных ступеней, подключенных к одному выходу с использованием команды OR STR



```

STR X14
DR NDT X9
DR NDT X17
DR X11
STR NDT X10
DR X19
DR X20
AND STR
DUT Y1

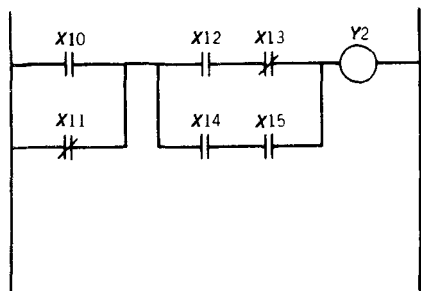
```

Рис. 12.15

Схема, состоящая из двух последовательных соединенных многозвенных ступеней, подключенных к одному выходу с использованием команды AND STR

колледжей и университетов. Программа такого ПК может быть представлена в виде распечатки с комментариями. На рис. 12.11 изображен ПК TI-510 с подключенным к нему малогабаритным программатором. Видна колонка зажимов ПК, предназначенных для подключения входов и выходов управляемого оборудования. В ПК TI-510 предусмотрены восемь выходов, обозначенных Y1—Y8, и 12 входов с обозначениями X9—X20.

Рассмотрим несколько примеров простых логических систем. На рис. 12.12—12.15 приведены лестничные логические схемы с соответствующими им программами для ПК. Каждая строка этих программ вводится в память посредством кнопки ENT после записи этой строки в соответствующем коде. В первых двух примерах (см. рис. 12.12 и 12.13) показаны наиболее распространенные базовые лестничные логические схемы. Любой входной контакт может быть инвертирован в соответствии с условием НЕ простым нажатием кнопки NOT непосредственно перед вводом соответствующего обозначения входного контакта. Эта процедура выбора входных контактов показана на рис. 12.14 и 12.15. Команды OR STR и AND STR используются для связи сборок и схем логических контактов. Каждый раз, как только начинается программирование новой сборки входных контактов, используется кнопка STP. Программисту предоставляется возможность путем



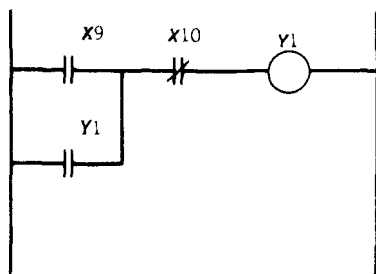
```

STR X10
OR NOT X11
STR X12
AND NOT X13
STR X14
AND X15
DR STR
AND STR
DUT Y2

```

Рис. 12.16

Лестничная схема с использованием команд OR STR и AND STR



```

STR X9
OR Y1
AND NOT X10
OUT Y1

```

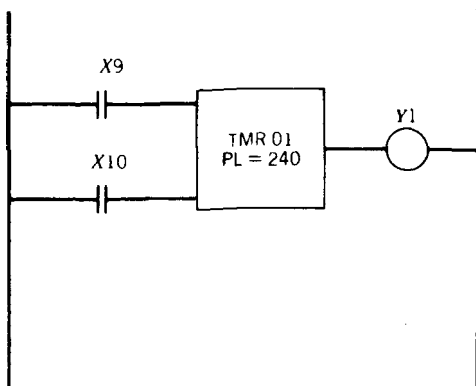
Рис. 12.17

Использование выхода в качестве входного контакта; пример реализации простой кнопки включения

последовательного использования кнопок OR STR и/или AND STR формировать гнезда схем. На рис. 12.16 представлено подобное гнездо. В этом примере команда OR STR обеспечивает связь параллельных групп контактов $X14$, $X15$ и $X12$, $X13$. Затем командой OR STR обеспечивается последовательная связь указанной группы с группой параллельно включенных контактов $X10$, $\overline{X11}$.

В приведенных примерах входы ПК рассматриваются в виде контактов. Возможно использование выходов одних ступеней логической схемы в качестве входов других ступеней схемы (рис. 12.17). Схема на рис. 12.17 выполняет функцию кнопки и рассмотрена в гл. 10 и 11. Определим, какой контакт кнопки выбран в качестве размыкающего и какой в качестве замыкающего. Выбор входов и выходов является произвольным, поэтому программист выбирает в системе любой из имеющихся входов и выходов по желанию. Какой выбор входов и выходов программист ни сделал бы, другой программист или оператор располагает возможностью осуществить их привязку при отладке программы ПК.

Если определенный выход на ступени логической схемы не может быть использован для подключения какого-либо внешнего устройства, то подключения выходного терминала Y не требуется. Таким образом, в качестве выхода ступени логической схемы может использоваться любое из 50 промежуточных реле управления $C1—C50$. В этом случае реле управления могут использоваться в качестве входных контактов другой или даже той же самой ступени (или ступеней) соответствующей логической схемы. Если реле управления используются в качестве выходов на лестничной



STR X9
STR X10
TMR 01
PL = 00240
OUT Y1

Рис. 12.18
Программирование суммирующего таймера на время до 4 с для ПК

логической схеме, то их входные контакты могут использоваться в любом месте этой схемы. Реле управления могут использоваться для перехода на следующие шаги при цикловом управлении после проверки выполнения предыдущего шага при управлении с обратной связью.

Отображение таймера ПК в коде показано на рис. 12.18. Таймер представляет собой суммирующий таймер, в котором контакт *X10* разблокирует таймер, в то время как замыканием контакта *X9* осуществляется сброс таймера. Таким образом, контакты *X9* и *X10* являются по отношению к таймеру управляющими. Параметры этого таймера в точности соответствуют параметрам таймера с двумя входами (см. рис. 11.28). По достижении значения заданной уставки на выходе таймера появляется сигнал. Когда состояние контакта (*X10* на рис. 12.18) становится соответствующим логическому 0, сигнал с выхода таймера снимается, а сам таймер сбрасывается в состояние нуль. С этого момента цикл отсчета времени может начаться снова, как только произойдет активизация *X9* и снимется сигнал *X10*.

На рис. 12.19 представлена временная диаграмма таймера, показанного на рис. 12.18. Заданная уставка таймера соответствует 4 с. Эта уставка задана программистом путем определения переменной *PL* числом 240. Число 240 представляет собой время, измеряемое с дискретностью 1/60 с. Следовательно, 240/60 соответствует 4 с. Временная диаграмма показывает, что рассматриваемый таймер действует подобно таймеру с одним входом, если оба контакта *X9* и *X10* оказываются одновременно замкнутыми. Оба контакта замыкаются в момент времени 2. Тогда через 4 с в момент времени 6 на выходе таймера возникает сигнал, сохраняющийся до момента времени 10, когда оба входа не будут отключены. С момента времени 10 до момента времени 25 реализуется функция суммирования таймера. Индикация функции суммирования обеспечивается включением контакта *X10* с момента времени 12. После этого в отдельные переменные временные периоды контакт *X9* замыкается и размыкается до момента времени 22, когда содержимое таймера будет соответствовать уставке 4 с. В этот момент на выходе *Y1* появляется сигнал. При этом следует обратить внимание

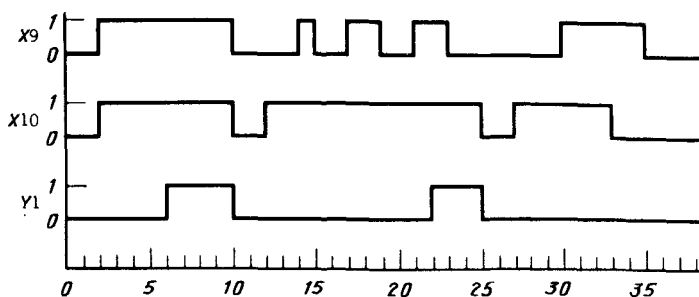


Рис. 12.19

Временная диаграмма работы таймера ПК, изображенного на рис. 12.18

на то, что на этом выходе сигнал *сохраняется* даже после того, как контакт $X9$ размыкается в момент времени 23. После отключения контакта $X10$ в момент времени 25 сигнал на выходе таймера становится равным нулю, и таймер сбросом подготавливается к работе в следующем цикле. Хотя рассматриваемый таймер включается снова в момент времени 27 и остается в этом состоянии в течение 6 с, на выходе сигнал не возникает, поскольку контакт $X9$ замыкается лишь на 3 с перед тем, как контакт $X10$ разомкнется снова. То, что контакт $X9$ остается в замкнутом состоянии до момента времени 35 (всего 5 с), не имеет значения, поскольку на выходе $Y1$ сигнал не появится до тех пор, пока не будет произведен сброс таймера (размыканием контакта $X10$).

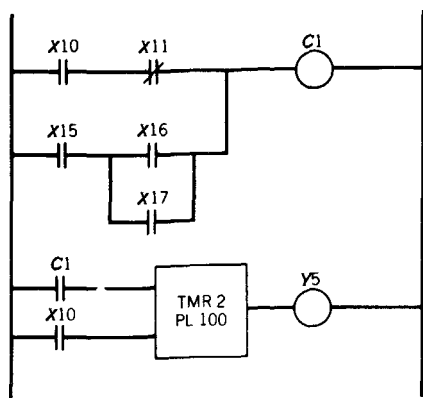
В рассматриваемой программе таймера ПК существует одна деталь, которую нужно пояснить. В четвертой строке программы уставка таймера запрограммирована с помощью следующей команды: $PL = 00240$.

Однако на клавиатуре пульта программатора отсутствуют кнопки PL или кнопка « $=$ ». Объяснение состоит в том, что указатель на светодиодах, находящийся на пульте, дает автоматическую подсказку программисту. Эта подсказка обеспечивает вывод на индикацию выражения

$$PL = 00000$$

каждый раз, как только устанавливается команда TMR . В этом случае программисту нужно произвести лишь ввод числа 240, которое заменяет соответствующие нули с правой стороны указанного выражения на индикации. После этого нажатием кнопки ENT производится ввод в память ПК любого числа, относящегося к какой-либо команде.

Входы в таймер, изображенный на рис. 12.18, представляют собой обычные контакты. Однако вполне возможно использование в подобных случаях для запуска таймера сложной логической схемы. Приведенный на рис. 12.20 пример служит для иллюстрации использования промежуточного реле управления. Вход $X10$ используется в двух ступенях логической схемы (одноименные

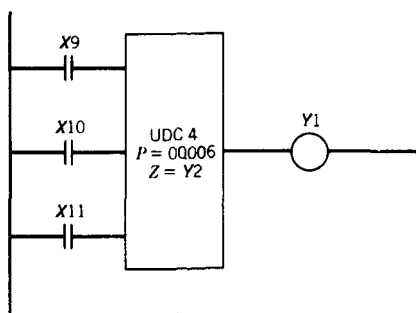


STR X10
AND NDT X11
STR X15
STR X16
OR X17
AND STR
OR STR
OUT C1
STR C1
STR X10

TMR 2
PL = 00100
OUT Y5

Рис. 12.20

Синхронизированный с помощью таймера ПК выход звена сложной ступенчатой схемы; синхронизированный выход заменяет несинхронизированный вход



STR X9
STR X10
STR X11
UDC 4
PL = 00006
Z = Y2
DUT Y1

Рис. 12.21

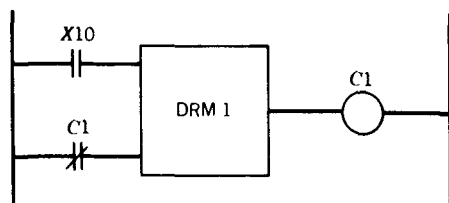
Программирование реверсивного счетчика ПК

входы могут повторяться столько раз, сколько это необходимо в лестничной логической схеме).

Программирование реверсивного счетчика практически аналогично программированию таймера. На рис. 12.21 показан счетчик

с уставкой ограничения. Функционирование этого счетчика осуществляется аналогично работе универсального реверсивного счетчика, который был описан ранее (см. рис. 12.3 и 12.4). Счетчик на рис. 12.21 фактически имеет два выхода: $Y1$ и $Y2$. На стандартном выходе $Y1$ появляется сигнал, когда содержимое счетчика достигает заданной уставки при работе в любом направлении — «плюс» или «минус». На выходе Z ($Y2$) появляется сигнал только в случае, когда содержимое счетчика соответствует нулю. При необходимости любой из этих выходов может быть обозначен как промежуточное реле управления C .

На рис. 12.22 показано программирование барабанного таймера с использованием примера 12.1. На рис. 12.22 приведена лестничная логическая схема, поскольку в программе не могут быть воспроизведены все функциональные особенности барабанного таймера. На схеме показаны два входа и один выход на барабанном таймере. Нижний вход, как в обычном таймере, представляет собой разблокированный выключатель барабанного таймера. Верхний вход является старт-стопным входом барабанного таймера, аналогичным верхнему запускаящему входу обычного таймера. На выходе барабанного таймера появляется сигнал, после того как барабан завершает цикл из 16 шагов. Барабанный таймер,



STR X10	12 =	09 = 00000	RD
STR NOT C1	13 =	10 = 00000	01111000
DRM 1	14 =	11 = 00000	MASK 04 =
PRSET = 01	15 =	12 = 00000	RD
OUTPUTS	SCN/CNT	13 = 00000	01110100
RD	RD	14 = 00000	MASK 05 =
01 = Y1	SC = 00060	15 = 00000	RD
02 = Y2	CNT/STP =	16 = 00000	01110010
03 = Y3	RD	MASKS =	MASK 06 =
04 = Y4	01 = 00002	RD	RD
05 = Y5	02 = 00001	MASK 01 =	01000000
06 = Y6	03 = 00002	RD	MASK 07 =
			RD
07 = Y7	04 = 00004	10000000	00000001
08 = Y8	05 = 00001	MASK 02 =	MASK 08 =
			(through MASK 16)
09 =	06 = 00001	RD	RD
10 =	07 = 00002	01100000	00000000
11 =	08 = 00000	MASK 03 =	OUT C1

Рис. 12.22

Программирование барабанного таймера ПК (пример 12.1)

изображенный на рис. 12.22, настраивается на непрерывную работу, поскольку на входе *C1*, который используется в качестве входа в барабанный таймер, сигнал постоянно отсутствует. Таким образом, барабанный таймер будет функционировать лишь в случае, когда на входе *C1* сигнал будет отсутствовать. Как только цикл барабанного таймера завершится, на входе *C1* появится сигнал, который немедленно обеспечит сброс таймера. Последний отключит вход *C1*, что разрешит начало следующего цикла. Так будет обеспечено повторение цикла работы барабанного таймера. Отключение входа *X10* обеспечивает остановку работы барабанного таймера, но не сбрасывает его в нулевое состояние.

После программирования двух входов и определения барабанного таймера с помощью кнопки DRM начинается последовательное программирование матрицы барабанного таймера. В начале ПК подсказывает пользователю, что требуется ввести определенный номер первого шага цикла таймера. Большинство пользователей ПК в качестве начального, возможно, выберут шаг № 1. Затем ЭВМ подскажет следующую последовательность команд:

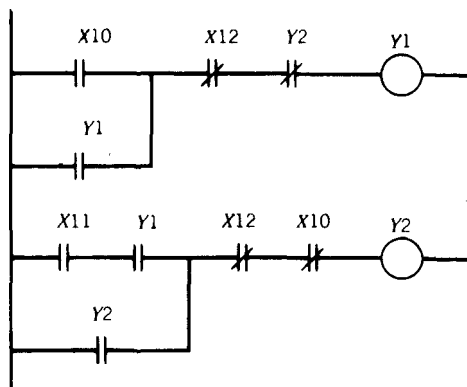
OUTPUTS	MASK 02
SCN/CNT	.
CNT/STP	.
MASKS	.
MASK 01	MASK 16

После каждой из приведенных подсказок пользователь нажимает кнопку RD (что означает СЧИТЫВАНИЕ), это приводит к появлению на дисплее ряда последовательных подсказок, которыми должен руководствоваться пользователь при вводе параметров матрицы в соответствии с рисунком. Сравнивая строки команд, приведенных на рис. 12.22, с соответствующими параметрами матрицы примера 12.1, приведенного на рис. 12.8, можно понять структуру формата при программировании барабанного таймера ПК.

Хотя при программировании полного барабанного таймера необходимо определить все его параметры, в общем случае программирование ПК представляет собой достаточно простую задачу. Редактирование и контроль программ имеют особенности, учитывающие возможности ПК. Многие ПК имеют сложные конструктивные особенности, связанные с управлением обратной связью аналогового процесса. Однако в данной главе рассмотрены лишь основные особенности программирования ПК. Рассмотрим принципы подготовки программ ПК для автоматизированного производства или управления промышленным роботом.

12.4. ПРОГРАММЫ ПОЛЬЗОВАТЕЛЯ

Примеры программирования ПК, приведенные выше, являлись показательными, соответствующие им программы не имеют практического значения. Рассмотрим примеры программирования ПК применительно к некоторым типам оборудования в автоматизированном производстве.



```

STR  X10
OR   Y1
AND NOT X12
AND NOT Y2
OUT  Y1
STR  X11
AND  Y1
OR   Y2
AND NOT X12
AND NOT X10
OUT  Y2

```

Пример 12.2. Управление скоростью конвейера автоматической сборочной линии. Допустим, что ПК используется для управления сборочной линией. Пуск линии должен производиться на малой скорости, а затем должно осуществляться переключение на большую скорость. При этом входы и выходы определяются следующим образом:

X10 — кнопка мгновенного действия, обеспечивающая пуск линии на малой скорости;

X11 — кнопка мгновенного действия, обеспечивающая переключение линии на большую скорость;

Рис. 12.23
Программа ПК для управления скоростью конвейера сборочной линии (пример 12.2)

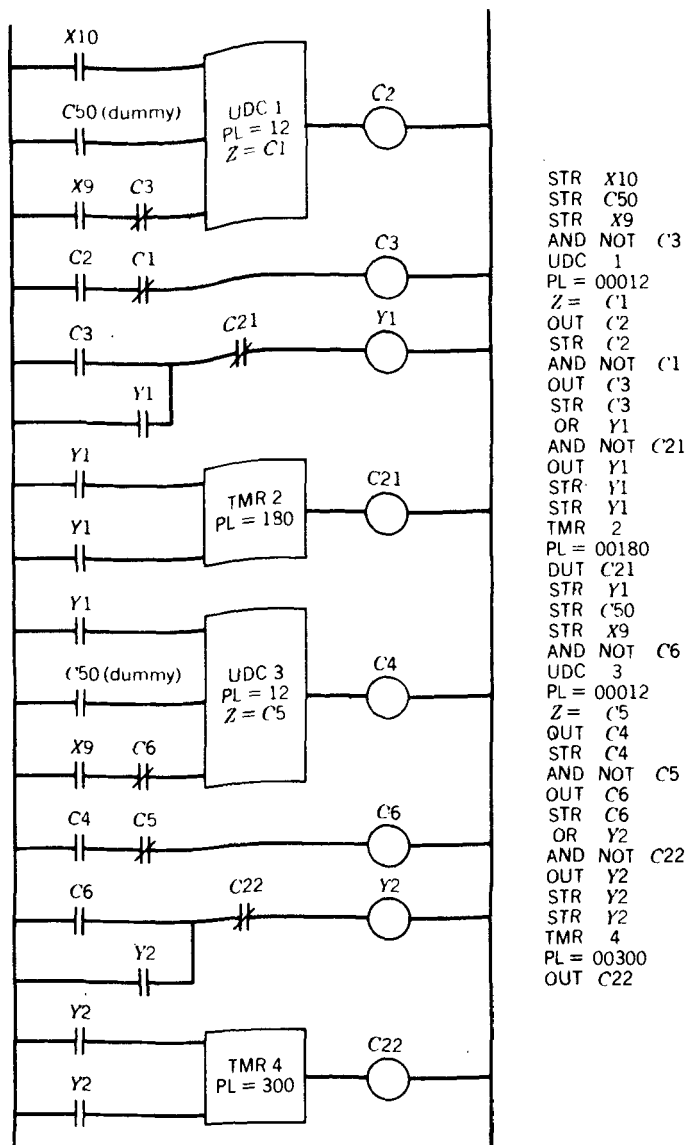


Рис. 12.24

Программа ПК для управления автоматической упаковкой (пример 12.3)

X12 — кнопка мгновенного действия, обеспечивающая остановку линии;

Y1 — малая скорость;

Y2 — большая скорость.

На рис. 12.23 приведена лестничная логическая схема и программа ПК, соответствующие приведенным выше условиям.

Пример 12.3. Управление автоматической упаковкой. Пузырьки с фармацевтическим средством упаковываются в коробки, каждая из которых рассчитана

на 12 пузырьков. Коробки устанавливаются в контейнер, рассчитанный на 144 пузырька. Упаковочный автомат получает сигнал (длительностью 3 с) от счетчика изделий каждый раз, когда на автоматический конвейер подаются 12 пузырьков, и другой сигнал (длительностью 5 с), когда контейнер заполнен 144 пузырьками.

При программировании ПК идентифицируются следующие входы и выходы:

X9 — система включена;

X10 — датчик счетчика изделий;

Y1 — сигнал на упаковку в коробки (3 с);

Y2 — сигнал на упаковку в контейнер (5 с).

На рис. 12.24 для приведенного примера представлены лестничная логическая схема и программа ПК. На первый взгляд, схема на рис. 12.24 может показаться чересчур сложной. Однако это не так. Такие громоздкие схемы являются типичными для практического применения. Обычно после проведенного анализа подобные схемы разбиваются на отдельные элементы, приемлемые для восприятия. Следует отметить, что *отображение* нижней части схемы, представленной на рис. 12.24, представляет собой копию верхней части той же схемы. Нижняя часть схемы отличается от верхней только наименованиями компонентов и цифровыми индексами. Каждая элементарная ступень рассмотрена выше. Назначение каждой ступени становится ясным после изучения представленной схемы и следующих параграфов.

Промежуточные реле управления используются в нескольких местах рассматриваемой программы. Реле управления C50 использовано только для наладки реверсивных счетчиков и других функций не выполняет. Счетный вход «минус» в данном случае не имеет существенного значения, поскольку количество производимых изделий никогда не уменьшается. Выходы счетчика представляют собой также реле управления, так как сигналы на них возникают, когда содержимое счетчика становится нулем или достигает заданной уставки (12). Поскольку сигнал на выходе счетчика нужен лишь при достижении его содержимым максимальной уставки, то для снятия этого сигнала в состоянии счетчика «0» вместо «12» требуется следующая ступень схемы. Для первого счетчика используется результирующий выход C3. Сигнал на этом выходе обеспечивает сброс этого счетчика и появление соответствующего сигнала на выходе Y1. Сигнал на выходе Y1 удерживается в течение 3 с таймером TMR2. Аналогичная схема используется для сигнала длительностью 5 с, который фиксирует заполнение контейнера на 144 пузырька.

В цифровых обозначениях элементов счетчиков и таймеров повторения отсутствуют. В некоторых ПК для таймеров и счетчиков предусмотрено одинаковое распределение памяти, что делает невозможным обращение к какому-либо таймеру и счетчику, имеющим одно цифровое обозначение.

12.5. УПРАВЛЕНИЕ ПРОМЫШЛЕННЫМИ РОБОТАМИ

Существуют два типа управления роботами с помощью ПК: управление в централизованном режиме и непосредственное управление. Рассмотрим особенности обоих типов управления.

Управление роботами в централизованном режиме. ПУ могут использоваться для централизованного управления оборудованием автоматизированного производственного процесса. В состав этого оборудования входит и робот. ПК управляет роботом по программе, обеспечивая выполнение последовательности операций в заданные моменты времени. ПК может также контролировать различные функции, взаимодействие с технологическим оборудованием, отклонение условий от нормальных или выявлять условия,

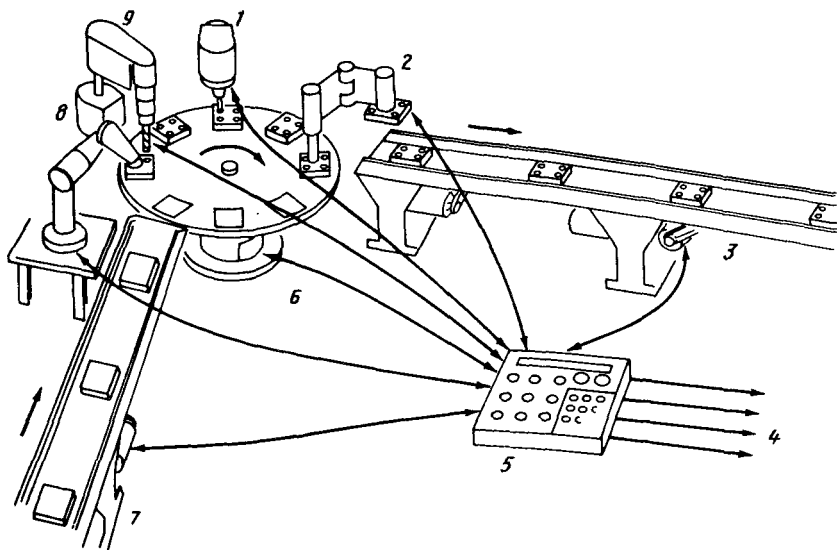


Рис. 12.25

Централизованное управление с помощью ПК автоматизированным производственным процессом с использованием промышленных роботов:

1 — инструмент для автоматической сборки; 2 — станция роботизированной разгрузки; 3 — привод разгрузочного конвейера; 4 — к другому производственному оборудованию; 5 — ПК; 6 — поворотный индексированный механизм; 7 — привод подающего конвейера; 8 — роботизированная станция загрузки; 9 — станок

приводящие к аварийной ситуации. При этом ПК осуществляет управление в соответствии с результатами контроля. При управлении промышленными роботами в централизованном режиме кроме центрального ПК робот оснащен автономным специализированным устройством управления, которое обеспечивает управление перемещениями рабочих органов робота в соответствии с программой. Однако на ПК могут быть возложены более широкие функции по сравнению с автономным устройством управления робота, как это видно из схемы, представленной на рис. 12.25.

Непосредственное управление роботами. ПК могут применяться не только в качестве устройств связи и синхронизации функционирования роботов с производственным оборудованием, но и для непосредственного управления роботом. Вид панели управления серийно выпускаемым роботом показывает, что во многих случаях в качестве устройства управления роботом используется ПК. ПК представляют собой идеальное средство для обеспечения простейших перемещений рабочих органов роботов. Наиболее часто ПК используются с пневматическими роботами типа взять-положить. С учетом информации, приведенной в гл. 2, 7, 10—12, можно *создать* собственный робот типа взять-положить с использованием пневмоцилиндров, золотников и конечных выключателей. Ниже рассмотрены два примера роботов

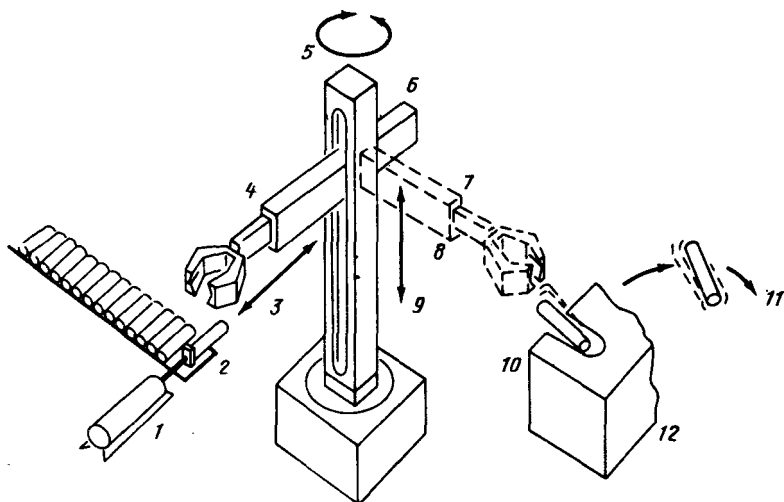


Рис. 12.26

Простой робот типа взять-положить, оснащенный ПК:

1 — устройство подачи деталей; 2 — позиция захвата; 3 — отвод; 4 — позиция руки для поворота по часовой стрелке; 5 — поворот по часовой стрелке; 6 — поворот против часовой стрелки; 7 — позиция руки для поворота против часовой стрелки; 8 — подъем; 9 — опускание; 10 — позиция для укладывания детали; 11 — автоматическое удаление деталей; 12 — станция обработки

типа взять-положить. В одном примере использовано управление в функции времени без обратной связи, во втором — управление с обратной связью.

Пример 12.4. Управление с помощью ПК в функции времени роботом типа взять-положить. Рассмотрим простой пример применения робота, представленный на рис. 12.26. Специалист по робототехнике определит представленный робот как робот типовой конфигурации, функционирующий в цилиндрической системе координат. Наиболее вероятно, что представленный робот оснащен цикловым управлением с помощью упоров. В этом примере можно ввести дополнительную координату для обеспечения движения захвата. Подобный робот может быть оснащен системой управления, выполненной на основе ПК.

На первом этапе следует связать каждую координатную ось перемещения рабочего органа робота с определенным выходом ПК так, чтобы перемещение по соответствующей координате выполнялось в любой заданный момент по сигналу от ПК. В рассматриваемом примере все рабочие органы робота остаются в заданном положении до тех пор, пока не наступит команда на смену позиции. Для каждой из четырех координат требуются два выхода, поскольку для перемещения по каждой координате в двух направлениях необходимы две команды. Для управления рассматриваемым роботом требуется $2 \cdot 4 = 8$ выходов. Выходам ПК можно присвоить следующие обозначения: Y1 — вращение стойки против часовой стрелки; Y2 — вращение стойки по часовой стрелке; Y3 — перемещение руки вверх; Y4 — перемещение руки вниз; Y5 — выдвижение руки; Y6 — отвод руки; Y7 — захват; Y8 — освобождение.

При идентификации координатных осей робота было использовано условие, что в начале цикла производится позиционирование стойки робота, а затем выдвижение захвата, которое является рабочим движением (см. гл. 7). Все выходы для управления координатными осями робота сгруппированы таким образом, что четные выходы относятся к одному направлению перемещения, а нечетные — к противоположному. Выбор выходов для всех координатных осей произведен

Рис. 12.27

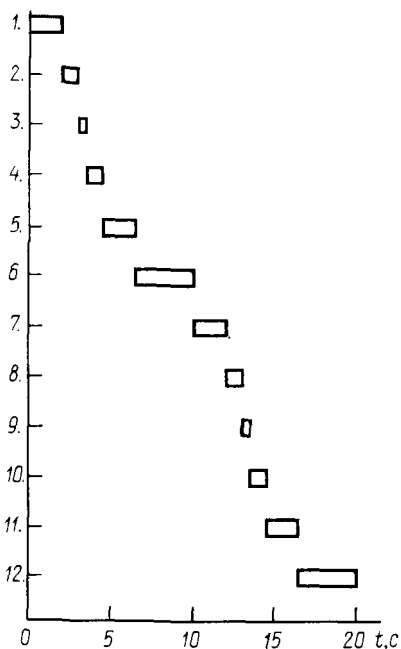
Временная диаграмма для иллюстрации примера 12.4:

1 — подвод руки; 2 — опускание руки; 3 — захват; 4 — подъем руки; 5 — отвод руки; 6 — вращение стойки против часовой стрелки; 7 — подвод руки; 8 — опускание руки; 9 — освобождение захвата; 10 — подъем руки; 11 — отвод руки; 12 — поворот стойки по часовой стрелке

произвольно. Однако при идентификации выходов, соответствующих координатным осям, использованы логические принципы, помогающие запомнить соответствие выходов и координатных осей.

При управлении в функции времени без обратной связи каждая запрограммированная функция практически выполняется удовлетворительно. Единственно, что необходимо делать при управлении в функции времени, это включать определенные выходы в соответствующие моменты времени и оставлять их включенными на заданное время. При этом прекращение перемещения рабочего органа робота производится по окончании выдержки времени или при наезде на механический упор или ограничитель в конце перемещения по соответствующей координате. Барабанный таймер ПК представляет собой идеальное средство для реализации приведенного способа управления в функции времени. Рассмотрим многофункциональную или цикловую диаграмму. На этой диаграмме представлена заданная или реальная последовательность перемещений рабочих органов робота, изображенного на рис. 12.26. Поскольку управление роботом осуществляется без обратной связи, контроль перемещения по каждой координате производится путем задания соответствующего временного периода. Ограничение перемещения по каждой координате контролируется с помощью механического упора. Может потребоваться изменение временных уставок в соответствующих пределах, поскольку скорости перемещений рабочих органов по одним координатам могут быть больше, чем по другим. Для идентичности управления перемещениями целесообразно временные уставки сохранять постоянными. Существует некоторая неопределенность относительно значения временной уставки, обеспечивающей контроль перемещения, поскольку времена перемещений рабочих органов могут меняться в зависимости от давления, силы сопротивления, нагрузки и других факторов. Следовательно, целесообразно немного увеличить временные уставки, с помощью которых контролируются перемещения по каждой координате, чтобы гарантировать выполнение очередной операции до начала следующей. Существует возможность гарантированного *перекрытия* по времени перемещений рабочих органов по всем координатам, заданным в программе робота. Однако пользователи и разработчики должны сами решать, на каких этапах программы робота такие перекрытия по времени могут быть разрешены. Будем считать, что любые перекрытия по времени перемещений рабочих органов недопустимы.

Для составления реальной программы управления роботом следует определить этап, с которого будет начинаться цикл. Обычно его можно выбрать произвольно. Но какой бы этап ни был выбран в качестве первого, после этого следует придерживаться определенной последовательности при программировании остальных элементов цикла. С логической точки зрения начало цикла следует, по-видимому, выбирать в момент, когда производится позиционирование стойки с поворотом по часовой стрелке, при нахождении рабочего органа в равнове-



**12.2. Матрица барабанного таймера ПК для робота
типа взять-положить (к примеру 12.3)**

Номер барабана		Выходы								Количество опросов на импульс = 30 Неисполь- зованные выходы
Шаг	Количество импульсов на шаг	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	
1	4	0	0	0	0	1	0	0	0	
2	2	0	0	0	1	0	0	0	0	
3	1	0	0	0	0	0	0	1	0	
4	2	0	0	1	0	0	0	0	0	
5	4	0	0	0	0	0	1	0	0	
6	7	1	0	0	0	0	0	0	0	
7	4	0	0	0	0	1	0	0	0	
8	2	0	0	0	1	0	0	0	0	
9	1	0	0	0	0	0	0	0	1	
10	2	0	0	1	0	0	0	0	0	
11	4	0	0	0	0	0	1	0	0	
12	7	0	1	0	0	0	0	0	0	
13—16	0	0	0	0	0	0	0	0	0	

шенном состоянии в верхнем отведенном положении, подготовленного к поднятию новой детали, как показано на рис. 12.26.

На рис. 12.27 приведена диаграмма цикла с указанием времени перемещений каждого рабочего органа робота в соответствующей последовательности. При вращении стойки перемещается основная масса робота. Поэтому вследствие высокой инерционности на выполнение указанных перемещений требуется максимальное время. Шаги с минимальной длительностью отработки имеют место при разжиге и закрывании захвата.

При настройке барабанного таймера ПК для завершения программы управления роботом необходимо выбрать количество циклов опроса входов-выходов на импульс. Минимальное время отработки шагов при управлении захватом, как показано на рис. 12.27, составляет 0,5 с. Следовательно, при частоте опроса 60 циклов/с длительность 30 циклов опроса должна быть достаточно малой, чтобы достичь требуемой точности. В табл. 12.2 приведена полная матрица барабана ПК.

Пример 12.5. Логическое управление с помощью ПК роботом типа взять-положить. В примере 12.4 рассмотрен барабанный таймер ПК; система управления является разомкнутой. При использовании управления с обратной связью для обеспечения *обратной связи* используются датчики контроля перемещения рабочих органов робота на каждом шаге. Для контроля заданной координаты позиционирования каждого рабочего органа робота применяются механические упоры или бесконтактные лутевые выключатели, от которых соответствующая информация передается в ПК. Можно запрограммировать ПК таким образом, чтобы не начинать отработку следующего шага до тех пор, пока не будет получено подтверждение о завершении исполнения предыдущего шага. Задержка отработки следующего шага может быть сколь угодно длительной. Следовательно, следует не только идентифицировать переменные выходов Y1—Y8, как это сделано в примере 12.4, но и идентифицировать переменные входов, связанных с каждым выходом. Это необходимо для обеспечения контроля выхода каждого рабочего органа робота с помощью датчиков в заданную координату. Произвольным условием для реализации приведенной задачи будет присвоение обозначения X11 входу, по которому поступает информация от путевого выключателя, который сигнализирует об окончании «вращения стойки против часовой стрелки» (выход Y1); аналогично вход X12 будет связан с входом Y2, X13 — с Y3 и т. д. и, наконец, X18 — с Y8. В табл. 12.3 представлена в табулированном виде пол-

12.3. Этапы цикла для робота типа взять-положить при непосредственном управлении от ПК

Движение рабочего органа робота	Выход ПК	Ограниченные перемещения (вход ПК)	Необходимые предварительные условия (входы ПК)
Подвод руки	У5	X15	X12, X18
Опускание руки	У4	X14	X15, X12, X18
Захват	У7	X17	X14, X12
Подъем руки	У3	X13	X17, X12
Отвод руки	У6	X16	X13, X17, X12
Поворот стойки против часовой стрелки	У1	X11	X16, X17
Подвод руки	У5	X15	X11, X17
Опускание руки	У4	X14	X15, X11, X17
Освобождение захвата	У8	X18	X14, X11
Подъем руки	У3	X13	X18, X11, X15
Отвод руки	У6	X16	X13, X11, X18
Поворот по часовой стрелке	У2	X12	X16, X11, X18

ная схема вместе с необходимыми предварительно назначенными для каждой координаты робота входами. В табл. 12.3 использована та же последовательность перемещений рабочих органов робота, что и в примере 12.4. Однако в этом случае отсутствует барабанный таймер, а также не проводится временной опрос выходов У. Вместо этого сигнал на каждом выходе формируется после того, как на основных входах появляются сигналы, связанные с окончанием отработки предыдущих шагов.

На первый взгляд может показаться, что существует лишь необходимость ввода информации об окончании отработки шага, предшествующего заданному, чтобы начать исполнение последнего. Такой алгоритм был бы справедлив, если бы перемещение рабочих органов по каждой координате имело место *лишь* однажды в запрограммированной последовательности перемещений рабочих органов робота. Однако в настоящем примере каждая ступень логической схемы должна содержать дополнительные логические входы, позволяющие отличить соответствующий шаг программы робота от других аналогичных. Например, на втором шаге в табл. 12.3 производится перемещение руки, при завершении которого на входе X14 появляется соответствующий сигнал. Если бы вход X14 использовался в качестве единственного логического входа для перехода на следующий шаг рассматриваемой программы, обеспечивающей захват, то рука робота *всегда* выполняла бы операцию захвата после завершения движения. Однако на восьмом шаге программы также производится отработка движения, но при этом после отработки этого движения не требуется захвата. Поэтому возникает необходимость использования других соответствующих входов в логической схеме, чтобы обеспечить контроль отработки на каждом этапе программы заданных и *только* заданных шагов. Поэтому для описания «необходимых предварительных исходных условий» для каждого шага, как представлено в табл. 12.3, используются дополнительные входы.

Все ступени логической схемы реализуются *одновременно*. Следовательно, безразлично, в какой последовательности осуществляется вход в ступени схемы в ПК. На рис. 12.28 изображена лестничная логическая схема, соответствующая табл. 12.3. Вход X9 используется для представления замыкающего контакта. С учетом использования входа X9 в качестве необходимого условия в каждой ступени логическая схема может быть получена непосредственно из упомянутой таблицы. Ступени схемы представлены не в последовательном порядке, как строки в табл. 12.3. Фактически для обеспечения логических альтернатив выходам, которые повторяются в управляющей программе робота, были использованы

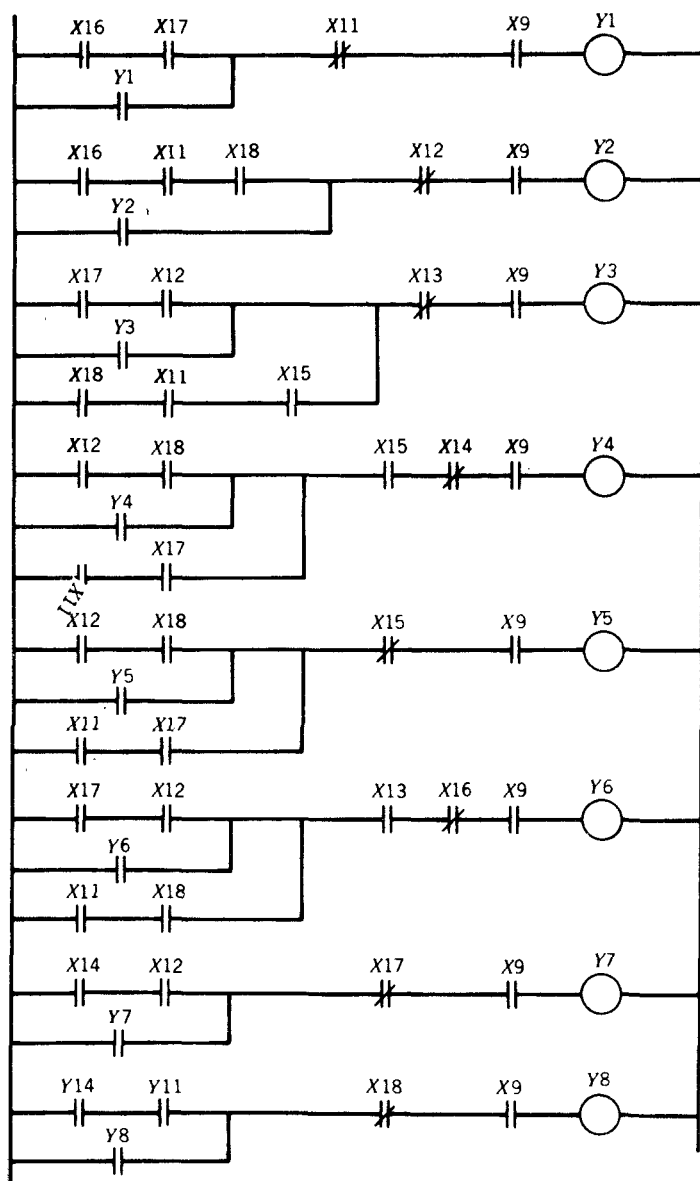


Рис. 12.28

Лестничная логическая схема для работа типа взять-положить (к примеру 12.5)

STR	X16	AND	X9	STR	X17
AND	X17	OUT	Y3	AND	X12
OR	Y1	STR	X12	OR	Y6
AND NOT	X11	AND	X18	STR	X11
AND	X9	OR	Y4	AND	X18
OUT	Y1	STR	X11	OR STR	
STR	X16	AND	X17	AND	X13
AND	X11	OR STR		AND NOT	X16
AND	X18	AND	X15	AND	X9
OR	Y2	AND NOT	X14	OUT	Y6
AND NOT	X12	AND	X9	STR	X14
AND	X9	OUT	Y4	AND	X12
OUT	Y2	STR	X12	OR	Y7
STR	X17	AND	X18	AND NOT	X17
AND	X12	OR	Y5	AND	X9
OR	Y3	STR	X11	OUT	Y7
STR	X18	AND	X17	STR	X14
AND	X11	OR STR		AND	X11
AND	X15	AND NOT	X15	OR	Y8
OR STR		AND	X9	AND NOT	X18
AND NOT	X13	OUT	Y5	AND	X9
				OUT	Y8

Рис. 12.29
Программа для ПК (к примеру 12.5)

логические схемы ИЛИ (параллельные ступени). На рис. 12.29 представлена управляющая программа ПК, с помощью которой реализована рассмотренная выше схема.

12.6. ПРЕИМУЩЕСТВА ПК

До сих пор все системы управления производственным оборудованием, рассмотренные в примерах, приведенных в гл. 10—12, могли быть выполнены с использованием отдельных логических компонентов или реле, описанных в гл. 2. Однако настройка ПК на выполнение тех же функций имеет ряд очевидных преимуществ, на которые целесообразно обратить внимание специалистам по робототехнике и автоматизации.

Гибкость. Наиболее существенным преимуществом ПК в сравнении с обычными релейными схемами является легкость, с которой логика системы может быть изменена, иными словами *гибкость* ПК. Как только происходит смена выпускаемой модели, производится переналадка на новую партию изделий или принимается решение для изменения стратегии управления, программное обеспечение ПК может быть изменено за считанные минуты. При этом не требуется никакого пересоединения проводников или изменения прокладки последних, а также замены каких-либо компонентов технических средств. В этом заключается классическое различие между изменениями в программном обеспечении и аппаратном обеспечении.

Гибкость ПК обуславливает дополнительные преимущества, даже если основные параметры процесса или стратегия управления остаются постоянными. В некоторых случаях целесообразно выполнять точную настройку робототехнических средств или автоматизированного оборудования путем соответствующей наладки в процессе отработки цикла. В этом случае настройка осуществляется более эффективно. Иногда повышение эффективности технологического процесса может быть достигнуто путем некоторого повышения производительности или сокращения простоев технологического оборудования. При использовании непрограммируемых релейных схем подобные изменения в технологическом цикле просто несоизмеримы с затратами. В то же время аналогичные коррекции, вносимые в технологический процесс, легко реализуются при наличии ПК и практически могут повысить производительность технологического процесса до максимального уровня.

Одним из случаев изменения в технологическом процессе является увеличение значения технологического параметра или повышение уровня автоматизации. При возрастании объема производства или повышении уровня сложности автоматизации соответствующего технологического процесса гибкость ПК дает положительный эффект. Однако ПК некоторых моделей в подобных случаях имеют ряд преимуществ по сравнению с другими моделями. При создании робототехнических средств или иного оборудования для автоматизации технологических процессов разработчику следует учитывать перспективы развития системы управления и в соответствующих случаях приобрести ПК модульной конструкции. Такие ПК по мере возрастания требований со стороны технологического процесса обеспечат развитие системы управления.

Ускорение процесса наладки. Хотя ускорение процесса наладки является прямым следствием гибкости ПК, это преимущество следует рассмотреть отдельно. Предположим, что нужно разработать схемы управления производственным оборудованием до создания полной системы управления. Как можно видеть на рис. 12.1, у ПК имеется некоторое количество входов и выходов. Назначение контактов этих входов-выходов осуществляется произвольно, поэтому не имеет значения, как они подключены к электрооборудованию производственных установок. Как правило, программист определяет функциональное назначение входов и выходов, после чего передает соответствующий список распределения функций по входам-выходам инженеру-электрику. Конечно, напряжение и мощность на разных выходах ПК могут варьировать, но эта проблема решается таким же образом, как и в обычных релейных схемах. Обычно выходы ПК рассчитаны на напряжение 110 В переменного тока при небольшой мощности. С помощью таких выходов обеспечивается управление электромагнитами переменного тока, пускателями, управляющими электродвигателями или контакторами, осуществляющими коммутацию силовых це-

пей. Несмотря на то, что мощность выходных цепей мала, напряжение 110 В переменного тока оказывается предпочтительным по сравнению с более низким напряжением постоянного тока, поскольку в первом случае максимально ослабляются помехи, возникающие вследствие электромагнитной индукции от находящихся поблизости мощных источников, установленных в производственном оборудовании. Схемы управления с низким напряжением питания обычно в большей степени подвержены воздействию электрических помех от находящихся рядом производственных установок. Выбор значения напряжения на входах и выходах ПК обусловлен необходимостью максимального снижения уровня таких помех. Входы ПК по указанной причине также могут быть рассчитаны на прием сигналов напряжением 110 В. Однако в ряде случаев могут использоваться входные сигналы при более низком напряжении 24 В постоянного тока. Как только будет выбрано напряжение стандартных входных и выходных сигналов, инженер-электрик может продолжать дальнейшую работу по подключению ПК к производственному оборудованию без каких-либо дополнительных данных.

Следующим аспектом наладки ПК является, безусловно, программирование самого ПК. Как было показано в предыдущем разделе, представление программы ПК в коде не вызывает затруднений и может быть выполнено эффективно и очень быстро любым специалистом, достаточно хорошо разобравшимся в принципах работы систем управления промышленной электроавтоматики, рассмотренных в гл. 10 и 11.

В разделе, посвященном программированию ПК, было показано, что некоторые ПК оснащены удобными пультами с клавиатурой, а конструкция дисплеев предусматривает вывод на экран схем электроавтоматики. Однако даже дешевые ПК, которые оснащены малогабаритным ручным подвесным пультом без дисплея, допускают очень простое программирование в коде логики электроавтоматики. Например, ввод и контроль программы для ПК, представленной на рис. 12.29, могут быть осуществлены достаточно просто в течение 5—10 мин.

Подчеркивая преимущество простоты программирования ПК при наладке в соответствии с требованиями пользователя, не следует упрощать задачу планирования, разработки и реализации системы управления промышленным оборудованием. Процесс программирования может быть прост, однако разработка автоматизированной системы может оказаться достаточно сложной. Инженеры по автоматизации должны уметь в точности определять функции процесса, подлежащего автоматизации, временные параметры и условия его протекания. Реальные схемы электроавтоматики для производственного оборудования отличаются большим разнообразием и сложностью и могут насчитывать десятки входов и выходов и сотни ступеней. Примером здесь может служить система управления специальным роботом, общий вид

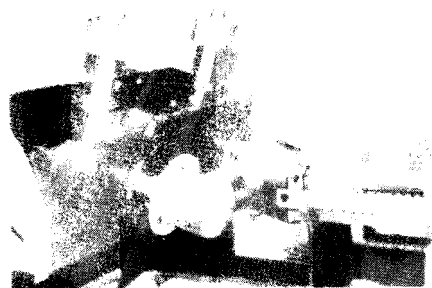
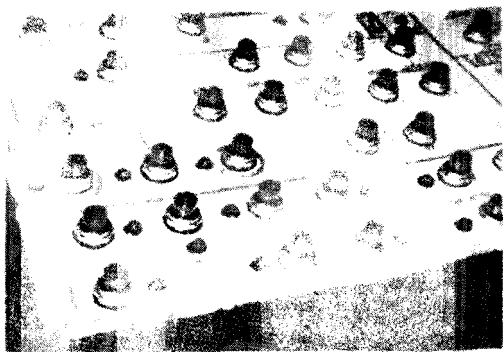


Рис. 12.30
Специализированная роботизированная система управления и транспортирования

представлен на рис. 12.30. Эта система с ограничением координатных перемещений выполнена на основе ПК. Программа обеспечивает управление последовательностью перемещений рабочих органов робота, состоящей из 35 шагов. Лестничная логическая схема включает 200 ступеней. Разработка подобной реальной системы должна проводиться с учетом всех необходимых деталей. В этом случае при определении входов и выходов нельзя допускать несовместимости и неопределенности.

Однако после окончания разработки лестничной логической схемы процесс программирования ПК представляет собой простую задачу, решение которой обычно занимает несколько минут. Последним этапом при наладке системы управления является проверка разработанной программы и корректировка ошибок, допущенных при планировке и разработке системы. Вследствие сложности реальной системы практически неизбежны небольшие коррективы. Поэтому преимущество программных коррекций снова делает ПК более предпочтительным по сравнению с релейными схемами.

Надежность и удобство обслуживания. Отказ станка во всех случаях представляет собой отрицательный фактор. Однако, как следует из гл. 3, последствия таких отказов являются значительно более серьезными при использовании автоматики и робототехники. В отличие от электромеханических реле ПК выполнены на электронных полупроводниковых схемах и таким образом лишены подвижных частей. Результатом являются резко возросшие надежность и долговечность, что практически обуславливает независимость этих параметров от количества циклов, выполненных различными логическими элементами. Эта независимость частоты выхода из строя долговечности электронного полупроводникового оборудования иллюстрируется плоским участком кривой трапециевидальной формы для надежности подобного оборудования. Си-

стемы управления промышленным оборудованием часто используются для управления в функции времени, что налагает на них жесткие требования в части надежности.

Частота отказов системы управления представляет собой лишь часть проблемы при оценке работоспособности или надежности всей системы. Другой частью является удобство обслуживания. При возникновении неисправности процедура ее нахождения в обычной релейной схеме является весьма длительной. При поиске неисправностей такого типа может использоваться вольтметр. Наоборот, при наличии дисплея и клавишного пульта программирования специалист имеет возможность легко произвести контроль различных участков программы ПК на соответствие сигналов и выполнить таким образом проверку логической части. Даже дешевые системы, не оснащенные дисплеем, как правило, предусматривают использование метода контроля сигналов на каждом логическом элементе.

Функция сбора данных. ПК могут располагать множеством преимуществ обычных ЭВМ, предназначенных для управления технологическими процессами. Такие преимущества заключаются в автоматической распечатке в табулированном виде количества деталей, отбраковки по качеству, часов работы и других данных, представляющих интерес для централизованного управления соответствующим производственным оборудованием. Гибкость программирования ПК дает возможность выбора периодического или непрерывного контроля переменных параметров технологического процесса.

Дополнительные входы-выходы. Совершенствование роботов и автоматизированного производственного оборудования, оснащенного шаговыми приводами, предъявляет дополнительные требования к ПК, которые могут обеспечить на определенных выходах последовательности импульсов. С помощью ПК, выполненного на основе полупроводниковых схем, значительно эффективнее по сравнению с обычными релейными схемами могут быть организованы импульсные выходы. К дополнительным выходам ПК также могут быть подключены дисплеи и печатающие устройства, хотя такое усложнение обычно связывается с ЭВМ, предназначенными для управления технологическими процессами. Особенности ЭВМ будут рассмотрены в гл. 13. По мере возрастания сложности ПК становится все меньше отличий последнего от универсальной ЭВМ для управления технологическим оборудованием. Подобно управляющим ЭВМ ПК может подключаться к центральной ЭВМ для обеспечения централизованного управления заводом от ЭВМ.

Стоимость. Для простейших логических систем, включающих всего несколько входных контактов, обычные релейные схемы могут оказаться более экономичными по сравнению с ПК, по крайней мере с точки зрения первоначальных затрат. При увеличении сложности системы управления приблизительно до десяти

реле и более применение ПК становится более экономичным. В случае очень сложных логических систем сравнение стоимости ПК и релейных схем становится уже нецелесообразным.

Кроме первоначальных затрат следует учесть компактность конструкции ПК, применение которых может существенно сократить необходимые производственные площади и массу, что имеет большое значение для малогабаритных автоматизированных производственных систем и роботов. Также может быть учтена эффективность ПК с энергетической точки зрения, поскольку полупроводниковые приборы ПК требуют меньшего количества энергии по сравнению с обычными релейными схемами.

12.7. НЕДОСТАТКИ ПК

Наряду с такими преимуществами, как широкие функциональные возможности, привлекательный внешний вид, гибкость и удобство в эксплуатации, ПК имеют ряд недостатков, которые рассмотрены ниже. Простота программирования и гибкость ПК приводят к большим ошибкам в программах чаще, чем в хорошо продуманных релейных схемах, реализуемых со значительными трудовыми затратами. При легкости внесения коррекций в систему будет вводиться большее количество изменений. Наряду с коррекциями, улучшающими характеристики системы, также могут вноситься коррекции, ухудшающие ее характеристики.

Для разработчиков систем управления на основе ПК рекомендуется устанавливать реле защиты по питанию, обеспечивающее отключение всех выходов ПК при перебросе выключателя из одного состояния в другое. При этом настоятельно рекомендуется, чтобы такие реле защиты по питанию были электромеханической конструкции, а не полупроводниковыми. Электронные устройства отличаются очень высокой надежностью, однако введение дополнительного реле защиты по питанию, отличающегося принципиально иной конструкцией, повышает помехозащищенность ПК. Это объясняется тем, что на электромеханические реле оказывают влияние помехи иного типа, чем помехи, влияющие на электронную часть ПК. Безусловно, электромеханические реле вследствие различных причин выходят из строя, что также следует принимать во внимание при разработке обычных релейных схем. Влияние физических факторов на работоспособность электронных систем будет рассмотрено в гл. 14.

В ряде случаев внезапное отключение питания на некотором выходе может создать более серьезную аварийную ситуацию, чем, наоборот, появление ненужного сигнала. Программист, обслуживающий ПК, должен с особой тщательностью программировать выходы, с тем чтобы учесть возможность возникновения редкого неблагоприятного сочетания некоторых факторов. Большинство ПК располагают возможностью *независимого* включения или выключения сигнала на некотором выходе, несмотря на состояние

запрограммированной логической схемы. Независимое управление выходами может также использоваться для диагностирования. Использование независимо управляемых выходов дает возможность пользователю ПК избежать нежелательных последствий неисправностей.

В ы в о д ы

ПК представляет собой простое устройство, которое в значительной степени упрощает разработку сложных систем управления, реализация которых ранее производилась на основе электро-механических реле. Одновременно с указанным преимуществом ПК отличаются значительно большей гибкостью и позволяют уменьшить объем технических средств, а также снизить их стоимость.

Некоторые устройства со сходными функциями и конструкцией имеют одну и ту же аббревиатуру ПК. Программируемые контроллеры и персональные компьютеры сходны по конструкции, поскольку и те и другие включают центральный процессор, а также запоминающие устройства, применяемые в ЭВМ. Однако ПК обеспечивают быстрое исполнение одних и тех же программ с повторением их несколько тысяч раз в час. Наоборот, персональный компьютер или универсальная ЭВМ обеспечивает исполнение разнообразных сравнительно больших программ. Кроме того, входы и выходы ПК предназначены для управления производственным оборудованием, в то время как входы и выходы универсальных ЭВМ используются для ввода и вывода данных. При реализации логических схем управления производственным оборудованием в большинстве случаев требуется очень высокое быстродействие, которое обеспечивается ПК. Поэтому ПК могут использоваться при реализации систем управления как непрерывного, так и дискретного типов.

Основой для программирования ПК служит лестничная логическая схема. Устройство для программирования является автономным с возможностью временного подключения к ПК. Такое исполнение устройства для программирования объясняется тем, что после ввода запрограммированной лестничной схемы функционирование ПК будет продолжаться до тех пор, пока не потребуются проведения изменений. Счетчики, таймеры и даже кулачковые или барабанные таймеры могут быть реализованы путем перепрограммирования резидентного программного обеспечения в ПК.

ПК широко применяются для управления промышленными роботами как в автономном режиме, так и в составе роботизированных комплексов. ПК представляют собой идеальное средство управления роботизированным производственным модулем. В этом случае в функции ПК входит синхронизация координатных перемещений рабочих органов робота с обслуживаемыми станками. ПК отличаются столь высокой гибкостью, что они могут функционировать на уровне интеллектуального устройства управления

роботом и позволяют инженеру по автоматизации создать промышленный робот непосредственно из основных компонентов средств автоматизации, рассмотренных в гл. 2.

В гл. 12 приведены примеры, иллюстрирующие принцип управления промышленными роботами с помощью ПК в соответствии с двумя концепциями: с обратной и без обратной связи. Управление без обратной связи предусматривает использование таймеров. Обычно при этом типе управления применяются барабанные таймеры. Управление с обратной связью предполагает наличие датчиков, связанных с соответствующими входами ПК. Датчиками формируются сигналы об окончании отработки шагов производственного процесса.

Преимущества, обусловленные гибкостью, простотой наладки, контролем системы, надежностью, удобством обслуживания и стоимостью, становятся очевидными для каждого изучающего основы работы ПК. Не столь очевидны некоторые недостатки ПК. Несмотря на надежность ПК и их полупроводниковых схем, во многих случаях нецелесообразно полностью полагаться на электронные средства. Применение реле защиты по питанию электромеханической конструкции позволяет обеспечить прерывание питания на всех выходах в случае возникновения редких неисправностей электронной системы управления.

Упражнения и вопросы

12.1. Аббревиатура ПК имеет несколько значений для специалиста по автоматизации. Назовите по крайней мере три из этих значений.

12.2. Является ли ПК компьютером? Объясните, в какой степени ПК может быть отнесен к ЭВМ и в какой нет.

12.3. Что понимается под термином *цикл работы ПК*? Какова длительность типового цикла ПК?

12.4. Сравните преимущества двух принципиальных способов управления с помощью ПК: управление с обратной и без обратной связи. Какой тип управления предусматривает использование таймеров?

12.5. Разработайте лестничную схему и напишите программу для ПК с целью обеспечения автоматического включения и отключения предупредительной световой сигнализации с частотой 1 Гц.

12.6. Разработайте лестничную схему и напишите программу для ПК, обеспечивающую автоматическую работу заводской системы отопления ежедневно с 7 ч утра до 6 ч вечера. Для решения такой задачи существуют по крайней мере три различных метода: с использованием таймеров, таймеров и счетчиков, а также барабанного таймера.

12.7. Разработайте лестничную схему для автоматического включения заводской системы отопления с 6 ч утра до 9 ч вечера только в *рабочие дни*; в выходные дни система отопления должна отключаться.

12.8. Промышленный робот с ограничением по координатным перемещениям, управляемый от ПК, обслуживает автоматизированную производственную установку с двухшаговым циклом. Цикл представляет собой последовательность шагов, отработка которых осуществляется с помощью выходных сигналов ПК. Окончание отработки контролируется с помощью соответствующих входных сигналов. Разработайте лестничную схему и программу для ПК с использованием обратной связи для контроля окончания отработки каждого шага в соответствии с программой перед началом отработки следующего шага. При разработке учтите предположение, что рабочий орган по каждой координатной оси

робота будет находиться в соответствующей координате до тех пор, пока на входе в ПК не появится команда на выполнение перемещения рабочего органа в заданную координату.

	<i>Выход ПК</i>	<i>Датчик контроля перемещения</i>
Зажим	<i>Y8</i>	<i>X18</i>
Подъем	<i>Y4</i>	<i>X14</i>
Подвод	<i>Y6</i>	<i>X16</i>
Пауза	10 с	
Отвод	<i>Y5</i>	<i>X15</i>
Поворот стойки против часовой стрелки	<i>Y2</i>	<i>X12</i>
Подвод	<i>Y6</i>	<i>X16</i>
Пауза	6,5 с	
Отвод	<i>Y5</i>	<i>X15</i>
Опускание	<i>Y3</i>	<i>X13</i>
Разжим захвата	<i>Y7</i>	<i>X17</i>
Поворот стойки по часовой стрелке	<i>Y1</i>	<i>X11</i>

12.9. При выполнении упражнения 12.8 используйте принцип управления без обратной связи с барабанным таймером; на завершение перемещения каждого рабочего органа робота отводится 2 с. Составьте матрицу барабанного таймера. Программу для ПК составлять не надо.

12.10. Автономный промышленный робот используется для загрузки-разгрузки четырех различных станков по запросу. Промышленный робот оснащен автономным устройством управления, однако в качестве устройства централизованного управления роботизированным комплексом используется ПК. Во время работы всех четырех станков рабочий орган робота ожидает в заданной позиции, которая контролируется входом *X10*. Как только на станке завершается цикл обработки, на робот автоматически поступают соответствующие сигналы на разгрузку и загрузку. Первый сигнал от станка поступает на вход *X11*, второй сигнал от станка — на вход *X12*, третий сигнал от станка — на вход *X13* и четвертый сигнал от станка — на вход *X14*. Если робот не занят обслуживанием другого станка, то обслуживание станка, от которого поступил запрос, осуществляется без задержки, после чего производится повторный пуск станка и возврат рабочих органов робота в исходное положение. Управление всей работой робота по обслуживанию станка выполняется с помощью ПК, однако, если робот занят обслуживанием другого станка, то обслуживание станка, от которого поступил соответствующий запрос, задерживается. Если ситуация складывается таким образом, что освобождения робота ожидают два-три станка, то не имеет значения, какой из ожидаемых станков будет обслужен первым. При выдаче команды на обслуживание роботом станка ПК обеспечивает наличие сигнала на определенном выходе: *Y1* — для первого станка; *Y2* — для второго станка и т. д. Сигнал на каждом выходе *Yi* поддерживается до тех пор, пока не снимется с входа *X1i* сигнал запроса на обслуживание соответствующего станка. Разработайте лестничную схему и напишите программу для централизованного управления роботом с помощью ПК.

12.11. В системе роботизированного обслуживания четырех станков из упражнения 12.10 ввести возможность ручной загрузки и разгрузки станка. Как только станок будет вручную загружен и будет повторно произведен его пуск, необходимость повторного обслуживания станка с помощью робота в данном цикле отпадает. Разработайте лестничную схему и напишите программу для ПК с учетом ручной загрузки станка в сочетании с роботизированной загрузкой в автоматическом режиме.

12.12. Определите факторы риска, связанные с безопасностью, которые возникают вследствие разрешения использования ручной загрузки станков при их обслуживании с помощью автоматической роботизированной системы. Какие блокировки можно было бы ввести при разработке лестничной схемы для ПК с целью предотвращения обнаруженных факторов риска?

13. ЭВМ ДЛЯ УПРАВЛЕНИЯ В РЕАЛЬНОМ ВРЕМЕНИ

Два-три десятилетия назад первые ЭВМ были однопроцессорными. Программисты подготавливали программы с учетом их последовательной обработки, когда в определенный момент времени обрабатывалась только одна программа. В каждой программе также последовательно с помощью центрального процессора ЭВМ выполнялись отдельные команды. Время исполнения программ было невелико, обычно в пределах нескольких секунд или нескольких минут. Однако во время обработки одной программы ЭВМ не может использоваться для решения других задач.

Современные ЭВМ также производят обработку программ в пакетном режиме. Такие программы записаны на языках, которые применялись ранее, например Бейсик, Фортран и Кобол. Однако обеспечение техническими средствами и организация выполнения задач в современных ЭВМ подверглись изменениям. В настоящее время еще существует небольшое количество однопроцессорных ЭВМ. Последние отличаются малыми размерами и невысокой стоимостью. Большие универсальные (оформленные в виде стойки) ЭВМ в настоящее время никогда не предназначаются для реализации одной непрерывной последовательности программ.

Прежде чем приступить к дальнейшему рассмотрению материала, следует предположить, что читатель до некоторой степени знаком с программированием на ЭВМ в пакетном режиме и что большинство читателей настоящей главы возможно уже имеют опыт программирования на ЭВМ на языке Бейсик, Фортран или другом ориентированном на пользователя языке, рассчитанном на последовательную обработку программ. Однако читатели, не являющиеся программистами, не должны чувствовать себя дилетантами в этом вопросе, поскольку в действительности настоящая глава, как и книга в целом, посвящена не программированию на ЭВМ, по крайней мере не программированию на языках Бейсик и Фортран. Предметом рассмотрения в настоящей главе является современная ЭВМ, функционирующая в реальном времени. Такая ЭВМ обеспечивает практически одновременный контроль и управление большим количеством роботов или автоматизированных производственных установок. Программисту,

имеющему опыт программирования на языке Бейсик или Фортран, может показаться странным, что ЭВМ может контролировать или управлять большим количеством производственных установок одновременно.

В дальнейшем при изложении материала ключевым будет слово «одновременно». Современные ЭВМ все еще исполняют в один и тот же момент времени лишь одну кодированную команду. Однако быстродействие таких ЭВМ дает возможность совершать быстрые переходы от программы к программе, используя *режим разделения времени*. Это позволяет производить одновременный контроль нескольких производственных установок.

Даже однопроцессорные ЭВМ 50-х и 60-х гг. имели некоторые возможности для контроля состояния своего периферийного оборудования. При этом было важно обеспечить прерывание обработки программы и исполнить диагностические подпрограммы при возникновении механической неисправности в приводе устройства считывания с перфоленты, дисководе или, в худшем случае, в самом центральном процессоре. Такая возможность самодиагностики в развитом виде применяется в современной робототехнике и при автоматизации производства. Если возможна организация прерывания однопроцессорной ЭВМ при создании аварийной ситуации в основном оборудовании, то имеется аналогичная возможность прерывания для решения задач с высшим приоритетом. Эта идея привела к разработке (в начале 60-х гг.) концепции работы ЭВМ в режиме разделения времени. В то время основной областью применения ЭВМ все еще оставалась обработка данных. Однако с появлением возможности работы в режиме разделения времени любая программа, находящаяся в режиме длительной отработки, может быть прервана на несколько миллисекунд. Этот интервал времени необходим для полного исполнения небольшой программы, которая введена с клавиатуры терминального устройства в ЭВМ. Концепция разделения времени практически исключает ожидание программистом отработки небольшой программы при отработке большого пакета программ. Центральный процессор может обеспечить отработку большого количества таких небольших программ. При этом у пользователя создается впечатление, что ЭВМ используется исключительно им. Такое впечатление особенно правдоподобно в случае, когда основной задачей ЭВМ является реализация небольших программ, вводимых с удаленных терминалов. Среди первых таких систем, использующих режим разделения времени, были системы резервирования, функционирующие в реальном времени, распространенные на авиалиниях в 60-х гг.

Режим разделения времени применялся для эффективного управления программами разного объема и приоритета, предназначенными для последовательной обработки данных. Следующим новым шагом было объединение концепции разделения времени с концепцией контроля обеспечения техническими сред-

ствами ЭВМ. Если работа ЭВМ может быть прервана при обнаружении неисправности в технических средствах ЭВМ, то аналогичное прерывание может быть обеспечено автоматически при возникновении соответствующих внешних условий, создающих аварийную ситуацию. Во время прерывания информация об условиях может быть выведена на распечатку или на звуковую аварийную сигнализацию; также может быть разрешено продолжение автоматического процесса на следующем этапе при условии передачи сигналов автоматической коррекции на производственную установку. Высокое быстродействие такой ЭВМ позволяет улучшить динамические характеристики управляемого производственного оборудования путем организации замкнутой системы управления непосредственно от ЭВМ. Дополнительное преимущество заключается в том, что ЭВМ может производить сложные расчеты на основе технологических данных, поступающих в ЭВМ от датчиков производственного оборудования. Это позволяет осуществлять коррекции технологического процесса в реальном масштабе времени с большей эффективностью по сравнению с оператором-станочником.

Таким образом, кроме использования в научной области и для обработки данных, ЭВМ стала важным инструментом для автоматизации производства. Возможно представление управляющей ЭВМ в качестве интеллектуального робота, поскольку она обладает некоторыми антропоморфными характеристиками и выполняет некоторые операции, которые прежде выполнялись только людьми.

13.1. УРОВНИ РЕАЛИЗАЦИИ

Для оценки различных уровней или ступеней иерархии, на которых возможно использование ЭВМ для целей автоматизации и управления производственными процессами, остановимся на неавтоматизированном процессе, для управления которым не предусмотрено использования ЭВМ (рис. 13.1). В этом случае оператор контролирует протекание процесса визуально с помощью самописцев, счетчиков и сигнализаторов. Управление этим процессом осуществляется с помощью выключателей и других органов, находящихся на пульте оператора. Информационный поток проходит через органы управления технологическим процессом, поскольку без использования управляющей ЭВМ оператор должен обладать более высокой квалификацией в технологии и управлении технологическим процессом. Кроме того, оператор должен сам вести журнал регистрации работы оборудования, хотя данные, заносимые в этот журнал, могут быть получены из графиков, воспроизводимых самописцами.

Обработка данных не в реальном времени. Как показано на рис. 13.2, ЭВМ, предназначенная для обработки данных, может иметь некоторые преимущества по сравнению с оператором при управлении технологическим процессом с помощью упрощенного

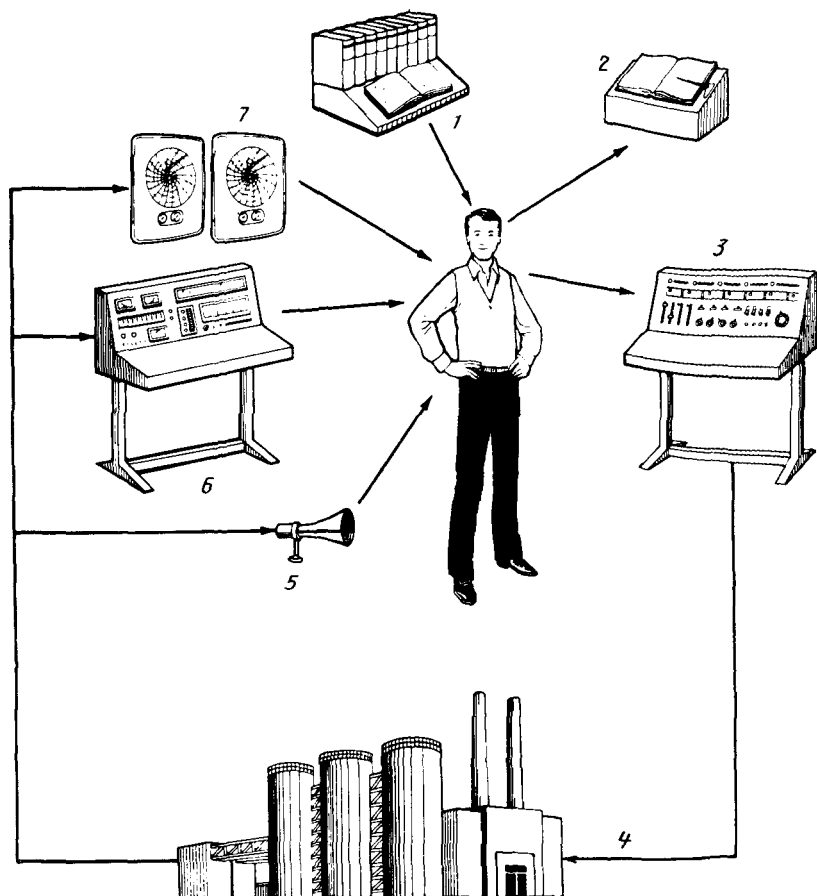


Рис. 13.1

Управление неавтоматизированным технологическим процессом:
 1 — руководства по эксплуатации; 2 — регистрационный журнал; 3 — пульт управления; 4 — технологический процесс; 5 — звуковая сигнализация; 6 — индикация и сигнализация; 7 — регистрирующие приборы

анализа технологических параметров, значения которых получены визуальным путем или путем измерения. При обработке данных не в реальном масштабе времени возможен статистический контроль качества и проведение исследований оптимизации процесса. Однако в концепции обработки данных не в реальном масштабе времени имеется существенный недостаток — временная задержка. К тому времени, когда будут произведены преобразование зарегистрированных данных в цифровой вид и их последовательная обработка с помощью ЭВМ, вносить оператору коррекции в технологический процесс окажется слишком поздно. Подобный режим управления производственными процессами получил широкое распространение в то время, когда еще в основном при-

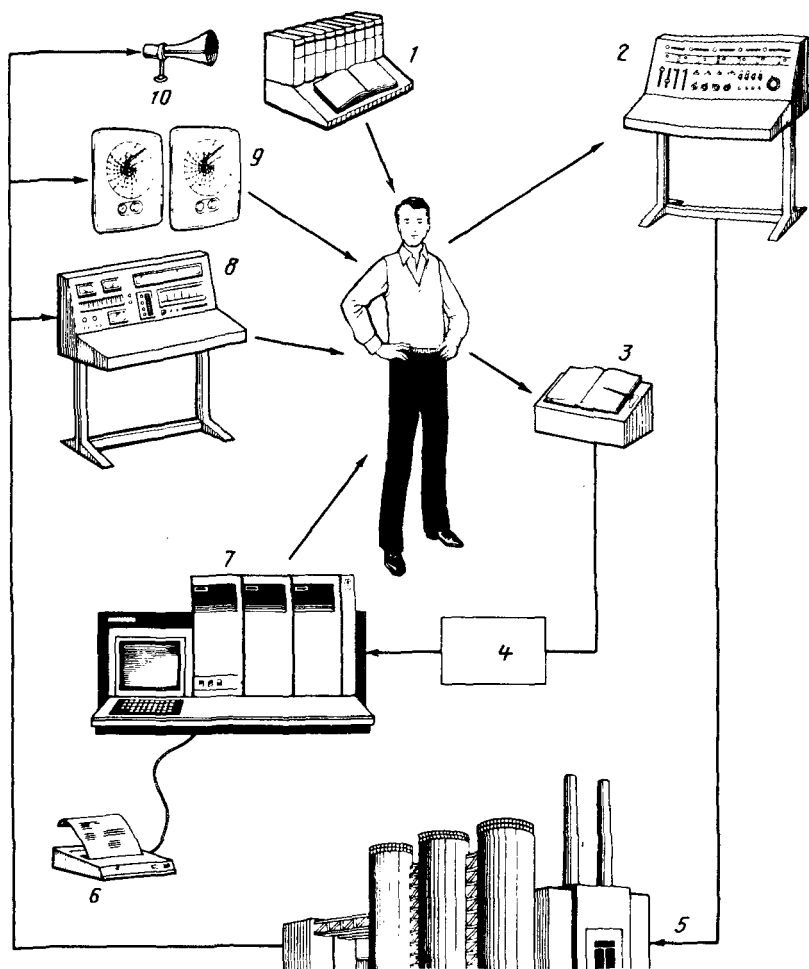


Рис. 13.2

Управление технологическим процессом с помощью ЭВМ не в реальном времени: 1 — руководства по эксплуатации; 2 — пульт управления; 3 — регистрационный журнал; 4 — устройство ввода данных; 5 — технологический процесс; 6 — печатающее устройство; 7 — ЭВМ, функционирующая не в реальном времени; 8 — индикация и сигнализация; 9 — регистрирующие приборы; 10 — звуковая сигнализация

менялись большие ЭВМ в режиме последовательной обработки данных. В настоящее время такой подход считается устаревшим, однако концепция управления с помощью ЭВМ не в реальном времени до сих пор используется на ряде заводов. Преимуществом такой концепции является отсутствие необходимости в использовании дорогостоящих датчиков и технических средств интерфейсов для связи производственного оборудования с ЭВМ.

Контроль технологического процесса. Следующим шагом после исследования данных не в реальном масштабе времени является

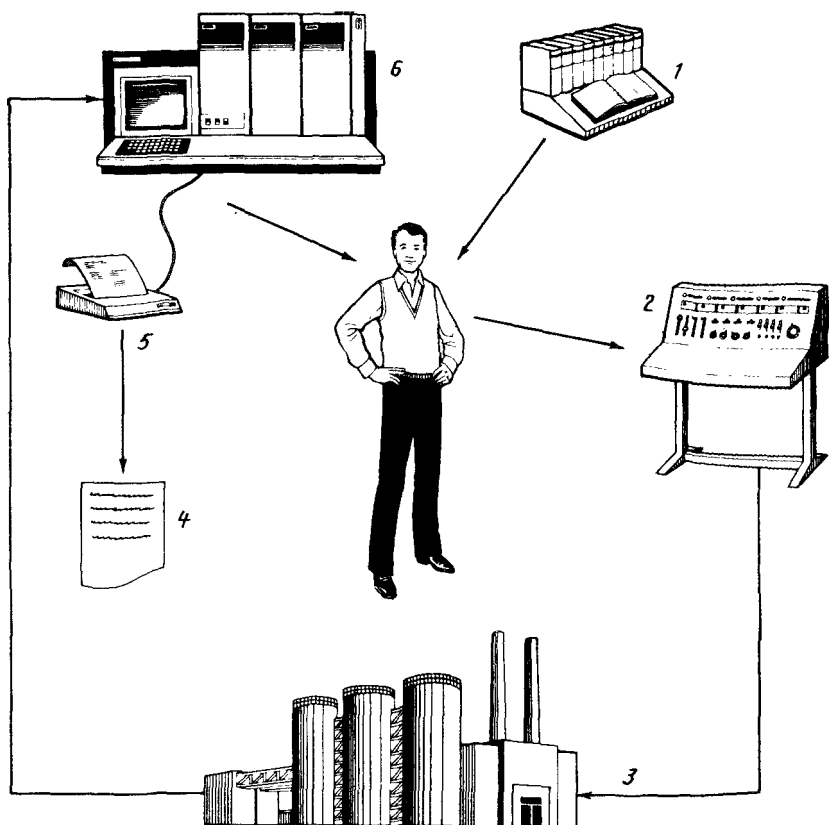


Рис. 13.3

Контроль технологического процесса с помощью ЭВМ в реальном времени:
 1 — руководства по эксплуатации; 2 — пульт управления; 3 — технологический процесс; 4 — распечатка контрольных данных; 5 — печатающее устройство; 6 — ЭВМ, функционирующая в реальном времени

контроль технологического процесса (рис. 13.3). ЭВМ связана с определенными датчиками контроля технологического процесса или счетчиками. Обработка соответствующих данных позволяет ЭВМ следить за ходом технологического процесса. Во время контроля технологического процесса ЭВМ не может непосредственно воздействовать на производственное оборудование и произвести при необходимости соответствующие коррекции технологических параметров. Отсутствие такой возможности объясняется тем, что ЭВМ в этом случае не имеет выходов, воздействующих на управляемое производственное оборудование. Целью такой концепции автоматизации является периодическое информирование административного и обслуживающего персонала о ходе конкретного технологического процесса. При этом оператор освобождается от необходимости внесения данных о ходе технологического про-

цесса в журнал регистрации. Кроме формирования периодических сообщений системы контроля технологического процесса с помощью ЭВМ могут также выдавать без задержки сообщения об изменении технологических параметров операторам или обслуживающему персоналу. В число таких контролируемых параметров может входить, например, температура гидробака. Однако оператор в этом случае все еще должен обеспечивать непосредственное управление производственной установкой. Он также должен обладать квалификацией технолога, достаточной для проведения коррекций технологических параметров в зависимости от «температуры гидробака» или другой технологической информации, полученной с помощью ЭВМ от датчиков контроля технологического процесса.

Контроль с помощью ЭВМ в реальном времени с диалоговой поддержкой. Следующим шагом при автоматизации производственного оборудования является контроль с помощью ЭВМ в реальном масштабе времени с диалоговой поддержкой. При этом виде контроля ЭВМ обеспечивает не только непосредственный контроль протекания технологического процесса, но и выдает конкретные рекомендации оператору для оптимального управления производственной установкой. Например, в то время как ЭВМ, контролирующая технологический процесс, может регистрировать температуру гидробака и периодически выводить для оператора на распечатку справочную информацию, ЭВМ с диалоговой поддержкой, функционирующая в реальном времени, может обеспечить вывод на экран дисплея следующее сообщение: «Температура гидробака 187°, что превышает допустимое значение; для уменьшения подачи топлива к печам закрыть клапан K9 на $\frac{1}{4}$ оборота».

Формирование подобного сообщения значительно повышает степень сложности системы по сравнению с обычной системой контроля технологического процесса. Однако система с указанной возможностью относится к замкнутому типу. При этом ответственность за управление технологическим процессом полностью лежит на операторе, который использует руководство по эксплуатации и за которым остается принятие окончательного решения с целью коррекции заданных технологических параметров.

На рис. 13.4 представлена схема, иллюстрирующая режим контроля с диалоговой поддержкой технологического процесса в реальном времени с помощью управляющей ЭВМ. Следует отметить, что в этом случае оператором не используются руководства по эксплуатации. Возможно, что наиболее существенным преимуществом контроля с диалоговой поддержкой по сравнению с контролем без диалоговой поддержки является хранение руководств по эксплуатации в кодированном виде в памяти ЭВМ. При этом исключается необходимость использования оператором руководства по эксплуатации, представляемого в печатном виде. Руководства, представляемые в печатном виде, вполне пригодны

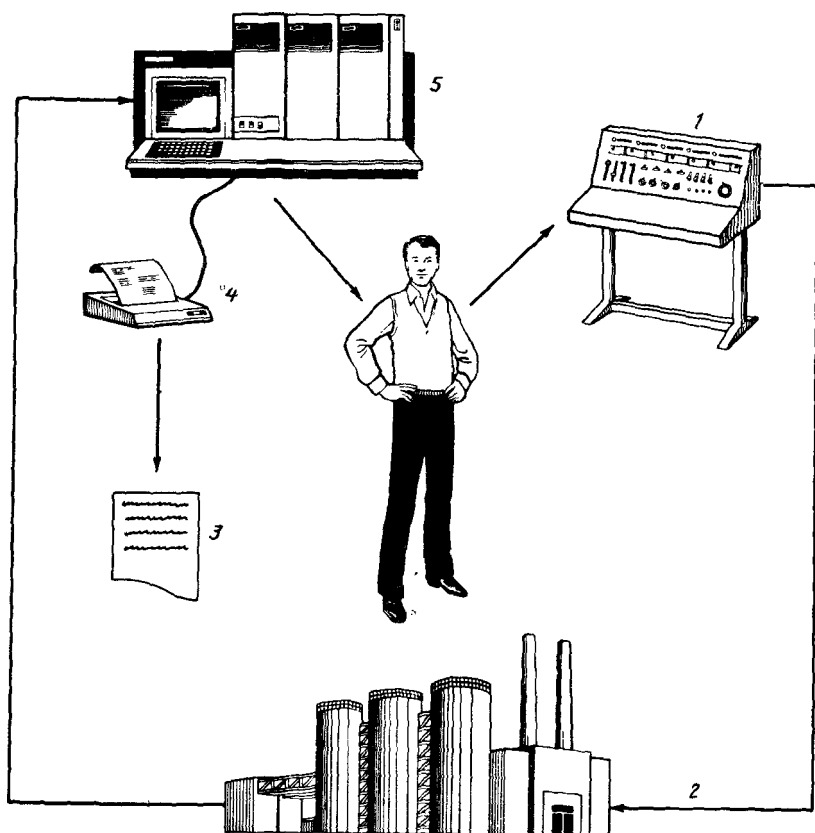


Рис. 13.4

Контроль технологического процесса с помощью ЭВМ в реальном времени с диалоговой поддержкой:

1 — пульт управления; 2 — технологический процесс; 3 — распечатка контрольных данных; 4 — печатающее устройство; 5 — ЭВМ, функционирующая в реальном времени

в качестве документации. Однако при работе руководства в таком виде, как правило, не позволяют достаточно быстро находить соответствующие инструкции. Обычно операторы вынуждены запоминать существенные выдержки для управления технологическим процессом, в противном случае они лишаются возможности достаточно быстрого вмешательства в процесс с целью коррекции технологических параметров при возникновении соответствующих условий. Например, нельзя допустить неопытного оператора к управлению краном с использованием руководства по эксплуатации для решения возникающих проблем без предварительного прохождения курса обучения. Таким образом, хотя руководства по эксплуатации при использовании концепции контроля в реальном времени с диалоговой поддержкой хранятся в памяти ЭВМ и доступны для оператора, тем не менее эти руководства могут не

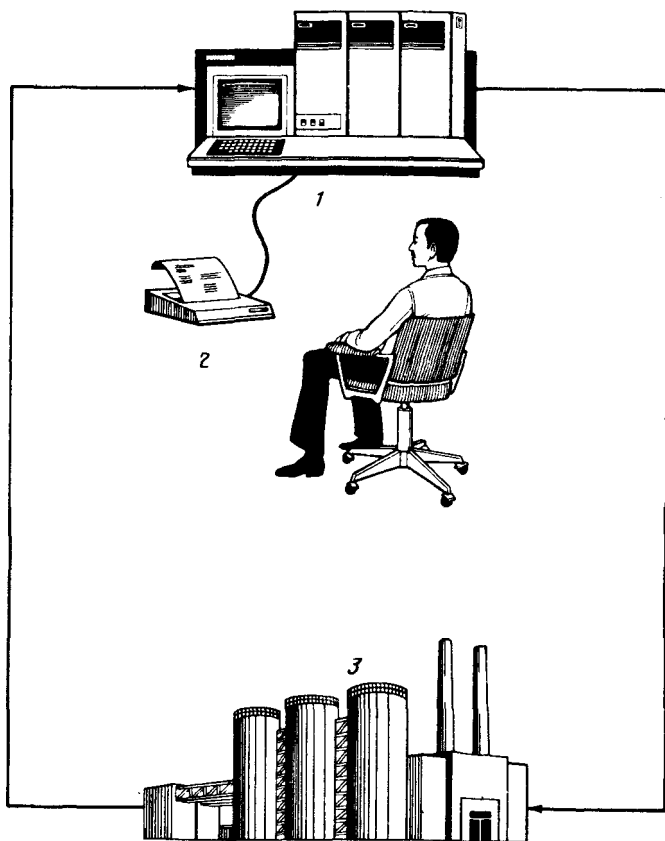


Рис. 13.5

Управление технологическим процессом от ЭВМ с обратной связью:
1 — ЭВМ, функционирующая в реальном времени; 2 — технологический процесс; 3 — печатающее устройство

использоваться в повседневной работе, в то время как от обслуживающего персонала в этом случае не требуется большого опыта и квалификации технолога. Это является существенным преимуществом при использовании концепции в реальном времени с диалоговой поддержкой.

Управление с обратной связью. Высший уровень автоматизации производства с помощью ЭВМ в реальном масштабе времени соответствует управлению с обратной связью (рис. 13.5). Следует обратить внимание на то, что на этом рисунке сидящий оператор не принимает участия в управлении замкнутым технологическим процессом. Все коррекции технологических параметров в этом случае выполняются с помощью обратной связи. Оператор может оставаться на своем месте для устранения нештатных и непредвиденных ситуаций, поскольку полная автоматизация технологического процесса с учетом исключения влияния любых помех, кото-

рые могут возникать в процессе работы, достигается лишь в редких случаях.

Для каждой функции управления технологическим процессом от ЭВМ могут быть применены аналогии действий оператора. Функция контроля технологического процесса может быть сравнима с действиями специалиста или научного сотрудника, который занимается сбором данных. Функция контроля в реальном времени с диалоговой поддержкой аналогична роли консультанта из числа вспомогательного персонала. Такой консультант может давать советы и выдавать рекомендации обслуживающему персоналу, не располагая, однако, правом непосредственного вмешательства в производственный процесс. Функцию замкнутого управления можно сравнить с действиями оператора, выполняющего ряд последовательных операций. Обслуживающий персонал, как и управляющая ЭВМ, вносит коррективы в управляемый технологический процесс. Теперь подробнее рассмотрим характеристики технических средств ЭВМ, предназначенных для управления технологическим процессом.

13.2. КОНЦЕПЦИИ УПРАВЛЕНИЯ

Использование ЭВМ для управления производственным оборудованием с обратной связью в реальном времени позволяет реализовать сложные концепции управления оборудованием с переменными параметрами. При этом не требуется затрат на технические средства, реализующие обычное аналоговое управление для каждого контура управления. Целью таких систем управления, предназначенных для технологического процесса с непрерывно изменяющимися параметрами, является коррекция технологических параметров относительно заданного уровня на основе данных измерений. В производственных условиях существует множество помех от внешних источников, которые могут обусловить отклонение технологических параметров от заданных значений. Кроме того, оператор или специалист из числа обслуживающего персонала может внезапно изменить значение уставки с помощью переключателя или другого органа управления. В этом случае в функцию системы управления технологическим процессом входит обеспечение коррекций технологических параметров в соответствии с новым значением заданного параметра. Любое отклонение параметра от заданного в теории управления называется «ошибкой» рассогласования. Ошибка может рассматриваться как положительная в случае превышения значением технологического параметра значения заданной уставки и отрицательной, если значение технологического параметра меньше значения заданной уставки. *

* Известно, что значение знака ошибки рассогласования в теории управления противоположно значению знака ошибки, принятому в производственной практике. Однако значение знака, принятое в данной книге, делает для читателя, не знакомого с теорией управления с обратной связью, более понятными графические зависимости.

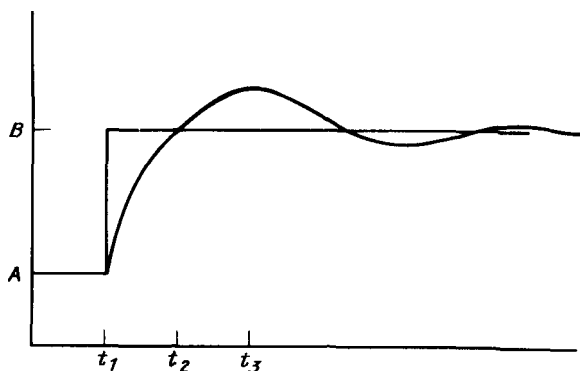


Рис. 13.6

Реакция устройства управления на основе ЭВМ роботом на внезапное изменение координаты плеча робота из координаты A в координату B:

1 — координата плеча робота; 2 — заданная координата; 3 — фактическая координата

На рис. 13.6 представлена временная зависимость положения плеча робота относительно заданной уставки. До момента t_1 , когда отсутствует возмущающее воздействие, положение плеча робота совпадает с заданной координатой. Внезапное изменение заданной координаты рабочего органа робота приводит к большой отрицательной ошибке рассогласования. Если бы система управления была идеальной, она обеспечила бы переход от одной координаты к заданной без ошибки рассогласования. Однако в действительности переход рабочего органа не может быть осуществлен, в особенности в случае перемещения механической руки крупногабаритного промышленного робота, оснащенного гидроприводом. Ошибка рассогласования через короткое время в момент t_2 уменьшается до нуля. Однако в этот момент невозможно остановить движение. Поэтому возникает перерегулирование, достигающее максимума в момент t_3 . Во время перерегулирования система управления определяет, что знак ошибки рассогласования сменился на положительный, и соответственно корректирует ошибку. В конце концов колебания при перерегулировании уменьшаются до нуля, при этом рабочий орган выходит в заданную координату.

Желательно, чтобы разрабатываемые системы управления быстро компенсировали ошибку рассогласования, поскольку, чем быстрее будет достигнута заданная координата, тем скорее рабочий орган достигнет нового объекта. Однако здесь существует противоречие, поскольку быстрая компенсация ошибки приводит к перерегулированию. Перерегулирование может быть столь велико, что может обусловить колебания системы в широком диапазоне между циклами перерегулирования. При этом ряд последовательных циклов перерегулирования могут даже увеличить, а не уменьшить полное время позиционирования. О подобной системе говорят, что она «неустойчива». Средством повышения

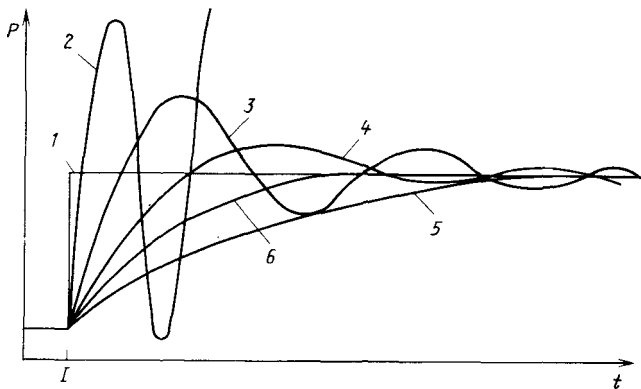


Рис. 13.7

Регулировочные характеристики $P(t)$ систем управления с различным быстродействием:

1 — идеальное состояние системы; 2 — неустойчивое состояние системы; 3 — коэффициент затухания 0,5; 4 — коэффициент затухания 0,25; 5 — чрезмерное демпфирование; 6 — критическое демпфирование; I — изменение уставок; коэффициент демпфирования равен отношению амплитуд последовательных циклов перерегулирования; критическое демпфирование — демпфирование, достаточное для предотвращения любого перерегулирования

устойчивости в этом случае является «демпфирование». В то же время некоторые процессы протекают столь медленно, что могут рассматриваться как слишком инерционные. На рис. 13.7 показаны в сравнении характеристики систем управления с различным быстродействием.

Пропорциональное управление. Наиболее понятная концепция управления состоит в том, что компенсация ошибки рассогласования производится без задержки пропорционально значению этой ошибки. Такой тип управления называется пропорциональным и описывается следующим уравнением:

$$C_p(t) = K_p e(t), \quad (13.1)$$

где K_p — коэффициент пропорциональности; $e(t)$ — ошибка рассогласования (в функции времени); $C_p(t)$ — значение коррекции, пропорциональное ошибке (в функции времени).

Если коэффициент пропорциональности K_p является большим, то и быстродействие системы является высоким. В этом случае говорят, что система обладает высоким «коэффициентом усиления». Безусловно, при слишком большом коэффициенте усиления система может стать неустойчивой.

На рис. 13.8 приведена графическая зависимость, иллюстрирующая поведение системы при пропорциональном управлении. После затухания переходного процесса фактическое значение регулируемого параметра системы не будет в точности соответствовать заданному значению. Здесь необходима постоянная компенсация, поскольку система пропорционального управления не может полностью исключить ошибку рассогласования. Для

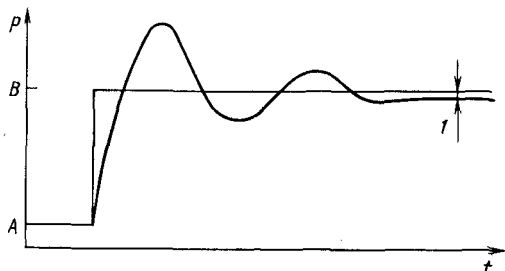


Рис. 13.8
Измененные параметров процесса $P(t)$ при пропорциональном управлении:

I — статическая ошибка рассогласования

осуществления компенсации необходимо измерить ошибку рассогласования, причем зна-

чение ошибки должно быть достаточным для распознавания последней датчиком рабочего органа.

Интегральное управление. Интегральное управление, называемое иногда управлением с компенсацией «запаздывания» или управлением с «коррекцией» ошибки, предусматривает накопление данных об ошибке рассогласования и формирование коррекции, пропорциональной полной накопленной ошибке. Интегральное управление описывается уравнением

$$C_i(t) = K_i \int_0^t e(t) dt, \quad (13.2)$$

где K_i — коэффициент интегрального управления; $e(t)$ — ошибка рассогласования (в функции времени); $C_i(t)$ — интегральная коррекция (в функции времени).

Нетрудно видеть, что постоянное рассогласование, показанное на рис. 13.8, может быть скомпенсировано интегральным управлением. Хотя рассогласование невелико, поскольку оно накапливается в течение некоторого времени, оно может стать существенным, и в конце концов, ответом будет управляющий сигнал, сформированный устройством интегрального управления. Таким образом, весьма целесообразно использовать пропорционально-интегральное управление, причем пропорциональное — для формирования управляющего сигнала в функции ошибки, а интегральное — для компенсации ошибки рассогласования.

Дифференциальное управление. Дифференциальное управление является наиболее сложной концепцией, иногда называется управлением в функции «скорости». Дифференциальное управление применяется не столь широко и в основном в сочетании с другими типами управления. Дифференциальное управление описывается уравнением

$$C_d(t) = K_d \frac{de(t)}{dt}, \quad (13.3)$$

где K_d — коэффициент дифференциального управления; $e(t)$ — ошибка рассогласования (в функции времени); $C_d(t)$ — дифференциальная коррекция (в функции времени).

При дифференциальном управлении не учитываются ни текущая, ни интегральная ошибки. В этом случае управление произ-

водится по скорости, являющейся производной ошибки рассогласования по пути. Таким образом, использование дифференциального управления позволяет до некоторой степени прогнозировать протекание технологического процесса. Действие дифференциального регулятора в начале компенсации ошибки рассогласования направлено *противоположно* действию пропорционального регулятора, поскольку пропорциональный регулятор учитывает направление вектора ошибки (к перерегулированию). Дифференциальное управление может использоваться для демпфирования перерегулирования. Дифференциальное управление, сбалансированное надлежащим образом, может обеспечить эффективное улучшение уровня пропорционального и интегрального управления.

В наиболее сложных системах управления используются все три типа управления непрерывным процессом — пропорциональное, интегральное и дифференциальное. Этот комбинированный тип управления описывается уравнением

$$C(t) = C_p(t) + C_i(t) + C_d(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}. \quad (13.4)$$

Специалисты по управлению обычно используют уравнение (13.4), в котором учтено требование, заключающееся в том, чтобы коэффициент усиления системы G удовлетворял всем трем типам управления — пропорциональному, интегральному и дифференциальному. Регулируемый параметр для интегрального управления представляет собой интегральный *период* T_i , связанный с коэффициентом усиления следующей зависимостью:

$$K_i = G/T_i.$$

Регулируемый параметр для дифференциального управления представляет собой дифференциальный период T_d , связанный с коэффициентом усиления следующей зависимостью:

$$K_d = GT_d.$$

Выражение для полного управляющего сигнала на выходе устройства управления для управления переменным параметром имеет вид

$$P = P_0 + G_e(t) + G \frac{1}{T_i} \int_0^t e(t) dT + GT_d \frac{de(t)}{dt}, \quad (13.5)$$

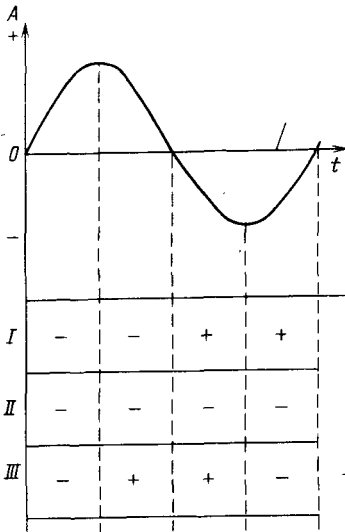
где P_0 — уставка; P — выходной сигнал управления переменным параметром.

Специалисты по управлению определяют ошибку как разность уставки и текущего значения регулируемого параметра, а не наоборот, как определили мы в данном разделе.

Рис. 13.9

Таблица положительных и отрицательных воздействий, компенсирующих погрешность воспроизведения параметра процесса $A(t)$ при трех основных режимах управления ($A = 0$ — уставка процесса):

I — пропорциональным; II — интегральным;
 III — дифференциальным



Уравнение (13.5) в большей степени совместимо с техническими средствами реальной системы управления, однако уравнение (13.4) лучше объясняет общую концепцию управления, предусматривающую использование трех типов управления: пропорционального, интегрального и дифференциального. Рис. 13.9 может рассматриваться как иллюстрация к уравнению (13.4); на рисунке в таб-

личном виде приведена чувствительность (положительная или отрицательная) каждого типа управления для различных позиций регулирования переменного регулируемого параметра на изменение уставки.

Оптимизация управления. Выбор идеального сочетания (K_p , K_i и K_d) типов управления для заданной системы представляет собой предмет оптимизации системы управления. Оптимизация требует проведения подробного исследования. Сложные исследования характеристик системы целесообразно производить с использованием преобразования Лапласа. Этот аппарат для проведения аналитических исследований существовал задолго до появления роботов и современных средств автоматизации производства. Инженеры-электрики давно использовали аппарат теории управления при расчете электрических устройств и систем. Теорией управления также пользовались инженеры-механики при разработке механизмов управления параметрами механических систем.

Появление ЭВМ для числового управления технологическими процессами привело к *цифровому представлению* непрерывных технологических параметров для анализа и управления. Цифровое квантование, используемое при оптимизации аналоговых систем управления, представляет собой новый тип измерения и реализуется с помощью дискретной техники. По сравнению с преобразованием Лапласа для исследования дискретных систем более подходящим является аппарат Z -преобразования. Подробное изложение принципов оптимизации систем управления с помощью преобразования Лапласа или Z -преобразования выходит за рамки настоящей книги. Для успешного внедрения роботов

и средств автоматизации производства нет необходимости в понимании принципов оптимизации систем управления. Для специалиста по автоматизации необходимо знать основные принципы различных типов управления и то, что реализация этих принципов возможна с помощью ЭВМ в реальном времени. Теперь остановимся на характеристиках ЭВМ, функционирующих в реальном времени, что позволяет использовать последние для управления роботами и в качестве средств автоматизации.

13.3. ИНТЕРФЕЙС ПРОИЗВОДСТВЕННОГО ОБОРУДОВАНИЯ

Кроме обычных периферийных устройств ввода-вывода, как например, печатающие устройства, дисплеи и пульта управления, которые могут подключаться к ЭВМ, должна быть обеспечена жесткая связь управляющей ЭВМ с датчиками, регистрирующими технологические параметры процесса. Для контроля технологического процесса и состояния оборудования без использования обратной связи в реальном времени необходимо использовать лишь контактные входы для подключения датчиков, выключателей, клапанов и счетчиков. Для управления с обратной связью требуются выходы для подключения пускателей электродвигателей, реле, приводов и клапанов. Входы и выходы производственного оборудования могут быть двух основных типов: цифровыми и аналоговыми.

Цифровые входы и выходы. Цифровые или дискретные входы и выходы проще стыкуются с управляющей ЭВМ, поскольку сигналы на них уже представлены в виде, соответствующем обрабатываемым ЭВМ данным. Входные и выходные сигналы обычно представляются в двоичном виде, так как большинство из них являются логическими параметрами. Ниже приведены примеры параметров цифровых входов и выходов.

Входные параметры: 1) наличие или отсутствие объекта; 2) положительное или отрицательное напряжение; 3) температура, отличная от уставки; 4) уровень жидкости, отличный от уставки.

Выходные параметры: 1) пуск или останов двигателя; 2) открывание или закрывание клапана; 3) приемка или отбраковка детали.

Некоторые параметры в приведенных выше примерах являются аналоговыми переменными (как, например, температура или уровень жидкости). Однако ввод или вывод данных из ЭВМ осуществляется в дискретном виде, причем сигнал формируется при переходе через уставку.

Для обработки на ЭВМ в реальном масштабе времени может быть предусмотрена очень эффективная упаковка цифровых данных. Например, состояния восьми различных логических параметров могут храниться в одном регистре ЭВМ объемом 1 байт (восемь двоичных разрядов). Однако цифровые данные, представленные в упакованном виде, неудобны для отработки програм-

мами, написанными на языках высокого уровня, таких как Бейсик или Фортран. Рабочие программы, предназначенные для обработки цифровой информации, в ряде случаев более эффективно могут быть написаны на языке ЭВМ низкого уровня, например на ассемблере.

Аналоговые входы и выходы. Если необходим ввод в ЭВМ текущих значений аналоговых технологических параметров, таких как температура и давление, то возникает необходимость соответствующего преобразования аналоговых величин в дискретные, поскольку ЭВМ может воспринимать лишь цифровую информацию. Перед вводом во всех случаях следует произвести преобразование аналоговых сигналов в дискретные, так как ЭВМ представляет собой цифровое устройство. При выводе информации для управления технологическим процессом во всех случаях используются ступенчатые функции или дискретные данные, поскольку ЭВМ не может изменять выходной параметр в аналоговом виде. Необходимость аппроксимации при преобразовании данных вследствие использования цифровой ЭВМ является основой *метода* аналого-цифрового преобразования и наоборот. Термин аналог используется в управлении технологическими процессами, так как физические непрерывные переменные параметры должны быть преобразованы в некоторый электрический аналоговый сигнал (обычно напряжение или ток), до того как они смогут быть интерпретированы и преобразованы с приближением в цифровую форму для ввода в ЭВМ, управляющую технологическим процессом.

Поскольку аналого-цифровое преобразование связано с аппроксимацией, следует установить, с какой точностью это преобразование должно осуществляться. Выбор точности должен быть обоснован, поскольку чем выше точность, тем большее количество двоичных цифр требуется для представления аналоговой величины. Непрерывная аналоговая входная переменная может быть представлена и одной двоичной цифрой. В этом случае эта переменная соответствует дискретной входной переменной, например, «температура выше уставки или ниже уставки», аналогично цифровым входам-выходам, рассмотренным выше. Двоичное приближение к аналоговому значению позволяет произвести выбор одного из четырех уровней дискретных значений, наиболее близкого по значению к фактическому аналоговому. В общем виде это положение имеет вид

$$N = 2^n, \quad (13.6)$$

где N — количество дискретных уровней, сформированных квантованием аналоговой величины; n — количество двоичных разрядов, которые используются для представления преобразованной величины в цифровом виде.

Аналоговый сигнал от датчика в виде напряжения поступает на аналого-цифровой преобразователь. В этом преобразователе

Рис. 13.10

Дерево решения для процедуры аналого-цифрового преобразования:

I — двоичный код; II — десятичный код

производится последовательное сравнение указанного аналогового сигнала с эталонным напряжением для каждого двоичного разряда. В результате сравнения формируется последовательность двоичных чисел. Двоичные числа располагаются слева направо (от старшего до младшего порядка).

Процесс аналого-цифрового преобразования целесообразно представить в наглядной форме в виде дерева решений. На рис. 13.10 представлено такое дерево решений, на котором три двоичные цифры эквивалентны числу 2^3 , т. е. восьми цифровым уровням, используемым для приближения в этом примере аналоговой величины. В результате преобразования аналогового значения 6,6 становится цифровым (в двоичном виде) 101. Представленное значение (среднее значение) из диапазона аналоговых значений, соответствующих числу 101, соответствует значению

$$0,5 (7,50 + 6,25) = 6,875.$$

Следовательно, погрешность округления при преобразовании равна:

$$6,875 - 6,600 = 0,275.$$

Ниже приведен пример расчета для различных эталонных аналоговых величин.

Пример 13.1. Аналого-цифровое преобразование. Аналоговая переменная процесса, которая может варьировать в диапазоне от 0 до 10 В, установлена на уровне 2,9 В. Какое цифровое значение после аналого-цифрового преобразования будет сформировано во входном регистре управляющей ЭВМ?

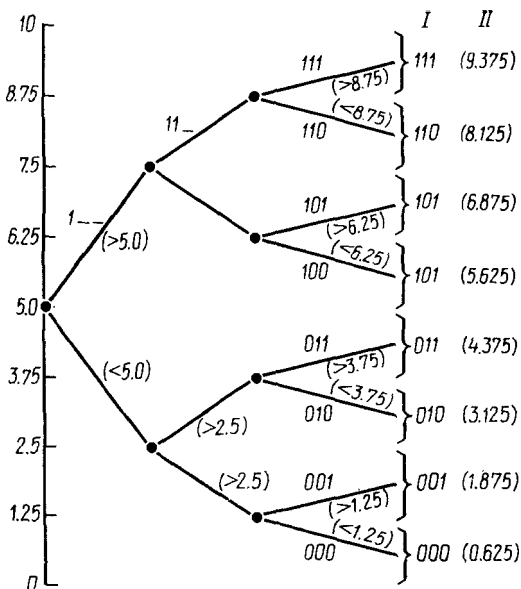
Решение. Технологическая переменная $X = 2,9$ В; число после преобразования $Y = ?$

Используя дерево решений, представленное на рис. 13.10, получим

$$X < 5,0 \rightarrow Y = 0_0;$$

$$X > 2,5 \rightarrow Y = 0_11;$$

$$X < 3,75 \rightarrow Y = 010.$$



Приближенное число (в двоичном виде) 010 в этом примере используется для представления аналогового значения 2,9. Представленное значение (среднее значение) из диапазона аналоговых значений, соответствующих двоичному числу 101, соответствует значению

$$0,5 (3,75 + 2,50) = 3,125.$$

Следовательно, погрешность округления при преобразовании $3,125 - 2,900 = 0,225$ В.

Кроме количества двоичных цифр, соответствующих аналоговой переменной, существует проблема выбора диапазона в соответствии с точностью. В общем случае параметр реального процесса может быть преобразован в электрический аналоговый сигнал, который нужно будет усилить или ослабить в заданном диапазоне напряжений, соответствующем диапазону аналого-цифрового преобразователя. Наиболее широко применяемый в промышленности стандартный диапазон для аналого-цифровых преобразователей составляет 0—10 В. Усиление или подавление напряжения сигнала до стандартного диапазона является задачей технических средств. Здесь должен быть принят во внимание ожидаемый диапазон технологической переменной. Позже при промежуточных расчетах программа управляющей ЭВМ может обеспечить умножение приближенных числовых периодов на коэффициент масштабирования. Такая операция даст числовое представление в соответствии с полным диапазоном исходной аналоговой переменной.

Расчет дискретности D аналого-цифрового преобразователя может быть произведен в функции количества двоичных цифр, соответствующих числовому приближению к полному диапазону непрерывной переменной, а именно

$$D = \frac{R}{2^n}, \quad (13.7)$$

где R — полный диапазон непрерывной переменной процесса; n — количество двоичных цифр, соответствующих числовому приближению.

Максимальная погрешность аналого-цифрового преобразования соответствует половине дискретности:

$$\Delta = \pm 0,5 \left[\frac{R}{2^n} \right] = \pm \frac{R}{2^{n+1}}. \quad (13.8)$$

Пример 13.2. Дискретность аналого-цифрового преобразования и максимальная погрешность.

ЭВМ управления автоматизированным оборудованием обеспечивает контроль подвода захвата робота к объекту с помощью бесконтактного высокочастотного датчика перемещения. Диапазон перемещения захвата робота к объекту составляет 1219 мм. В состав аналого-цифрового преобразователя входит шести-разрядный двоичный регистр, обеспечивающий запоминание приближенного числового значения, соответствующего значению аналоговой переменной, которая, как предполагается, является точной и линейной во всем диапазоне 0—1219 мм. Какова дискретность аналого-цифрового преобразователя для этой переменной процесса и какова максимальная погрешность преобразования?

Решение. Дискретность по (13.7)

$$D = \frac{1219}{2^6} = 19 \text{ мм},$$

максимальная погрешность

$$\Delta = \pm \frac{1219}{2^7} = \pm 9,5 \text{ мм} = 0,5D.$$

Можно предусмотреть одно цифровое значение на каждом конце диапазона перемещения для максимального и минимального значений вне диапазона с целью избежания ошибочного считывания. Платой за эту возможность будет снижение дискретности вследствие уменьшения общего количества цифровых уровней на два, а именно:

$$D = \frac{R}{2^n - 2} \quad (13.9)$$

и

$$\Delta = 0,5D = \pm 0,5 \left[\frac{R}{2^n - 2} \right] \quad (13.10)$$

или

$$\Delta = \pm \frac{R}{2^{n+1} - 4}. \quad (13.11)$$

В примере 13.3 показано использование индикаторов верхнего и нижнего пределов.

Пример 13.3. Аналого-цифровое преобразование с индикаторами пределов.

Роботизированная станция контроля управляется в реальном времени с помощью ЭВМ. Измеряемой переменной здесь является масса деталей от 45 до 68 кг. Однако аналого-цифровой преобразователь обеспечивает индикацию максимального и минимального значений за пределами диапазона. Это требуется для определения массы изделий вне диапазона. Если цифровой регистр рассчитан на три двоичные цифры, предназначенные для отображения преобразованного значения, то насколько снижена точность измерения в результате использования индикаторов пределов?

Решение. С индикаторами пределов по (13.9)

$$D = \frac{68 - 45}{2^3 - 2} = 3,83 \text{ кг},$$

$$\Delta = \pm 0,5D = \pm 0,5 \cdot 3,83 = \pm 1,92 \text{ кг}.$$

Без индикаторов пределов

$$D = \frac{68 - 45}{2^3} = 2,87 \text{ кг},$$

$$\Delta = \pm 0,5D = \pm 0,5 \cdot 2,87 = 1,44 \text{ кг}.$$

Снижение дискретности

$$\Delta D = 3,83 - 2,87 = 0,96 \text{ кг},$$

увеличение максимальной погрешности

$$\Delta_1 = 1,92 - 1,44 = 0,48 \text{ кг}.$$

Процесс аналого-цифрового преобразования называется кодированием. Целесообразно снова декодировать цифровое значение переменной в соответствии с формулой

$$Y = R \sum_{i=1}^n \frac{B_i}{2^i} + \frac{1}{2} \left[\frac{R}{2^n} \right] + Y_{\min} =$$

$$= R \left[\frac{B_1}{2} + \frac{B_2}{4} + \dots + \frac{B_n}{2^n} \right] + \frac{1}{2} \left[\frac{R}{2^n} \right] + Y_{\min}, \quad (13.12)$$

где Y — цифровое значение, приближенное к аналоговой переменной процесса; R — полный диапазон аналоговой переменной; B_i — значение двоичного числа $(n+1-i)$ -порядка; n — количество двоичных цифр при аналого-цифровом преобразовании; $Y_{\min}$ — минимальное ожидаемое значение переменной процесса.

Приведенная формула получена из соотношения стандартного двоично-десятичного преобразования с учетом коррекции с целью размещения каждой цифровой дискреты в середине диапазона с полным количеством в масштабе к полному диапазону аналоговой переменной.

Уравнение (13.12) может быть упрощено следующим образом:

$$Y = R \left[\sum_{i=1}^n \frac{B_i}{2^i} + \frac{1}{2} \left[\frac{1}{2^n} \right] \right] + Y_{\min} = R \sum_{i=1}^{n+1} \frac{B_i}{2^i} + Y_{\min} =$$

$$= R \left[\frac{B_1}{2} + \frac{B_2}{4} + \dots + \frac{1}{2^{n+1}} \right] + Y_{\min}. \quad (13.13)$$

Уравнение (13.13) является более компактным в части необходимых расчетов, однако несколько проигрывает по наглядности в сравнении с уравнением (13.12). Читатель может обосновать вывод уравнений (13.12) и (13.13) с помощью рис. 13.10. Можно использовать любое из двух уравнений, однако следует подчеркнуть следующее: ни уравнение (13.12), ни уравнение (13.13) не отвечают случаю, когда индикаторы верхнего и нижнего пределов включены в кодирование. Декодирование в подобном случае для различных значений n рассмотрено в учебном примере, приведенном ниже.

Пример 13.4. Декодирование данных аналого-цифрового преобразования. Двоичный регистр ЭВМ содержит приближенное цифровое значение аналоговой переменной, которая может изменяться в диапазоне 8,4—12,6 МПа (1200—1800 англ. фунт/кв. дюйм). В регистре записано двоичное число 1011. Какому давлению это значение соответствует?

Решение. Используя уравнение (13.13), получим

$$Y = (1800 - 1200) \left[\frac{1}{2} + \frac{0}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} \right] + 1200 =$$

$$= 1631,25 \text{ англ. фунт/кв. дюйм.}$$

Важной областью применения принципа декодирования является преобразование выходного сигнала управляющей ЭВМ

в аналоговый вид. Этот тип декодирования называется цифроаналоговым преобразованием выходного сигнала и представляет собой операцию, обратную аналого-цифровому преобразованию входного сигнала. Однако следует подчеркнуть, что цифроаналоговое преобразование выходных сигналов имеет не столь большое значение, как аналого-цифровое преобразование входных сигналов. Дело в том, что производственное оборудование в большинстве случаев включает преобразователь, который может воспринимать выходные сигналы от устройства управления, представленные в цифровом виде. В качестве примера здесь можно привести шаговый двигатель, которым оснащаются рабочие органы робота. ЭВМ устройства управления выдает дискретный ряд импульсов напряжения на шаговый двигатель, который обеспечивает перемещение рабочего органа на заданную величину. Следует также учесть, что измерение и контроль работы производственного оборудования занимают большую часть времени функционирования управляющей ЭВМ по сравнению со временем, затрачиваемым на формирование выходных сигналов. Это обуславливает большую важность аналого-цифрового преобразования по сравнению с цифроаналоговым.

Как аналого-цифровое, так и цифроаналоговое преобразование должны выполняться на периодической основе. Для выполнения преобразования требуется ограниченное количество времени. В промежутках между циклами преобразования управляющая ЭВМ выполняет другие функции. При этом следует определить значение переменной, которое должно быть выбрано между циклами преобразования. Очевидный алгоритм предполагает сохранять значение переменной на предварительно вычисленном уровне до завершения следующего цикла преобразования. Однако другим алгоритмом принимается во внимание скорость изменения переменного параметра, особенно в случае, когда скорость изменения велика. На рис. 13.11 приведено сравнение этих двух алгоритмов. Погрешность, возникающая между циклами преобразования, может превышать дискретность преобразования. Исследование этого вопроса входит в задачу теории дискретных систем, представляющую определенный интерес. Однако подробное изложение этой сложной проблемы выходит за рамки настоящей книги.

Рассматривая указанную проблему дальше, следует отметить, что аналоговые входные и выходные сигналы не имеют для роботов и автоматизированного производственного оборудования столь большого значения, как цифровые входные и выходные сигналы. Автоматическая коррекция технологических параметров формируется именно на основе анализа сочетания состояний цифровых логических сигналов, характеризующих технологический процесс, а не по уровню изменения аналоговых переменных. Управление промышленным оборудованием производится, как правило, с помощью дискретных логических функциональных выходных сигналов, а не ответных аналоговых сигналов. Дей-

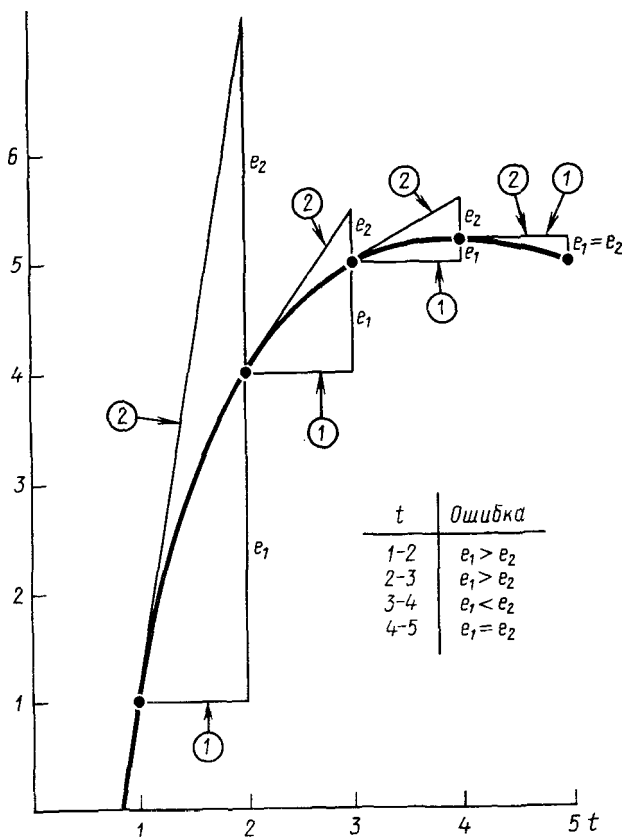


Рис. 13.11

Сравнение стратегий приближения аналоговой переменной при квантовании:
 1 — период квантования; 2 — скорость изменения параметра за период квантования

ствительно, заводы с непрерывными технологическими процессами, как, например, химические, нефтеочистительные и цементные, электростанции имеют множество критических аналоговых технологических параметров, которые подлежат тщательному контролю. Однако в робототехнике и автоматизированном производстве изделия, технологические процессы, а также параметры процесса управления в большинстве случаев являются дискретными.

13.4. ПРЕРЫВАНИЯ

Представим автоматизированный технологический процесс с множеством переменных параметров, который управляется от ЭВМ в реальном времени. Для контроля соответствующих параметров в заданные временные интервалы должна быть установлена некоторая последовательность операций. Значения неко-

торых параметров или условий соответствуют состоянию, требующему незамедлительного прерывания исполнения ЭВМ других функций. При создании условий, не требующих столь срочного вмешательства в технологический процесс, прерывание может быть задержано до проведения периодической проверки с помощью ЭВМ параметров, характеризующих эти условия. Последовательность исполнения прерываний устанавливается назначением приоритетов. В обоих приведенных случаях используются прерывания. Однако при возникновении условий, соответствующих аварийной ситуации, используется безусловное прерывание, а в другом случае — прерывание по таймеру, после опроса параметров, характеризующих контролируемые условия.

Генератор реального времени. Управление таймерными прерываниями осуществляется с помощью генератора реального времени, который имеется в составе управляющей ЭВМ. Генератор позволяет синхронизировать работу ЭВМ с внешним периферийным оборудованием, в том числе с контролируемым и управляемым производственным оборудованием. Как правило, генератор является высокочастотным, в то время как в счетчиках, используемых в режимах, описанных в гл. 11 и 12, в соответствии с требованиями программ применяются низкочастотные импульсы. Поскольку генератор функционирует в реальном времени, формирование импульсов производится непрерывно, независимо от программируемой логики. При использовании генератора для реализации управляющей программы должен быть организован доступ к адресным регистрам. В соответствующий регистр записывается некоторое положительное целое число. Подающиеся на вход регистра импульсы вычитают из этого числа по единице, до тех пор пока содержимое регистра не станет равным нулю. По достижении нуля в регистре генератор автоматически организует прерывание. Это прерывание может быть использовано программистом для назначения программы, с помощью которой реализуется функция, представляющая собой ответ на это прерывание. Такая программа может проанализировать технологические параметры с целью определения необходимости ввода функции, соответствующей условиям с отклонением от нормы.

Приоритеты. ЭВМ, функционирующая в реальном времени совместно с программным обеспечением, рассчитана на формирование управляющих сигналов, поступающих на управляемое производственное оборудование, в зависимости от информации от датчиков этого оборудования. Следовательно, ЭВМ может находиться как в рабочем состоянии, так и в состоянии ожидания в зависимости от процесса. В зависимости от значений технологических параметров с отклонением от нормы могут создаваться аварийные условия, в то время как ЭВМ решает первые по очередности задачи.

Несмотря на высокое быстродействие ЭВМ может сложиться ситуация, при которой становится затруднительным проанализи-

ровать большое количество условий. Отклонение каждого условия от нормы обуславливает прерывание и выдает ЭВМ запрос на реализацию соответствующей функции. Типовой пример может предусматривать потенциальную возможность прерывания работы ЭВМ в любой момент по одному из 50 условий или по всем условиям одновременно, если к этому добавить периодическое таймерное прерывание для опроса других 200 условий работы в интервалы длительностью, например, 5 с. Хотя время, необходимое для обслуживания любого из прерываний, может измеряться в миллисекундах, наличие большого количества условий с отклонением от нормы может обусловить неспособность ЭВМ завершить обслуживание всех прерываний. Эта особенность работы ЭВМ в реальном времени привела к необходимости разработки сложных алгоритмов программного обеспечения, в функции которого входит назначение приоритетов для различного рода прерываний работы.

Подробный анализ приоритетов прерываний дает специалисту по автоматизации очень детальную информацию об управляемом процессе. На основе полученной информации в ряде случаев могут быть улучшены характеристики производственного оборудования. Это является одним из дополнительных преимуществ автоматизации производства. Предположим, например, что управляющая ЭВМ контролирует количество деталей в позициях загрузки на роботизированной сборочной станции. Если количество различных позиций загрузки велико, а размеры буферного накопителя для каждой позиции малы, то может иметь место перегрузка ЭВМ, контролирующей количество деталей и обслуживающей указанные позиции. Улучшение конструктивных характеристик транспортно-загрузочной системы может быть достигнуто за счет увеличения размеров буферных накопителей.

Как только прерываниям ЭВМ будут присвоены соответствующие приоритетные уровни, ЭВМ сможет игнорировать прерывания с низким приоритетным уровнем во время обслуживания прерываний с более высоким приоритетом. Для описания запрета действия прерываний с низким приоритетом используется термин *маскирование прерываний*. Система, которая распознает различные уровни приоритетов прерывания, называется *многоуровневой системой прерывания*. На рис. 13.12 приведено сравнение такой системы с одноуровневой системой прерывания.

Назначение соответствующих приоритетных уровней встречается более часто, чем считают некоторые специалисты. Например, рассмотрим следующий список условий прерываний:

1. Низкий уровень давления рабочей жидкости в гидropriводе рабочего органа робота.
2. Пустой загрузочный бункер.
3. Задержка деталей в спускном желобе.
4. Сообщение, передаваемое на экран дисплея.
5. Оператор включает переключатель на пульте управления.

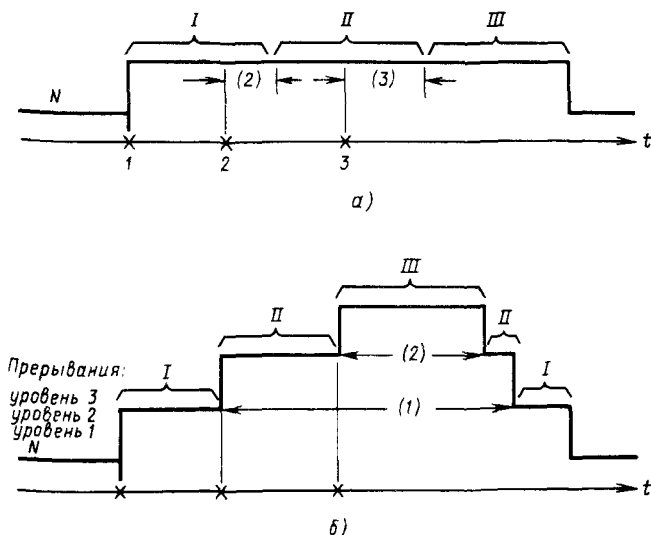


Рис. 13.12

Сравнение быстродействия ЭВМ при одноуровневой (а) и многоуровневой (б) системах прерывания:

I—III — задачи; N — нормальный режим; (1)—(3) — ожидание

Какое из приведенных выше условий обладает наивысшим приоритетом? Наинизшим приоритетом? Большинство, вероятно, будет считать, что условие I по срочности обладает высшим приоритетом, а условие 5 — низшим приоритетом. Действительно, в результате более внимательного анализа границ каждого условия высший приоритет был бы присвоен условию 5, а условию 1 — низший приоритет. При создании критической ситуации оператору может быть дано указание принять на себя ручное управление производственным оборудованием. Представим себе промышленный робот, который не реагирует на команду, выданную оператором, вследствие перегруженности решением множества задач управления и контроля производственного процесса, которые предназначены для реализации в автоматизированном режиме. К сожалению, некоторые специалисты могут предлагать такое же распределение приоритетов.

Если вернуться к списку условий прерывания, условием прерывания с низким приоритетом может оказаться 4. При возникновении такой аварийной ситуации ЭВМ будет настолько перегружена, что не сможет обеспечить вывод на печать сообщений. Тогда какой смысл в определении неисправностей, когда у ЭВМ отсутствует время для вывода соответствующих сообщений на печать? Существует реальная возможность разработки диагностической программы для контроля серьезных аварийных ситуаций, возникающих в производственном процессе, с помощью

ЭВМ. Однако при этом у ЭВМ может не хватить времени для устранения создавшейся аварийной ситуации. Разве нельзя здесь провести аналогию с поведением некоторых людей в кризисных ситуациях? По мере накопления опыта в назначении приоритетов при программировании промышленных роботов и управляющих ЭВМ можно получить более полные данные о поведении людей.

Нами исследованы некоторые соображения о приоритетах в случае возникновения аварийной ситуации; теперь остановимся на противоположной проблеме: предположим, что ЭВМ обслужила все прерывания и простаивает! Такое положение должно быть нормой для ЭВМ, функционирующей в реальном времени, поскольку в случае возникновения непредвиденной неисправности у ЭВМ должно быть достаточно времени для обслуживания соответствующих программ прерывания. Существуют два пути решения этой проблемы, связанной с постоянной загрузкой ЭВМ:

1. Фоновая обработка данных в пакетном режиме.
2. Холостная обработка программ.

Фоновая обработка данных выполняется в режиме разделения времени и иногда называется обработкой не в реальном масштабе времени. Это не означает, что фоновая обработка имеет меньшее значение. Однако с точки зрения аварийной ситуации фоновая обработка данных, как правило, может быть отложена на несколько секунд, минут или часов, в зависимости от степени серьезности ситуации, требующей прерывания в реальном времени, и в зависимости от сложности функции, реализуемой ЭВМ для контроля такой ситуации.

Возможно, что больший интерес представляет собой холостная обработка программ. Эта концепция является наиболее трудной для восприятия программистами, имеющими опыт обычного программирования в пакетном режиме. Рассмотрим следующий сегмент программы, представленной на языке Фортран:

```
C TWIDDLE THUMBS
12 GO TO 13
13 GO TO 12
```

или на языке Бейсик:

```
12 GOTO 13 : REM TWIDDLE
13 GOTO 12 : REM THUMBS.
```

Приведенный выше сегмент программы представляет собой бесконечный цикл, который наиболее трудно воспринимается всеми начинающими программистами. Однако бесконечные циклы в программе находят широкое применение при программировании ЭВМ в реальном времени, а в ряде случаев применение бесконечных циклов необходимо для поддержания ЭВМ в рабочем режиме при ожидании условия прерывания. Следует помнить, что прерывание приведет к прекращению выполнения ЭВМ реализуемой команды и условному или безусловному переходу к исполнению следующей команды. Ветвление программы всегда неизменно,

и ЭВМ никогда не возвратится к прекращению исполнения программ, если не будет дана соответствующая команда, сформированная специальной программой.

13.5. ПРОГРАММИРОВАНИЕ УПРАВЛЯЮЩЕЙ ЭВМ

Здесь читатель должен понимать, что существуют некоторые различия между программированием управляющей ЭВМ в реальном времени и обычным программированием в режиме пакетной обработки. Однако возможно программирование на тех же самых или аналогичных простых языках компилятора. Вместо языка Фортран может применяться язык Process Фортран, вместо языка Бейсик возможно применение языка Power Бейсик. Вместо языка ассемблера может быть использован язык ПАЛ (*Process Assembly Language*). При использовании языка любого типа должны быть в наличии средства на этом языке, обеспечивающие распознавание (или запрет) параметров технологического процесса и/или

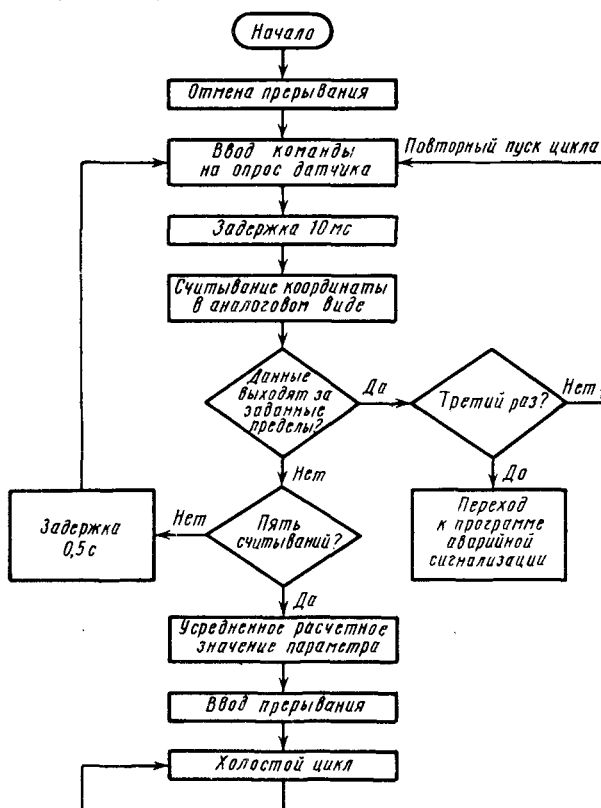


Рис. 13.13

Структурная схема алгоритма управления процессом по программе в реальном времени с помощью ЭВМ

прерываний в реальном времени. Такой язык должен также обеспечить прием и обработку входных сигналов от датчиков производственного оборудования и выдачу соответствующих выходных сигналов. Требования к высокой эффективности реализации программ и необходимость обработки последних в двоичных кодах обуславливают предпочтительный выбор языков ассемблера для многих случаев применения, хотя наряду с языками ассемблера также применяются языки компилятора.

На рис. 13.13 изображена схема алгоритма программы, реализованной на ЭВМ. Эта программа является частью системы программного обеспечения, предназначенного для контроля в реальном времени технологических параметров. Назначение алгоритма, структурная схема которого представлена на рис. 13.13, состоит в считывании усредненных значений параметра путем квантования в реальном времени. Другой функцией такой программы является аварийная сигнализация при перегрузке путем индикации с помощью аналоговых приборов. Перегрузка устанавливается при превышении измеряемым сигналом уставки.

Реальные программы, с помощью которых реализуется на ЭВМ структурная схема алгоритма, представленная на рис. 13.13, здесь для краткости не рассматриваются. На практике кодирование будет определяться применяемыми техническими средствами, а также программным обеспечением компилятора или ассемблера. Целью настоящей главы являлось рассмотрение концепций, позволяющих инженеру по автоматизации представить в целом подход к программированию на ЭВМ в реальном времени и применению программного обеспечения в автоматизированном оборудовании.

В ы в о д ы

ЭВМ для управления производственными процессами в реальном времени сходны с обычными ЭВМ, предназначенными для работы в режиме пакетной обработки с расширенными возможностями. Основными из этих возможностей являются считывание входной информации и «запись» или передача выходной информации, а также ответ на внешние прерывания. Автоматизация технологического процесса с помощью ЭВМ может вестись на нескольких уровнях — от обработки данных не в реальном времени до управления оборудованием с обратной связью.

ЭВМ для управления производственным процессом в реальном времени может принимать технологическую информацию в двух видах: цифровом и аналоговом. То же самое касается информации, передаваемой с выходов ЭВМ в производственное оборудование. Поскольку ЭВМ представляет собой цифровую машину, то вся аналоговая информация подлежит преобразованию (кодированию) с помощью аналого-цифрового преобразователя в цифровой код. Любое представление аналоговых данных является приближенным. При этом точность преобразования определяется

диапазоном представления аналогового параметра и числом двоичных разрядов при цифровом представлении. Преобразование цифровых данных в аналоговый вид называется декодированием. Несмотря на широкое распространение аналоговой информации, особенно при непрерывных технологических процессах в робототехнике и производственном автоматизированном оборудовании, более существенную роль продолжают играть цифровые входы и выходы.

Опрос датчиков и формирование выходных сигналов на приводы производственного оборудования, а также таймерные прерывания являются существенными функциями программной обеспечения ЭВМ, функционирующей в реальном времени. В кризисной ситуации ЭВМ легко может оказаться перегруженной прерываниями и не сможет выполнять более важные функции. При этом становится необходимым назначение приоритетов различным прерываниям и запрещение или маскирование прерываний с более низкими приоритетами во время обслуживания прерываний с более высокими приоритетами. Анализ и назначение приоритетов прерываний являются важными функциями и в ряде случаев требуют тщательно продуманных решений. Специалист, занимающийся автоматизацией производственного оборудования, должен обладать соответствующими опытом и квалификацией.

На практике программирование ЭВМ в реальном времени сходно с программированием пакетной обработки данных с использованием дополнительных команд прерываний, а также передачи и приема данных через входы-выходы. Требования, предъявляемые к ЭВМ, функционирующим в реальном времени, в части быстродействия и высокой производительности обуславливают в ряде случаев предпочтение использования языков ассемблера, хотя с такими ЭВМ применяются языки как ассемблера, так и компилятора. Специалист по автоматизации, познакомившись с рассмотренными базовыми концепциями и с основами применения ЭВМ, обеспечивающими управление в реальном времени, должен быть подготовлен к изучению специальных языков программирования, предназначенных для пользователей систем управления производственным оборудованием — роботами и автоматизированными производственными установками.

У п р а ж н е н и я и в о п р о с ы

13.1. Каковы принципиальные различия между ЭВМ, предназначенными для управления производственными процессами в реальном времени, и ЭВМ, использующимися в обычном режиме обработки данных?

13.2. Опишите по крайней мере три уровня или степени реализации ЭВМ, функционирующей в реальном времени, для автоматизированного оборудования.

13.3. Каково основное преимущество режима со вспомогательным контролем в реальном времени по сравнению с режимом обычного контроля технологического процесса при использовании ЭВМ для контроля производственного оборудования в реальном времени?

13.4. Какой тип преобразователя — аналого-цифровой или цифроаналоговый — считается более важным в робототехнике или в автоматизированном оборудовании и почему?

13.5. Какой наиболее серьезный недостаток способа обработки данных не в реальном времени в применении к управлению производственным процессом?

13.6. Промышленный робот измеряет силу сопротивления вставлению детали при сборочной операции с помощью датчика, который преобразует силу сопротивления в фунтах (кН) в аналоговое напряжение, поступающее на вход ЭВМ, обеспечивающей управление роботом в реальном времени. Диапазон изменения силы сопротивления при сборке составляет 20—50 фунтов (90—22,7 кН). При преобразовании силы используются пять двоичных цифр.

а) Какова дискретность аналого-цифрового преобразователя при измерении указанного технологического параметра в фунтах (кН)?

б) Какова максимальная погрешность преобразования?

в) Определить двоичное число, эквивалентное силе сопротивления при сборке 34 фунтов (154 кН).

г) Расшифруйте двоичное число 10111 для указанной системы измерения. 13.7. Обычные пределы температуры при сварке для некоторых роботов, предназначенных для дуговой сварки, составляют 6000—8000 °F. Температурный датчик обеспечивает преобразование температуры в аналоговый сигнал напряжения. Этот датчик связан через аналого-цифровой преобразователь с устройством управления роботом. Устройство управления выполнено на основе ЭВМ, функционирующей в реальном времени. Верхний и нижний пределы описываются тремя двоичными цифрами — 111 и 000 соответственно.

а) Рассчитайте дискретность аналого-цифрового преобразователя в градусах по Фаренгейту.

б) Определите максимальную погрешность преобразования.

в) Определите эквивалент температуры 7500 °F в двоичном коде.

13.8. В таблице приведены временные интервалы, в которых инициируется каждое из четырех прерываний, и временные интервалы, необходимые для обслуживания ЭВМ каждого из этих прерываний.

Прерывание	Время иницирования	Время обслуживания
1	20	6
2	24	4
3	29	9
4	35	2

Определите время окончания обслуживания прерывания 3, если:

а) все прерывания имеют один уровень;

б) прерывания имеют несколько уровней и приоритет каждого из прерываний выше каждого последующего;

в) прерывания имеют один уровень; при этом прерывание 2 маскировано.

13.9. Какой из основных режимов управления с обратной связью обеспечивает коррекцию накопленной погрешности рассогласования в системе управления? Какой режим управления обеспечивает минимальное перерегулирование?

13.10. Объясните понятие «коэффициент демпфирования».

13.11. Какой из трех режимов управления с обратной связью имеет наибольшую реакцию без задержки на регулируемый технологический параметр при внезапном изменении уставки технологического параметра? Какой режим управления без задержки компенсирует погрешность рассогласования?

13.12. Почему существует предел быстрогодействия системы управления при компенсации погрешности рассогласования между уставкой и текущим значением технологического параметра? Что произойдет в том случае, если коэффициент усиления системы управления окажется слишком большим?

14. МИКРОПРОЦЕССОРЫ

Основой устройства управления практически каждого промышленного робота или автоматизированного оборудования является по крайней мере один микропроцессор. Главой, посвященной микропроцессорам, в настоящей книге могла бы быть гл. 1 вместо гл. 14. Однако большинство специалистов-разработчиков и пользователей роботов не желают изучать эти микроЭВМ, выполненные на кристаллах (чипах), находящихся в составе указанного оборудования, которым им приходится заниматься. Для многих специалистов бывает достаточно располагать информацией, что программирование микропроцессоров с целью быстрого выполнения разнообразных расчетов, контроля операций и решения других задач осуществляется достаточно сложно. Однако крохотные кристаллы, на основе которых выполнены микропроцессоры с удивительными характеристиками, в большей степени, чем какая-либо другая технологическая разработка, оказали влияние на революционное развитие промышленных роботов и средств автоматизации производства.

Микропроцессор является *основным элементом* в конструкции автоматизированного производственного оборудования. Поэтому в заключительной части этого учебника или любого другого курса по автоматизации промышленного оборудования целесообразно рассмотреть особенности этого технологического открытия и исследовать его потенциальные возможности. Те читатели, которые не знакомы с технологией запоминающих устройств — постоянных типа ROM (ПЗУ), оперативных типа RAM (ЗУПВ), стираемых программируемых постоянных типа EPROM (СППЗУ) электрически перепрограммируемых постоянных типа EAROM (ЭППЗУ) — должны пересмотреть свое отношение к промышленным роботам и автоматизированному оборудованию, поскольку эти устройства являются основой успешного развития технических средств нового промышленного автоматизированного оборудования. Понимание принципов устройства и функционирования запоминающих устройств может раскрыть различие между пониманием существующей тенденции развития средств автоматизации и альтернативой к пониманию следующего: каждый день может случиться новое открытие! Для оценки потенциальных

возможностей микропроцессоров необходимо стать квалифицированным специалистом в программировании на языке ассемблера. На практике для программистов не представляет трудности выполнять программирование на таком высоком уровне, что подготавливаемые ими программы, реализуемые на микропроцессорах, представляют собой чуть ли не игры. При этом программисты упускают из виду потенциальные возможности своих «игрушек», применяемых в реальных средствах автоматизации производства. На практике компьютерные игры очень способствуют обучению программированию. При этом некоторые из наиболее интересных и сложных программ, реализуемых на микропроцессорах, рассчитаны на игры. Однако компьютерные игры не являются предметом рассмотрения в настоящей книге. Поэтому, избегая сложных деталей, попытаемся в этой главе раскрыть некоторые особенности микропроцессоров при их использовании в робототехнике и при автоматизации производства.

14.1. ПРЕИМУЩЕСТВА МИКРОПРОЦЕССОРОВ

Каковы наиболее характерные особенности микропроцессоров? Ответ на этот вопрос удивительно прост: микропроцессоры обладают характеристиками, сходными с предшествовавшими им вычислительными системами при меньшей стоимости. Чтобы проанализировать экономическую сторону указанной проблемы, сравним микропроцессоры с другой типичной и широко используемой в производстве машиной — вильчатым автопогрузчиком. В 1960 г. стоимость вильчатого автопогрузчика составляла 10 тыс. долл., а стоимость ЭВМ с памятью 64 К слов — около 1 млн. долл., не считая периферийного оборудования. В середине 80-х гг. стоимость вильчатого автопогрузчика, аналогичного по характеристикам автопогрузчику выпуска 1960 г., составляла около 18 тыс. долл., в то время как стоимость микропроцессорной интегральной схемы вместе с памятью 64 К слов не превышала 75 долл. Стоимость вильчатого автопогрузчика возросла. Однако, если бы его стоимость снижалась такими же темпами, как и стоимость ЭВМ с памятью 64 К слов, то в середине 80-х гг. стоимость автопогрузчика упала бы до 75 центов. Представьте теперь, что появилась возможность приобретения нового мощного вильчатого автопогрузчика стоимостью 18 тыс. долл. за 75 центов! Однако это только то, что произошло за последние два-три десятилетия с технологией производства вычислительной техники лишь в части стоимости. В то же время аналогичные снижению стоимости изменения произошли в части уменьшения энергопотребления и повышения быстродействия.

Рассмотрим пример с тяжелыми реактивными самолетами. В 1960 г. тяжелый четырехдвигательный реактивный самолет, рассчитанный на 200 пассажиров (Боинг 707) стоил приблизительно 6 млн. долл., т. е. в 6 раз больше ЭВМ с памятью 64 К слов.

Если бы технология создания реактивных самолетов в последние два-три десятилетия совершенствовалась такими же темпами, как и технология производства вычислительной техники, то каждый из нас оказался бы владельцем персонального реактивного самолета, такого же большого, как самолет № 1 ВВС президента США. Это объясняется тем, что стоимость таких самолетов оказалась бы значительно ниже, чем стоимость личных автомобилей, т. е. примерно на уровне 450 долл. Кроме того, экономия энергозатрат (стоимость топлива) оказалась бы столь велика, что тяжелым реактивным самолетам требовалось бы в расчете на милю меньше топлива, чем на такое же расстояние личному автомобилю. Подобное развитие технологии авиационного привело бы к необходимости полного пересмотра транспортных авиалиний, а в нашем обществе произошли бы невероятные изменения. Подобная ситуация сложилась с использованием микропроцессоров, внедрение которых находится лишь на начальном этапе. Результаты сравнения микропроцессоров с самолетами не являются преувеличением, поскольку существуют основания полагать, что последние достижения микропроцессорной технологии в конечном итоге окажут более глубокий эффект на наше общество в целом, и в особенности на промышленные предприятия, чем аналогичные достижения в технологии авиационного. Основанием для подобного заявления служит уровень универсальности самих микропроцессоров и возможности их широкого применения в робототехнике и автоматизированном производстве.

14.2. МИКРОПРОЦЕССОРЫ И МИКРОЭВМ

Термины *микропроцессоры* и *микроЭВМ* в настоящее время стали взаимозаменяемыми. Однако следует отметить их различие при использовании в роботах и автоматизированном оборудовании. МикроЭВМ представляет собой вычислительную систему для обработки данных. В состав такой системы в качестве центрального процессора входит микропроцессор. В вычислительной системе для ввода и вывода данных обычно используется пульт с клавиатурой и дисплеем. Такая система может оснащаться периферийным оборудованием, таким, как накопители на кассетах с магнитной лентой, дисководы с гибкими магнитными дисками, печатающие устройства. Однако микропроцессор в том качестве, в каком он рассматривается в настоящей главе, не имеет ни одного из указанных периферийных устройств. Микропроцессор выполнен на основе кремниевого кристалла, содержащего электронную схему, которая выполняет функции центрального процессора и включает несколько регистров памяти. При использовании дополнительных интегральных схем (ИС) для запоминания или ввода-вывода данных все устройство может рассматриваться в качестве микропроцессорной системы. Однако лучше оставить термин микроЭВМ за системами, которые оснащены, по крайней

мере, пультом с клавиатурой и дисплеем для ввода и вывода данных.

Специалисты по вычислительной технике могут спросить, каким образом можно использовать микропроцессор, если у него отсутствует оборудование, позволяющее производить программирование, вводить в него данные и получать на выходе расчетные данные. Однако подобный вопрос может быть задан лишь в случае, если микропроцессор рассматривать в качестве ЭВМ. По ряду параметров микропроцессоры имеют преимущества перед микро-ЭВМ. Чрезвычайно низкая стоимость и гибкость микропроцессоров, выполненных на кристаллах, отличающихся малыми габаритными размерами, обусловили их использование в роботах и автоматизированном оборудовании без периферийных устройств ввода-вывода данных, привязку которых к микропроцессорам могут осуществить специалисты по вычислительной технике.

Нужно признать, что на начальном этапе работ возникает необходимость временной привязки к микропроцессору некоторых средств отладки программного обеспечения пользователя. Однако стоимость микропроцессоров столь мала, что загрузка программного обеспечения пользователя в память микропроцессора выполняется один раз. При этом техническое исполнение микропроцессора может быть специализированным, рассчитанным на решение одной задачи в течение всего срока службы. В дальнейшем, при необходимости, гибкость микропроцессора позволяет создавать модификации или даже производить замену всего программного обеспечения. Однако в данном разделе важно подчеркнуть, что специалист по автоматизации располагает *возможностью* разработать дешевого специализированного устройства на базе микропроцессора с реализацией определенных функций.

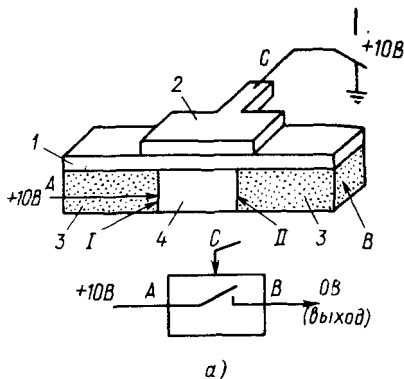
Сравним микропроцессор с ЭВМ, описанными в гл. 13. Универсальные ЭВМ, при этом не имеют значения — большие, малые или микро-, занимают свое место в робототехнике и автоматизированном производстве. Однако роль этих ЭВМ существенно отличается от роли микропроцессоров, применяемых в тех же областях. Применение микропроцессоров в действительности может быть связано с жесткой автоматизацией, поскольку здесь существует относительное постоянство в наладке, и при смене продукции требуется значительная переналадка. Однако микропроцессорная система в целом обладает большей гибкостью, чем жесткие системы автоматизации, и при переналадке требуется лишь незначительная доля затрат на технические средства по сравнению с жесткими системами автоматизации.

Программирование микропроцессоров, выполненных на малогабаритных кремниевых кристаллах, представляет собой сложную и трудоемкую задачу, которая рассмотрена ниже. Однако сначала рассмотрим конструктивные особенности интегральных микросхем, примером которых может служить микропроцессор.

Рис. 14.1

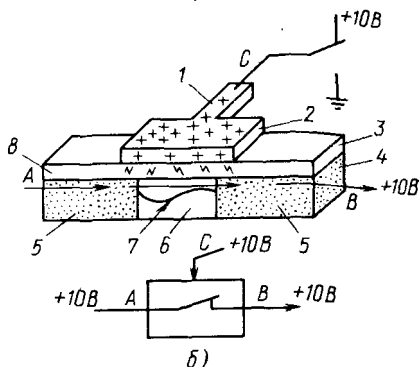
Конструкция транзистора *n*-типа МОП-структуры на микросхеме, функционирующего в релейном режиме:

a — выключатель: *I* — электрический ток блокирован *p-p*-переходом; *II* — ток к выходу отсутствует (0 В); *1* — изолирующий слой окиси кремния; *2* — металлическая пластина; *3* — кремний *n*-типа; *4* — кремний *p*-типа; *б* — переход материала *p*-типа в материал *n*-типа: *1* — управляющий сигнал 10 В вводит положительные заряды в металлическую пластину; *2* — металл; *3* — окисел; *4* — полупроводник; *5* — кремний *n*-типа; *6* — кремний *p*-типа; *7* — электрическое поле создает временный канал *n*-типа; *8* — электрическое поле, сформированное положительными зарядами на металлической пластине



14.3. ИНТЕГРАЛЬНЫЕ СХЕМЫ

На кремниевом кристалле могут быть размещены или нанесены электрохимическим путем схемы, реализующие программное обеспечение. Нет никаких трудностей в размещении схемного монтажа на большой печатной плате, поскольку монтаж выполняется



полняется в масштабе, который допускает визуальный контроль. Довольно просто распространить эту концепцию на микроскопический уровень. Однако затруднительно представить, каким образом такие компоненты микросхемы, как резисторы, конденсаторы и транзисторы, также могут быть в микроскопическом виде нанесены на кремниевый кристалл. Эта возможность существует благодаря тому, что кремний, являющийся непроводником, может быть трансформирован в полупроводник путем добавок соответствующих примесей. Более существенным является то, что электрическое напряжение, приложенное к полупроводнику, изменяет его проводимость. Возможность регулирования или управления электрическим током схемным путем с помощью сигнала является основой для создания полупроводниковых схем, выполняющих функции реле, усилителей, а также схем памяти ЭВМ. На рис. 14.1 показан принцип работы транзистора, который выполняет функции, аналогичные рассмотренным в гл. 2 для электромагнитного реле. На рисунке показаны два типа кремниевых полупроводников: *n*-типа и *p*-типа, которые получены путем внесения различных примесей. Ток, проходящий через полупроводник *n*-типа, блокируется малым участком кремниевое кристалла *n*-типа. Однако, если приложить электрическое напряжение к небольшому участку полупроводника *p*-типа, то последний может

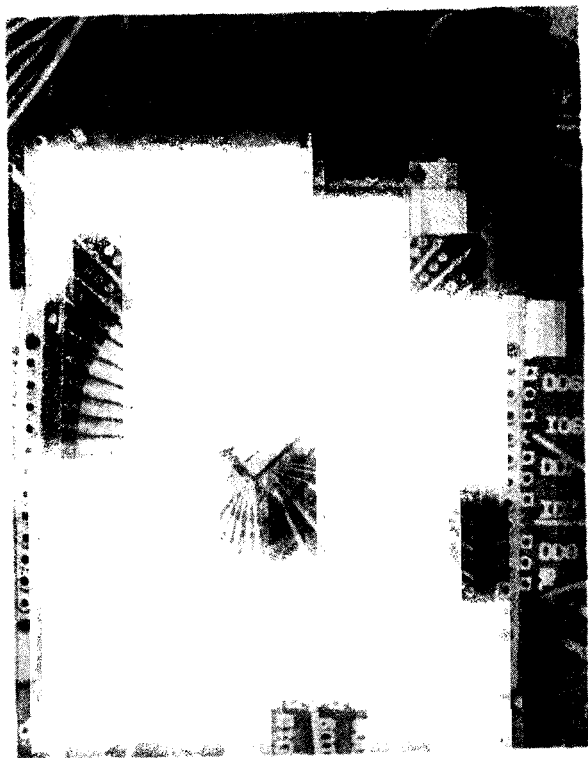


Рис. 14.2
Тестирование микросхем на кремниевой пластине

оказаться временно в проводящем состоянии и будет пропускать электрический ток подобно полупроводнику n -типа. Металлическая пластина, на которую подано электрическое напряжение, отделена от полупроводниковых слоев n - и p -типа изоляционным слоем, представляющим собой окись кремния. При этом три слоя представляют собой МОП-структуру (металл—окисел—полупроводник). В этом приборе отсутствуют какие-либо детали, подлежащие сборке: на слоях, расположенных на кремниевой пластине, должен быть нанесен монтаж или произведено травление схемы. Эта особенность позволяет производить ИС на МОП-структурах в микроскопическом масштабе с использованием фотографических процессов для нанесения печатного монтажа.

Плотность схем, размещаемых на полупроводниковых кристаллах, невероятно велика. Шмидт и Фарвел [63] сообщают, что в 1983 г. на кремниевом кристалле размером 6,35 мм размещалось до 100 000 транзисторов, а также резисторы и конденсаторы. Технология изготовления интегральных схем, выполненных на кристаллах, предусматривает на первом этапе разработку таких микросхем, а затем производство сотен идентичных кристаллов

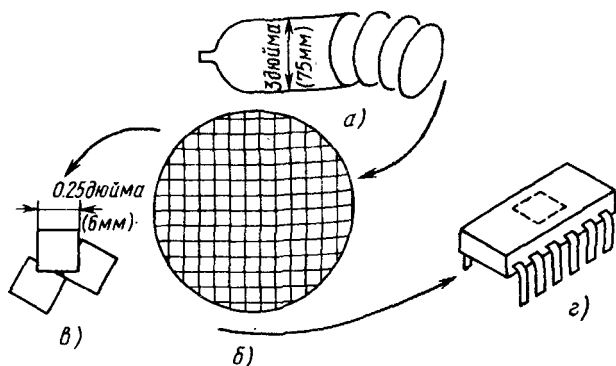


Рис. 14.3
Цикл производства ИС:

а — срезы толщиной 0,01 дюйма (0,25 мм) с кремниевого цилиндра представляют собой круглые плоские и гладкие пластинки; *б* — увеличенный образец кристаллов, выполненных на каждом срезе; *г* — нарезка отдельных ИС; *г* — каждая схема ИС изготавливается отдельно

на кремниевой пластине диаметром примерно 76 мм. Затем производится проверка кристаллов, поскольку практически неизбежно, что многие кристаллы на пластине окажутся бракованными. На рис. 14.2 показан контроль одной микросхемы. При этом контроле каждая из сотен аналогичных интегральных микросхем на кремниевой пластине оказывается неразрушенной. После контроля ИС выполняется разметка пластины и производится резка пластины на отдельные кристаллы. Схема производства пластин, нарезка кристаллов ИС, оформление кристаллов в модули, которые можно подключать к монтажу печатной платы, представлены на рис. 14.3.

Совершенствование технологии производства ИС позволяет существенно сократить их размеры. Ожидается, что совершенствование этого процесса будет продолжаться. В середине 80-х гг. в производстве были освоены *большие интегральные схемы (БИС)*. На кристаллах БИС стали выполняться микропроцессоры. Хотя БИС и отличаются большей компактностью по сравнению с ИС, выпускавшимися ранее, — *малой степени интеграции* и средней степени интеграции, — кристаллы БИС не являются пределом в технологии производства ИС. Уже созданы еще более компактные схемы — *сверхбольшие интегральные схемы (СБИС)* и *интегральные схемы со степенью интеграции выше сверхвысокой* (ХЛСИ)*. Очевидной целью миниатюризации схем является повышение их компактности, с тем чтобы их можно было встраивать в малогабаритные устройства, такие как часы, калькуляторы и некоторые роботы. Однако компактность в действительности не является первоочередной целью. По мере уменьшения габарит-

* Соответствует объему памяти более 1 Мбит — *Прим. перев.*

ных размеров ИС увеличивается их быстродействие и снижается потребление электроэнергии, при этом снижается и их стоимость.

Контроль качества в производстве ИС играет решающую роль. При производстве первых ИС на кремниевых пластинах удавалось получить один годный кристалл на 20 бракованных. Такой низкий уровень качества считался приемлемым, поскольку осуществлялся контроль каждого кристалла. Совершенствование технологического процесса постепенно увеличило производительность при изготовлении пластин в единицах годных кристаллов. Это привело к существенному снижению стоимости ИС и соответственно изделий, выполненных на основе ИС. Повышение производительности с 5 до 50 % приводит к снижению стоимости ИС на 90 %, т. е. в 10 раз. Неудивительно, что электронные калькуляторы, часы и микроЭВМ существенно подешевели в 80-е гг.

Проблемы качества технологии производства ИС продолжают сохраняться и в 80-х гг. При этом, несмотря на совершенствование технологического процесса производства, контроль и отбраковка ИС продолжают осуществляться после их изготовления. Очень существенным моментом является условие недопустимости загрязнения кремния микроскопическими пылинками, что, вероятно, приводит к браку всей ИС. Это обуславливает необходимость производства в очищенных от пыли помещениях, в которых содержание в воздухе загрязняющих частиц составляет менее чем 100 частиц на $0,028 \text{ м}^3$, т. е. воздух должен быть чище в 100 раз, чем в современной больнице.

Трудности изготовления качественных кристаллов тормозят разработку значительно больших по степени интеграции кристаллов ИС, на основе которых могли бы быть созданы ЭВМ большой мощности, возможно на одном кристалле. В журнале «Ньюсвик» было опубликовано сообщение Марбаха, Роджерса и Конанта [39], согласно которому разработчик ЭВМ Джин Амдал занят созданием кристалла ИС размером с пластину. Такая ИС заменит сотни обычных ИС. Чтобы понять важность задачи создания такого кристалла, следует учесть, что «чертеж суперкристалла, на котором мельчайшие компоненты представлены в масштабе 2,54 мм, занял бы площадь размером с два футбольных поля» [39]. Несмотря на очистку воздуха в производственных помещениях и применение современной технологии, производство ИС на кристаллах размером с пластину все еще отличается несовершенством. Для компенсации незначительных дефектов кристаллы размером с пластину содержат резервные схемы, для того чтобы полная схема при наличии таких дефектов оставалась работоспособной.

Перспективы разработки быстродействующих малогабаритных кристаллов ИС с малым энергопотреблением открывают новые возможности для разработки микропроцессоров, предназначенных для роботов и автоматизированного оборудования. Получив

общее представление об ИС, рассмотрим структуру ИС микропроцессора, поскольку последний играет важную роль в автоматизации роботов и производственного оборудования.

14.4. СТРУКТУРА МИКРОПРОЦЕССОРА

Хотя существуют различия между ЭВМ и микропроцессором, в состав последнего должны входить функциональные элементы центрального процессора ЭВМ, ограниченная память, а также средства обмена цифровой информацией между компонентами микропроцессора, между микропроцессором и внешними устройствами (однако не обязательно периферийными устройствами, как это имеет место в микроЭВМ). Рассмотрим структуры компонентов, из которых состоит микропроцессор. Остановимся на особенностях, которые ставят микропроцессор для инженеров по робототехнике и автоматизации производства в ряд важнейших устройств.

Центральный процессор. Основной частью микропроцессора или любой ЭВМ является центральный процессор. В его состав входят цифровые регистры для хранения логических данных в двоичном коде. Эти регистры используются для хранения промежуточных данных при расчетах или обработке логической информации. Один из таких регистров называется аккумулятором. Это название объясняется функциональным назначением регистра, который обеспечивает сложение содержимого одной ячейки памяти с содержимым другой ячейки. Аккумулятор во многих случаях также действует как буферное устройство для ввода данных в двоичном коде в центральный процессор из памяти, в память и при обмене данными с периферийными устройствами. Другим важным регистром является *регистр команд* или дешифратор, который обеспечивает запоминание и интерпретацию выполняемой команды ЭВМ, которая представлена в двоичном коде. Следующими являются *адресные регистры*, предназначенные для обращения к командам программы или данным в памяти.

В состав центрального процессора также входит арифметико-логическое устройство (АЛУ), которое на основе булевой логики выполняет стандартные арифметические операции, такие как сложение и вычитание. Логическая схема достаточно сложна. Однако принципы ее построения соответствуют структуре, описанной в гл. 10.

Наконец, центральный процессор должен располагать некоторыми средствами временной синхронизации для обеспечения координации и управления всеми операциями. Генератор, входящий в состав центрального процессора, может использоваться программистом для синхронизации импульсов, поступающих на вход какого-нибудь регистра. Благодаря этому функции микропроцессора могут быть синхронизированы с функциями внешних устройств на реальном объекте и в реальном времени. Однако

доступ к генератору в большинстве случаев для пользователя ограничен, и в случаях, когда последнему необходим доступ к таким устройствам, применяются специальные кристаллы, на которых реализованы программируемые интервальные таймеры. Реальный генератор в центральном процессоре микропроцессора работает на очень высоких частотах — например, 8 МГц.

Память. Некоторые двоичные регистры используются для хранения двоичной информации двух типов: команд в двоичном коде и данных в двоичном коде.

Регистры для хранения информации обоих типов называются *памятью*. Регистры памяти в большинстве микропроцессоров имеют объем восемь двоичных разрядов. Однако некоторые микропроцессоры имеют 16-разрядные двоичные регистры памяти. Объем регистра памяти соответствует *слову*, которое является основной характеристикой микропроцессора. Когда рассматривается микропроцессор с памятью 64 К слов, то имеется в виду, что такая память соответствует 64000 регистрам. Конечно, память объемом 64 К слов в ЭВМ с 16-разрядными словами в 2 раза превышает объем памяти 64 К слов в ЭВМ с восьмиразрядными словами.

Доступ к регистрам памяти в микропроцессоре осуществляется с меньшей частотой, чем доступ к регистрам центрального процессора, например к аккумулятору. Однако доступ к отдельным регистрам памяти, расположенным в любом месте, должен быть прямым в случае, если последние предназначены для хранения команд программы или данных пользователя. Бесплезно иметь быстродействующий центральный процессор, чтобы он находился в режиме ожидания в течение каждого командного цикла до окончания последовательного поиска требуемых данных или путем опроса 64000 регистров памяти. Гибкость и вычислительная мощность микропроцессора требуют организации доступа к каждому регистру посредством абсолютной адресации вместо последовательного опроса файла. Прямой доступ к регистрам памяти через абсолютные адреса является основной особенностью микропроцессорных ЗУПВ, что означает *запоминающее устройство с произвольной выборкой* или оперативную память. В микроЭВМ во многих случаях требуется большой объем оперативной памяти. При этом кристаллы ЗУПВ подключаются к основному микропроцессорному кристаллу.

Содержимое регистров универсальной памяти изменяется по мере ввода в машину новых программ или данных. На практике это относится в большей мере к микроЭВМ и в меньшей — к микропроцессорам. Однако для управления стандартными обязательными процедурами, применяемыми в любых микропроцессорах, всегда требуются некоторые данные программно-реализуемых команд. При этом целесообразно зарезервировать определенные области памяти для того, чтобы программист по ошибке, вследствие невнимательности, не смог записать в эти регистры памяти

новые команды или данные, нарушающие работоспособность микропроцессора. Для исключения ошибочной записи программистом используется постоянная память, в которую предварительно записываются основные программы. Регистры такой памяти входят в состав *постоянного запоминающего устройства*. Программист лишен возможности записи в постоянную память. Читателю, представляющему различия между микроЭВМ и микропроцессором, нетрудно понять, что постоянная память имеет большее значение для микропроцессора, чем для микроЭВМ. Однако в состав микроЭВМ и микропроцессора должно обязательно входить по крайней мере небольшое ПЗУ. В микроЭВМ процесс перепрограммирования осуществляется неоднократно, для чего используется оперативное запоминающее устройство. В то же время микропроцессор может использоваться для автоматизации одной производственной установки. При этом, поскольку программирование микропроцессора производится лишь однажды за все время службы, постоянная память является единственным приемлемым для таких случаев типом памяти.

Технические требования к микропроцессору, предназначенному для использования в производстве, включают условие возможности изменения содержимого ПЗУ в некоторых пределах. Например, может оказаться желательной возможность программирования микропроцессора не у производителя, а у пользователя. При этом возникает противоречие между поставленным требованием и невозможностью программирования ПЗУ. Указанному требованию перепрограммирования ПЗУ пользователями микропроцессора удовлетворяет память типа ППЗУ — программируемое постоянное запоминающее устройство. Противоречие здесь возникает лишь однажды, поскольку после программирования ППЗУ пользователем последнее становится постоянным.

Однако при использовании в промышленности в некоторых случаях возникает противоречие, заключающееся в том, что при совершенствовании или коррекции технологического процесса, управляемого с помощью микропроцессора, требуется внесение изменений в программное обеспечение, хранящееся в ППЗУ. В соответствии с этим требованием производства было разработано стираемое программируемое постоянное запоминающее устройство — СППЗУ. Цена СППЗУ несколько превышает цену ППЗУ. Однако СППЗУ играют чрезвычайно большую роль при автоматизации производства, позволяя производить отладку программ на языке ассемблера для автоматизированных установок и роботов при очень низких затратах на технические средства. Процедура перепрограммирования кристалла СППЗУ предусматривает снятие ИС с печатной платы и экспонирование кристалла под ультрафиолетовыми лучами высокой интенсивности в течение нескольких минут. При переналадке эксплуатируемого автоматизированного оборудования затраты на перепрограммирование СППЗУ являются значительными. Однако такие затраты включаются в стои-

мость переналадки, а длительность автоматизированного технологического цикла достаточно велика, чтобы оправдать затраты на такую переналадку. Таким образом, СППЗУ применяется в управлении роботами и автоматизированными установками при длительных циклах работы без переналадки. При переналадке в таком случае не требуется замены технических средств, необходимой при жесткой автоматизации.

Существует еще один тип запоминающего устройства, применяемого в микропроцессоре, — электрически перепрограммируемое постоянное запоминающее устройство (ЭППЗУ). ЭППЗУ, отличаясь от СППЗУ значительно большей стоимостью, обладает преимуществом, которое заключается в возможности перепрограммирования кристалла, находящегося на печатной плате. Перепрограммирование ЭППЗУ осуществляется почти столь же просто, как и ЗУПВ. Если ЭППЗУ является стираемым, как и ЗУПВ, то возникает вопрос: почему в таком случае просто не использовать ЗУПВ? Ответ состоит в том, что ЭППЗУ может быть энергонезависимым, т. е. данные, хранящиеся в этой памяти, не могут быть стерты автоматически при отключении напряжения, подаваемого на кристалл ИС. К сожалению, ЗУПВ обычно энергозависимо и, как правило, применяется для хранения данных промежуточных расчетов, а также технологических данных, потеря которых допускается при снятии напряжения питания. Энергонезависимость микропроцессорной памяти является очень большим преимуществом, особенно для хранения управляющих программ или программ постоянных циклов для роботов и автоматизированного оборудования.

Передача данных. Выше рассмотрены особенности микропроцессора и различных типов памяти, применяющейся в микропроцессорах. Однако еще не рассматривались процедуры передачи данных из памяти к регистрам центрального процессора и наоборот, а также обмена данными с внешними устройствами. Очевидно нецелесообразно использовать автономный информационный канал от каждого двухразрядного двоичного регистра запоминающего устройства к двухразрядным регистрам центрального процессора. Поэтому все ячейки памяти и некоторые регистры центрального процессора соединены между собой общими проводниками. Такие общие проводники называются шинами. На первый взгляд такой подход кажется невозможным, поскольку в соответствии с законом элементарной физики одновременно по одному электрическому проводнику может быть передан лишь один электрический импульс (соответствующий одному биту информации). Объяснение применимости упомянутого закона устанавливается через термин *время*. Очень высокая частота генератора центрального процессора позволяет синхронизировать любой импульс генератора точно в определенный момент времени, совпадающий с наличием данных на входах логической схемы И, что приводит к разрешению передачи данных через схему И. Сказанное следует

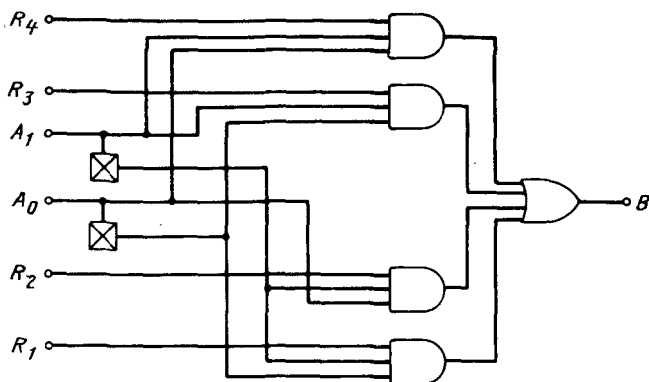


Рис. 14.4

Логическая схема И, используемая для управления передачей данных на общую шину.

непосредственно из принципов, рассмотренных в гл. 10. Возможно использование аналогичной схемы И для управления передачей данных через информационную шину с помощью адресного регистра. Использование одnorазрядной информационной шины, внешней по отношению к микропроцессору, будет проиллюстрировано с помощью упрощенного примера.

Пример 14.1. Управление информационным потоком на информационной шине.

Для выбора одного из четырех битов четырехразрядного двоичного числа, передаваемого от промышленного робота в микропроцессор через одnorазрядную информационную шину, используется двухразрядный адресный регистр. Обозначим два разряда адресного регистра логическими переменными A_0 (младший разряд) и A_1 (старший разряд) и переменными состояний робота от R_1 до R_4 . Обозначим логическое состояние информационной шины B . Для описания взаимосвязи между переменными состояний робота, адресным регистром и информационной шиной воспользуемся выражениями булевой алгебры и логической схемой. Одновременно на этой информационной шине может появляться не более одного бита информации. Определить двоичный и десятичный адреса для каждой переменной состояния робота.

Решение. Уравнение булевой алгебры

$$B = \bar{A}_1 \bar{A}_0 R_1 + \bar{A}_1 A_0 R_2 + A_1 \bar{A}_0 R_3 + A_1 A_0 R_4.$$

	Адрес	
	двоичный	десятичный
R_1	00	0
R_2	01	1
R_3	10	2
R_4	11	3

Логическая схема изображена на рис. 14.4.

Передача данных к периферийным устройствам и от них (ввод/вывод) осуществляется словами. Одновременно передается объем информации в одно слово. Обмен данными с внешними устройствами представляет собой проблему, отличную от передачи

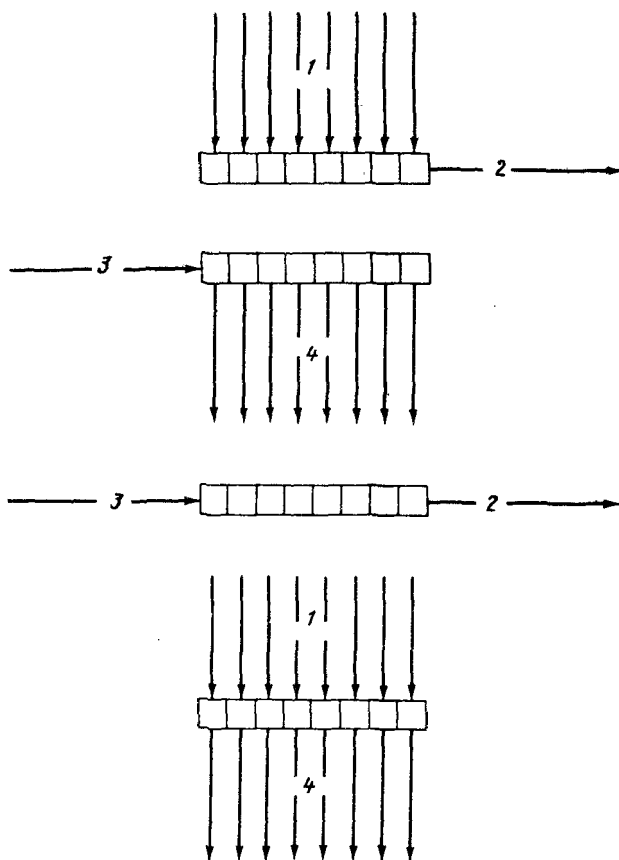


Рис. 14.5

Параллельный и последовательный ввод-вывод данных:

1 — параллельный ввод; 2 — последовательный вывод; 3 — последовательный ввод;
4 — параллельный вывод

данных в пределах устройства, поскольку внешние устройства не синхронизированы с импульсным генератором микропроцессорного устройства. Для ввода-вывода данных словами используется параллельная и последовательная передача. Наиболее просто осуществляется параллельный ввод-вывод данных, поскольку при этом способе для передачи каждого разряда слова используется отдельный проводник. Следовательно, в ЭВМ с восьмиразрядными словами для передачи каждого слова двоичными разрядами необходимы восемь контактных выводов. Наоборот, при последовательном вводе-выводе данных для передачи одного слова требуется лишь один проводник. При последовательном вводе-выводе данных отдельные двоичные разряды передаются последовательно с заданной частотой, что обеспечивает форми-

рование целого слова. Различие между последовательным и параллельным вводом-выводом данных показано на рис. 14.5 для стандартного восьмиразрядного (в один байт) слова.

Имеется полная возможность размещения всех функций ЭВМ, выполняемых описанными выше узлами — центральным процессором, памятью и устройствами ввода-вывода данных, на одном кристалле ИС. Такие ИС играют огромную роль для роботов и при автоматизации производства благодаря большим потенциальным возможностям, малым габаритам и особенно вследствие низкой стоимости.

Инженер по автоматизации, знакомый с принципами работы и программирования таких микросхем, обладает широким кругозором в области автоматизации на современном этапе, представляя ее потенциальные возможности, причем более широко, чем инженер, только использующий руководства по эксплуатации промышленных роботов и автоматизированного оборудования. Рассмотрим принципы программирования таких микропроцессоров.

14.5. ПРОГРАММИРОВАНИЕ

Программирование микропроцессоров, предназначенных для управления промышленными роботами или другим автоматизированным производственным оборудованием, выполняется на языке низкого уровня. Данные и команды, используемые в микропроцессоре, представляются в двоичном коде, что создает определенные трудности при программировании даже для наиболее подготовленных программистов. Применение систем счисления более высокого порядка, как, например, восьмеричной, десятичной или шестнадцатеричной, упрощает процесс программирования. Однако, когда мы говорим «упрощает», то не имеем в виду упрощение, которое обеспечивается применением языков высокого уровня, таких как Бейсик или Фортран. Система (десятичного) счисления с основанием 10 является наиболее известной, и безусловно эта система практически была бы наиболее универсальной системой, если бы число 10 было образовано возведением в целую степень числа 2. В гл. 10 было рассмотрено преобразование из двоичной в десятичную систему счисления. При таком преобразовании учитывается, что число 10 не может быть получено возведением числа 2 в некоторую степень, соответствующую целому числу. Восьмеричная система с основанием 8 является достаточно удобной, поскольку выполняется равенство $2^3 = 8$. Кроме того, восьмеричная система широко применялась в больших ЭВМ, особенно в 70-х гг. Однако язык программирования микропроцессоров базируется на шестнадцатеричной системе счисления. Важная роль, которую играет число 16 при использовании микропроцессоров и в программировании некоторых промышленных роботов, обуславливает необходимость получения некоторого представления о шестнадцатеричной системе счисления инженерам по робототехнике и автоматизации производства.

14.1. Символы шестнадцатеричной системы счисления

Десятичные	Шестнадцатеричные	Двоичные	Десятичные	Шестнадцатеричные	Двоичные
0	0	0	8	8	1000
1	1	1	9	9	1001
2	2	10	10	A	1010
3	3	11	11	B	1011
4	4	100	12	C	1100
5	5	101	13	D	1101
6	6	110	14	E	1110
7	7	111	15	F	1111

Шестнадцатеричная система счисления. Использование шестнадцатеричной системы счисления представляет известные трудности для программиста, обеспечивающего программирование микропроцессора. Однако эта система счисления является не столь сложной, как это может показаться на первый взгляд. Удобство применения этой системы счисления связано с выполнением равенства $2^4 = 16$, которое означает, что любое четырехразрядное двоичное число может быть представлено одноразрядным шестнадцатеричным числом. В этом случае широко применяемые в микропроцессорах слова объемом восьми двоичных разрядов (один байт) могут быть представлены двумя шестнадцатеричными разрядами. В некоторых моделях микропроцессоров все шире начинают применяться двоичные 16-разрядные (двухбайтные) слова, которые достаточно просто представляются в шестнадцатеричном коде, т. е. 16-разрядное слово может быть представлено четырьмя шестнадцатеричными разрядами.

В структуре 16-разрядного числа для представления первых десяти знаков используются обычные десятичные цифры (от 0 до 9). В остальной части структуры используются шесть дополнительных символов в виде первых шести букв латинского алфавита (с A по F). Таким образом, шестнадцатеричная система счисления, которая представляется начинающему программисту столь сложной, на самом деле является простой. В табл. 14.1 приведены шестнадцатеричные символы с эквивалентами в десятичной и двоичной системах счисления.

С помощью табл. 14.1 не представляет трудности преобразовать шестнадцатеричные числа в двоичную систему счисления, используемую в микропроцессоре. Ниже приведены примеры преобразования:

$$2F \text{ (восьмер.)} = 00101111 \text{ (двоичн.)}$$

$$CB \text{ (восьмер.)} = 11001011 \text{ (двоичн.)}$$

Язык ассемблера. Большинство читателей, вероятно, знакомы с одним или более языками программирования, ориентированными на пользователя, как, например, Бейсик или Фортран.

Программы, записанные на этих или аналогичных языках, исполняются последовательно, по одной команде. Порядок исполнения последовательности команд программы может изменяться. При этом во время исполнения программы может быть сформирована управляющая команда, которая может обеспечить изменение последовательности исполнения программы. Однако исполнение программы и в этом случае остается последовательным — не более одной команды одновременно. Независимо от типа используемого языка программирования перед исполнением производится представление всех программ на машинном языке в двоичном коде. Машинный язык также обеспечивает последовательную отработку программы — по одному шагу. Однако один шаг программы, представленной на языке Бейсик или Фортран, соответствует некоторой последовательности шагов программы, записанной на машинном языке. При этом принцип последовательного исполнения программы сохраняется, а программа, написанная на языке Бейсик или Фортран, представляет собой последовательность команд, каждая из которых в свою очередь является последовательностью на машинном языке. Команды на языке Бейсик или Фортран называются «макрокомандами», так как каждая из них включает несколько команд на машинном языке.

Однако большинство микропроцессоров программируется на языке ассемблера. Основное различие между языком ассемблера и языками высокого уровня заключается в том, что команды языка ассемблера связаны с командами машинного языка непосредственно или почти непосредственно в противоположность макрокомандам языков высокого уровня.

Любой язык ассемблера или машинный язык включает следующие элементы.

1. *Адрес команды*, который описывается последовательностью двоичных цифр (на машинном языке) или меньшим количеством шестнадцатеричных (на языке ассемблера); адресом определяется ячейка памяти (ПЗУ или ЗУПВ), соответствующая этой команде. Определенный адрес на языке ассемблера может быть представлен через буквенно-цифровые символы.

2. *Код операции*, представляющий собой небольшие последовательности двоичных цифр (машинный язык) или две-три буквы латинского алфавита (иногда числа), образующие mnemonic код (язык ассемблера), которым определяется функциональное назначение соответствующей команды; код операции грамматически обычно определяется как глагол. Ниже приведены примеры mnemonic кодов * на языке ассемблера: SUB — *subtract* (вычесть); MPY — *multiply* (умножить); STR — *store* (сохранить в памяти).

3. *Операнд(ы)*. Число (числа) как адрес(а), представленные в двоичном коде (на машинном языке) или в шестнадцатеричном

* На русский язык не переводятся; представляют собой аббревиатуру английских слов — *Прим. перев.*

коде (на языке ассемблера), используемые в дальнейшем, определяют функцию команды; обычно операнд может быть интерпретирован как объект, над которым глагол, представленный в коде операции, должен совершить определенное действие. На языке ассемблера операнды могут быть представлены буквенно-цифровым символом вместо абсолютного адреса или числовой константы.

Согласно принятым правилам, эти три компонента, из которых состоит команда, представленная на уровне языка ассемблера, располагаются в указанном порядке в одну строчку. Справа от команды программист может записывать комментарий, предназначенный для пояснения функционального назначения этой команды. Ниже для иллюстрации приведены несколько примеров полных команд.

СНКІ LDA ROBOT — код операции LDA в мнемоническом виде, входящий в состав команды, означает «Загрузить в регистр А». В ячейке памяти, обозначенной символическим адресом ROBOT, содержатся данные, предназначенные для загрузки в регистр А. СНКІ представляет собой символический адрес команды.

SLA 7 — содержимое регистра А смещается влево на семь двоичных разрядов. Стираются семь цифр в старших разрядах. Семь старших разрядов справа заполняются двоичными нулями.

JNZ ROUTI — переход к ROUTI для исполнения следующей по порядку команды при условии, что содержимое регистра А не равно нулю.

EI — разрешение прерывания; эта команда разрешает прервать работу центрального процессора для исполнения специальной операции. Прерывание может произойти в любой момент, когда возникает такая необходимость, подтверждаемая сигналом, сформированным по условию прерывания.

DI — запрет прерывания; центральный процессор при этой команде будет игнорировать прерывания (до тех пор, пока не будет введена команда EI). При этом центральный процессор обеспечивает исполнение некоторых операций с высоким уровнем приоритета.

JMP IDLE — безусловный переход к ячейке памяти, обозначенной символически IDLE, для исполнения следующей команды.

Пример 14.2. Диагностические программы на языке ассемблера. В предыдущем примере рассматривались переменные четырех логических состояний, передаваемые от промышленного робота в микропроцессорное устройство управления. Предположим, что все переменные, которыми описываются четыре состояния робота, соответствуют логической единице. Тогда функции этих переменных определяются следующим образом:

R_1 — кто-то или что-то вмешалось в рабочий цикл робота;

R_2 — рабочая жидкость в гидробаке превысила верхний допустимый предел;

R_3 — разгрузочный желоб для обработанных деталей заполнен и ожидает загрузки с помощью роботизированной разгрузочной станции;

R_4 — разгрузочный бункер (вибрационный бункер), предназначенный для деталей, загружаемых с помощью робота, пуст.

Четыре бита информации, поступающих от робота, представляют собой полубайт. Предположим, что эти данные хранятся в ячейке ЗУПВ микропроцессора, которая символически обозначена ROBOT. Далее предположим, что в микропроцессоре используются слова длиной 1 байт и что интересующий нас полубайт находится в половине байта, соответствующей младшим разрядам. Напишите программу на языке ассемблера для проверки каждого условия робота от R_1 до R_4 с переходом к диагностическим программам с ROUT1 до ROUT4 соответственно, обслуживающим указанные условия по требованию.

Решение. Присвоим этой программе, написанной на языке ассемблера, имя TEST. Каждый раз, когда требуется проверить входы от R_1 до R_4 на наличие на них условий аварийности, эта программа может быть направлена для исполнения в ячейку памяти TEST. Предположим, что каждая диагностическая программа от ROUT1 до ROUT4 имеет соответствующий выход или вход для проверки других условий, обуславливающих аварийное состояние, или на ожидание возможных прерываний по требованию. Ниже приведена программа на языке ассемблера с комментариями; эта программа обеспечит исполнение диагностических программ от ROUT1 до ROUT4 по требованию:

TEST DI	— запрет прерываний. Во время обслуживания аварийных условий от R_1 до R_4 не должно быть никаких прерываний работы центрального процессора.
CHK1 LDA ROBOT	— загрузка содержимого ячейки ЗУПВ с адресом ROBOT в регистр A.
SLA 7	— стирание информации и старших разрядах.
JNZ ROUT1	— если содержимое регистра A не равно нулю ($R_1 \neq 0$), перейти к диагностической программе ROUT1.
CHK2 LDA ROBOT	— повторная загрузка содержимого ячейки ROBOT в регистр A.
SRA 1	— стирание условия R_1 .
SLA 7	— стирание информации в старших разрядах для выбора R_2 .
JNZ ROUT2	— переход к программе ROUT2 при условии $R_2 \neq 0$.
CHK3 LDA ROBOT	— загрузка содержимого ячейки ROBOT обратно в регистр A (повторно).
SRA 2 }	выбор условия R_3 .
SLA 7 }	
JNZ ROUT3	— переход к программе ROUT3 при условии $R_3 \neq 0$.
CHK4 LDA ROBOT	— загрузка содержимого ячейки ROBOT в регистр A (повторно).
SRA 3 }	выбор условия R_4 .
SLA 7 }	
JNZ ROUT4	— переход к программе ROUT4 при условии $R_4 \neq 0$.
EI	— запрет прерывания.
IDLE LDA ROBOT	— в действительности это фиктивная программа, предназначенная для загрузки микропроцессора во время ожидания какого-нибудь прерывания.
JMP IDLE }	

Как можно видеть из примера 14.2, программирование на языке ассемблера требует детализации и является утомительной процедурой по сравнению с языками Бейсик и Фортран. Однако манипуляция с логическими двоичными разрядами представляет собой связанную детализированную процедуру. Хотя для обработки данных на ЭВМ удобно применять языки компилятора, любой значительный отход от двоичной системы счисления может создать такое же количество проблем, как и при обработке логических переменных в случаях использования промышленных роботов и автоматизированного производственного оборудования. Кроме того, язык ассемблера в ряде случаев может обеспечить

некоторую экономию вычислительной мощности ЭВМ вследствие близости к машинному языку.

Операторы языка ассемблера для микропроцессоров, представленные выше, являются примерами операторов, используемых в языках ассемблера для всех микропроцессоров. Инженер по автоматизации, приобретя опыт программирования одного микропроцессора, сможет достаточно просто запрограммировать и другие, пользуясь справочной таблицей команд, относящейся к конкретному микропроцессору, подлежащему программированию, поскольку между языками ассемблера существуют лишь незначительные различия. Здесь не ставится цель подробно рассмотреть все различия, существующие между языками микропроцессоров, или даже изучить язык одного микропроцессора, поскольку в этом случае будет обеспечена зависимость читателя от конкретного микропроцессора. Целью главы скорее является показать инженеру по автоматизации необычайные потенциальные возможности микропроцессоров, а также раскрыть некоторые их конструктивные особенности, особенности их программирования и использования. Ознакомившись с основными принципами микропроцессоров, рассмотрим направления, в которых происходит внедрение микропроцессоров в промышленные роботы и в автоматизированное производственное оборудование.

14.6. ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРОВ

Применение микропроцессоров в автоматизации идет в трех направлениях: основное оборудование, микроЭВМ и специализированные устройства управления. Дадим соответствующие пояснения по каждому из этих направлений.

Основное оборудование. Термин «основное оборудование» звучит противоречиво по отношению к «применению» микропроцессоров. Однако этот термин широко используется в области промышленной автоматизации. Инженер по робототехнике и автоматизации должен осознавать свое место в процессе автоматизации. Снижение стоимости и удивительные возможности микропроцессора делают последний весьма привлекательным устройством при создании нового оборудования. В этом направлении применения микропроцессоров инженер-конструктор рассматривает все режимы работы автоматизированной системы и все периферийное оборудование, с которым эта система должна взаимодействовать. Программирование практически всегда ведется на языке ассемблера, а основная память является энергонезависимой, обычно ПЗУ или ППЗУ. Затраты на программирование на языке низкого уровня легко оправдываются большим количеством одинаковых микропроцессоров, представленных для одной модели автоматизированного оборудования.

Не представит трудности дать примеры применений микропроцессоров в основном оборудовании среди промышленных

изделий. Микроволновые печи, автомобили, часы, стереосистемы и детские игрушки — все эти изделия используют интеллектуальные возможности ЭВМ, реализованные на дешевых микропроцессорах. Производитель промышленного производственного оборудования имеет большие доходы при реализации оборудования, оснащенного микропроцессорами. Хотя сам по себе кристалл имеет низкую стоимость, добавление его к крупному дорогому промышленному оборудованию столь значительно повышает ценность последнего, что производители смогут получать большие прибыли путем встройки микропроцессора в основное оборудование. Примерами промышленного оборудования, функциональные возможности которого резко возрастают при встройке в них микропроцессоров, являются промышленные таймеры, станки со сложными диагностическими системами, специализированное тестовое оборудование и лабораторные приборы. Появились небольшие внедренческие фирмы, созданные инженерами по автоматизации. Эти инженеры являлись специалистами как в технологии производства, так и в программировании и применении микропроцессоров, что помогло решить соответствующие проблемы производства. Микропроцессоры начинают находить все большее применение при модернизации устаревшего производственного оборудования, таким образом помогая ускорить освоение новой технологии.

Часто опытный инженер по автоматизации действительно осознает, что требования производства слишком далеки от новой технологии, чтобы использовать ее потенциальные возможности. Молодые высококвалифицированные программисты во многих случаях не понимают реальных требований производства и тратят время на разработку игр, реализуемых на микропроцессоре, с тем чтобы продемонстрировать свои высокие способности в программировании и удивительные возможности микропроцессора. Когда высокие способности программиста и возможности микропроцессора объединяются, потенциальные возможности микропроцессора становятся наиболее эффективными. Если сюда присоединяется партнер с капиталом и интуицией к рынку сбыта, то создается новый бизнес с достаточно приемлемой эффективностью.

Наиболее крупным недостатком использования микропроцессоров производителями основного оборудования является то, что в этом случае может быть неправильно выбрана стратегия управления и программирования. Для разработчика автоматизированного оборудования представляет большие трудности прогнозировать все варианты использования оборудования, а также квалификацию и данные операторов. Пожалуй, никогда в истории человеческий фактор и эргономика при разработке конструкций не имели столь большого значения. Чем сложнее становятся системы управления, тем менее доступны они для понимания пользователем, хотя должен иметь место совершенно обратный

процесс. Инженеры-конструкторы слишком часто разрабатывают оборудование, которое кажется им достаточно простым для понимания, поскольку они затратили несколько месяцев на изучение такой системы и ее характеристик. И все же такое оборудование достаточно сложно для понимания пользователем. Разработке системы, удобной для пользователя, способствуют сегодня большая вычислительная мощность и низкая стоимость микропроцессоров, что позволяет производить разработку диагностических программ, охватывающих всю систему.

Применение микроЭВМ. Использование микропроцессоров в качестве основы микроЭВМ и системы обработки данных сильно отличается от применения микропроцессоров в основном оборудовании. Для микропроцессоров, как и для микроЭВМ, обычно требуется большой объем памяти ЗУПВ. Микропроцессор, аналогично микроЭВМ, поддерживается обычным программным обеспечением, как правило записанным на гибком магнитном диске. Во многих случаях программирование пользователем микроЭВМ выполняется на языке Бейсик или на другом языке высокого уровня, но не на языке ассемблера. ППЗУ обычно используется для хранения операционных систем, интерпретаторов или компиляторов, а не программ для автоматизации, написанных пользователями.

При использовании микропроцессора в качестве вычислительной системы целесообразнее представлять его в качестве ЭВМ, а не ИС. Программы, которые могут быть реализованы на микроЭВМ, являются весьма разнообразными и могут быть представлены в большом количестве. МикроЭВМ может быть использована для непосредственного управления производственным процессом в реальном времени или для расчета оптимальных параметров процесса не в реальном времени. Использование микропроцессоров в качестве микроЭВМ аналогично применению ЭВМ в автоматизации производства, что показано в гл. 13.

Специализированные устройства управления. Третье направление использования микропроцессоров при автоматизации находится где-то между границами двух рассмотренных выше. Это направление предлагает наибольшие потенциальные возможности, которые может использовать инженер по автоматизации производства, эксплуатирующий заводское оборудование, в части внедрения автоматизации и снижения затрат на производство, по мере того как возникают соответствующие технические идеи. Программирование в этом случае может производиться на языке Бейсик или языке технолога типа FORTH. Обычно хранение программы производится в ЭППЗУ или СППЗУ. При этом микропроцессор является основой специализированного устройства. Остается возможность дальнейшей модификации или полного изменения конфигурации программного обеспечения специализированного микропроцессорного устройства. Специализированная микропроцессорная система может не оснащаться периферий-

ными устройствами ввода-вывода данных, как, например, клавиатуры пультов управления, дисплеи или печатающие устройства, а также системами на основе микроЭВМ.

Раньше или позже при возрастании степени сложности оборудования инженер по робототехнике и автоматизации производства становится неудовлетворен приобретением оборудования с жестким циклом автоматизации. Такое оборудование или не соответствует требованиям, разработанным инженером по автоматизации, или располагает слишком большим набором функций, обуславливая необходимость покупки дополнительных технических средств. Инженер, решающий сложные проблемы автоматизации, знает долю прибыли пользователя основного оборудования, выполненного на основе микропроцессоров, в случае, когда микропроцессор используется в специализированном устройстве управления. Инженер по автоматизации представляет перспективы развития микропроцессорной техники и знает, каким образом наиболее эффективно использовать роботы и автоматизированное оборудование. Таким образом, настоящая глава является практически заключительной в нашей книге. В гл. 15 исследуются вопросы этики автоматизации и рассматриваются перспективы автоматизации.

В ы в о д ы

Микропроцессор является основным элементом в конструкции автоматизированного производственного оборудования, хотя он может показаться не столь значительным из-за малых размеров и ненаглядного функционального назначения. Однако знание роботов и автоматизированного производственного оборудования инженером не является полным до тех пор, пока он не изучит конструкцию, программирование и практическое применение этих малогабаритных ИС. Увеличение функциональных возможностей и снижение стоимости ИС за последние два-три десятилетия явились самым большим технологическим прорывом в нашем веке. Неудивительно, что роботы и автоматизированное оборудование, основа технологического развития, находятся в стадии интенсивного совершенствования.

В состав любой ЭВМ входит центральный процессор, память определенного объема и средства для передачи данных. Память может быть энергозависимой и энергонезависимой, для записи-считывания (ЗУПВ), только считывания (ПЗУ), а также промежуточной между двумя последними (ППЗУ, СППЗУ и ЭППЗУ). Память каждого из этих типов находит применение в определенных случаях в роботах и производственных автоматизированных установках.

Передача данных осуществляется посредством информационных шин с использованием логических схем И, принципы работы которых рассмотрены ранее. Передача данных к периферийному

оборудованию и обратно от последнего к микропроцессору осуществляется последовательно или параллельно через блоки ввода-вывода данных.

Инженеры по автоматизации должны иметь представление о программировании микропроцессоров, однако это не означает, что они должны быть специалистами по программированию на языке ассемблера. Программирование на языках ассемблера применяется в автоматизированном оборудовании, и инженер по автоматизации должен иметь это в виду. Однако на практике программирование производится на машинном языке и должно выполняться на машинном языке конкретного микропроцессора с использованием соответствующего руководства по программированию. В настоящей главе представлены несколько примеров программ с кратким пояснением для программирования на языке ассемблера — в шестнадцатеричной системе счисления.

Микропроцессоры находят применение, с одной стороны, в основном оборудовании, а с другой — в системах на основе микроЭВМ. Между этими типами оборудования находится третий тип — специализированные устройства управления, которые располагают максимальными потенциальными возможностями для увеличения производительности и автоматизации заводского оборудования. Образование инженера по автоматизации не считается законченным до тех пор, пока он не сможет вести разработку под заказчика специализированных роботов или устройств управления. Такая разработка необходима для решения проблем с использованием микропроцессоров на собственных предприятиях.

У п р а ж н е н и я и в о п р о с ы

14.1. Объясните различие в значениях терминов микроЭВМ и микропроцессор.

14.2. Определенное совершенствование технологического процесса производства пластин, на которых размещается 200 ИС, позволяет сократить стоимость производства пластин на 30 % и повысить производительность выпуска пластин с 5 до 80 %. В какой степени такое совершенствование повлияет на стоимость ИС?

14.3. Предположим, что ИС в вопросе 14.2 были разработаны для использования в устройстве управления промышленным роботом. Стоимость этих ИС составляла 60 % от 300 долл. стоимости устройства управления до модернизации производства. Какую долю в процентах от полной стоимости устройства управления этим роботом можно было бы отнести к ИС, если бы стоимость других компонентов устройства управления осталась бы без изменений? Предположим, что 30 % стоимости устройства управления составляет прибыль. Насколько могла бы быть уменьшена цена устройства управления робота после модернизации технологического процесса производства ИС, если бы прибыль осталась постоянной на уровне 30 %?

14.4. Некоторый микропроцессор оснащен ПЗУ с адресным пространством от 0000 до 9000 и ЗУПВ с адресным пространством от 9001 до FFFF. Каков объем памяти этого микропроцессора и каким образом распределяется объем памяти между ПЗУ и ЗУПВ?

14.5. Проведено исследование работы промышленного робота. При этом должно быть произведено сравнение двух альтернативных вариантов. В первом

варианте используется промышленный робот стоимостью 50 тыс. долл. Этот робот оснащен универсальной микроЭВМ. В ней в качестве стандартного средства программирования применяется язык высокого уровня, например ВАЛ. В другом варианте стоимость устройства составляет 30 тыс. долл., включая микропроцессорное устройство управления, в котором ЭПЗУ может быть запрограммировано на языке ассемблера. Ожидается, что разработка программы на языке ассемблера составит 20 тыс. долл., что в 5 раз превышает стоимость разработки программы для промышленного робота. При равенстве производительности и идентичности других характеристик у роботов в обоих вариантах, какому из них вы отдали бы предпочтение, если бы условия эксплуатации оставались неизменными неопределенное время? Какой из двух вариантов вы предпочли, если бы ожидаемые условия эксплуатации требовали перепрограммирования памяти примерно один раз в год?

14.6. Максимальная константа, которая может храниться в памяти некоторого микропроцессора, представляет собой десятичное число 255. Приведите эквивалент этого числа в шестнадцатеричной системе счисления. Какова длина слова в этом микропроцессоре? Максимальная длина адреса памяти этого микропроцессора составляет FFFF. Каков объем памяти микропроцессора в битах, байтах, полубайтах?

14.7. Некоторый микропроцессор оснащен интерфейсным устройством с параллельным вводом-выводом данных. Какое количество контактных выводов требуется для передачи данных с максимальным значением до числа FF, представленного в шестнадцатеричной системе счисления? Какое количество выводов потребуется при использовании интерфейсного устройства с последовательным вводом-выводом данных?

14.8. С помощью автоматизированной системы, выполненной на основе микропроцессора, обеспечивается непрерывный контроль восьми логических входных сигналов, поступающих от робота. Восемь входных сигналов поступают параллельно, по одному полубайту одновременно через четырехразрядный двоичный регистр. Этот регистр управляется от микропроцессора, который обеспечивает каждые 0,5 с попеременную передачу входных сигналов от 1 до 4 и от 5 до 8. Операция осуществляется микропроцессором с помощью выходной логической переменной (P), которая устанавливает логическую 1, когда появляются сигналы на входах 1—4 и логический 0, когда сигнал приходит на входы 5—8. Обозначим четырехразрядный двоичный регистр логической переменной A_1 — A_4 (соответственно от младшего до старшего разряда) и входными переменными от I_1 до I_8 (соответственно от младшего до старшего разряда)

а) Напишите логические выражения на языке булевой алгебры для переменных A_1 — A_8 в функции соответствующих входов I_1 — I_8 и P .

б) Нарисуйте логическую диаграмму, которая иллюстрирует работу микропроцессорной системы с параллельным вводом данных.

в) При $P = 1$ содержимое входного регистра соответствует шестнадцатеричному числу C; при $P = 0$ содержимое входного регистра соответствует шестнадцатеричному числу 9. На каких входах промышленного робота присутствует логический сигнал?

14.9. Содержимое регистра объемом 1 байт, предназначенного для параллельного ввода-вывода данных, включает шестнадцатеричное число FB. В скольких из восьми двоичных разрядов регистра ввода-вывода данных будет присутствовать логическая единица?

14.10. В пример 14.2 введем предположение относительно распределения приоритетов диагностических условий R_1 — R_4 в следующем порядке. Предположим, что оба условия «разгрузочный желоб заполнен деталями» и «подающий загрузчик пуст» требуют более неотложного исполнения, чем условие «температура гидробака превышает допустимый уровень». Отредактируйте программу, представленную на языке ассемблера в примере 14.2, с учетом измененных приоритетов.

14.11. Максимальный адрес памяти в некотором микропроцессоре соответствует шестнадцатеричному числу FFFF. Какой адрес следует перед 5000? (Ответ не соответствует числу 4999).

14.12. В памяти некоторого восьмиразрядного микропроцессора хранятся команды для реализации следующих подпрограмм:

<i>Символическое имя</i>	<i>Адрес памяти</i>	
TEST	START	END
ROBOT	9F99	A007
SHK	A022	A100
TOOL	0050	0A00
	BE12	BF00

Какая подпрограмма имеет наибольшую длину, а какая — наименьшую? Каков ее объем (в байтах)? Дайте перечень подпрограмм в той последовательности, в какой они используются в памяти микропроцессора.

15. ПЕРСПЕКТИВЫ И ЭТИЧЕСКИЕ ПРОБЛЕМЫ АВТОМАТИЗАЦИИ

Роботы, микропроцессоры, ПК и другие средства автоматизации не просто внесли изменение в технологические процессы и сделали предприятия более эффективными; они изменили и продолжают изменять всю нашу *жизнь*. Конструирование различных изделий, их производство, распределение и *даже потребление* в корне меняется благодаря тем переменам, которые влечет за собой автоматизация. В последней главе книги мы рассмотрим некоторые новые понятия и философские проблемы управления производством, связанные с этими изменениями. Такие понятия, как гибкие производственные системы (ГПС), групповая технология (ГТ) и производство с комплексным управлением от ЭВМ (ПКУ) привлекли к себе особое внимание специалистов промышленности и университетских ученых, которые начали понимать потенциальные возможности роботов и гибких автоматизированных систем.

Наряду с обсуждением современных направлений развития производственных систем, мы в данной главе ненадолго задержим внимание на продолжающихся исследованиях в области робототехники, в частности на самовоспроизводящихся системах, в которых одни роботы производят поколение других роботов. Глава заканчивается обсуждением этических проблем автоматизации, где мы попытаемся ответить на вопрос, действительно ли создаваемые нами роботы и системы автоматизации производства служат удовлетворению человеческих потребностей, а не каким-то другим целям.

15.1. ГИБКИЕ ПРОИЗВОДСТВЕННЫЕ СИСТЕМЫ (ГПС)

«В сегодняшнем обществе Баскина — Роббинса
все модные вещи имеют по крайней мере
31 оттенок»

Эти слова принадлежат Джону Нейсбитту и заимствованы из его известной книги [49]. Нейсбитт продолжает: «В Соединенных Штатах продаются 752 различные модели легковых и грузовых автомобилей — и это не считая цветов окраски, в которых они выпускаются. Если Вам нужна особо компактная модель, Вы можете выбрать из 126 различных типов. В Манхеттене нахо-

дится магазин, где продаются исключительно светильники; их ассортимент насчитывает 2500 различных типов. Наиболее экзотический светильник поступает из Финляндии, излучаемый им свет можно сравнить с сиянием солнца. Сегодня в США продается более 200 марок (или сортов, как их называют в промышленности) сигарет».

То, что заметил Нейсбитт, — это огромное разнообразие вариантов, предлагаемых потребителю в 80-е гг., что общеизвестно, но такое положение создалось не вдруг, и для большинства это было предметом для размышлений. А вот чего не сформулировал Нейсбитт — это *почему* появилось это разнообразие. Сейчас видно, что оно обязано своим появлением на свет гибкому производству, ставшему возможным благодаря компьютеру и микропроцессору. И хотя у нас очень богатый выбор, мы постепенно привыкаем к такому разнообразию, а это в свою очередь стимулирует производство к дальнейшему созданию все новых и новых образцов товаров.

Жесткая автоматизация 40—50-х гг. была идеальной для своего времени формой производства, ибо почти каждому нужно было одно и то же: новый «Форд» или «Шевроле», лампу дневного света, радиоприемник или телевизор. Такая обстановка диктовала устойчивую потребность в огромных объемах однотипных изделий — объемах, которые послужили стимулом для жесткой автоматизации. Сегодня, однако, рынок потребительских товаров гораздо сильнее нуждается в гибком производстве, обеспечивающем выпуск большого числа разновидностей изделий, моделей, фасонов.

В этой книге основное внимание сконцентрировано на гибкой автоматизации, противопоставляемой жесткой автоматизации. Способность микрокомпьютера обеспечить эту гибкость заставляет владельцев предприятий по-иному взглянуть не только на отдельные свои станки, но и на весь комплекс оборудования. Занимаясь техническим переоборудованием предприятий, они хотят теперь, чтобы все станки, роботы и средства транспортирования материалов были объединены в *гибкую производственную систему*. Ключевым словом здесь является слово «система». Гибкость такой системы обеспечивается благодаря легкости перенастройки роботов, станков с ЧПУ, промышленных ПК и микропроцессоров.

Прежде чем продолжить изложение, необходимо еще и еще раз подчеркнуть, что создание ГПС возможно только при условии применения роботов и подсистем автоматики, рассмотренных в предыдущих главах книги. А предпосылками для применения роботов и подсистем автоматики являются контроль отклонений качества изделий, обеспечение их ориентации и транспортирование, повышение надежности в эксплуатации установок и механизации рабочих мест. Поэтому прежде чем бросаться в омут ГПС, инженер должен быть уверен в том, что он действительно достигнет поставленных целей с помощью заводских агрегатов,

роботов и подсистем автоматизации производства и транспортирования материалов.

Разрабатывая проект автоматизации производства на предприятии, полезно мысленно представить себе общую картину полностью автоматизированной ГПС. Это заставит проектировщика подойти к обеспечению гибкости отдельных рабочих мест с точки зрения экономической целесообразности. Именно такое предвидение заставляет инженера выбирать универсальный робот для выполнения единственной операции вместо специализированного манипулятора, если это является экономически выгодным. Подобной стратегии придерживалась компания «Зингер» при автоматизации производства, о чем шла речь в гл. 8. Возможной, почти эквивалентной по стоимости, альтернативой здесь было применение специализированного манипулятора. Однако вместо этого был выбран универсальный робот, поскольку фирма имела намерение в будущем создать единую гибкую производственную систему.

На данном этапе развития гибкие производственные системы представляют собой скорее общую идею, нежели реальность. Но имеются и примеры действующих ГПС, особенно в сфере производства ЭВМ и электронных устройств. ГПС также успешно завоевывают позиции в производстве металлоизделий, когда каждая модель выпускается небольшими партиями. Примером может служить производство вентиля и трубопроводной арматуры. Такие отрасли промышленности располагают большим числом станков с ЧПУ и, следовательно, поддаются организации в гибкие производственные системы. Но в середине 80-х гг. предприятия все же делают упор на частичную, а не на полную автоматизацию. Широкая общественность не имеет представления о том, что почти на всех предприятиях уже применяются ПК и другие логические системы управления, наподобие тех, которые были рассмотрены выше. Некоторые из этих предприятий располагают станками с ЧПУ, немногие — промышленными роботами и очень малая часть — полностью автоматизированными ГПС.

Полезно упомянуть несколько типичных примеров ГПС, которые инженеры по автоматике смогут использовать в качестве образцов для подражания, когда наступит время им самим заниматься созданием аналогичных систем.

На одном из заводов фирмы «Дженерал Моторс» была введена в строй ГПС, в которой 22 робота обслуживают четыре линии станков, связанные конвейером [22]. Изделия — массивные, почти 14-килограммовые коробки скоростей из алюминия, — типичны для применения ГПС. Эта система, занимающая площадь около 22 000 м², изображена в плане на рис. 15.1. ГПС позволили избавиться от 91 % объема тяжелого ручного труда, применявшегося прежде в этом цехе. Для ГПС были выбраны роботы, которые снимают изделие с конвейера и устанавливают его на станок, а после обработки возвращают обратно на конвейер;

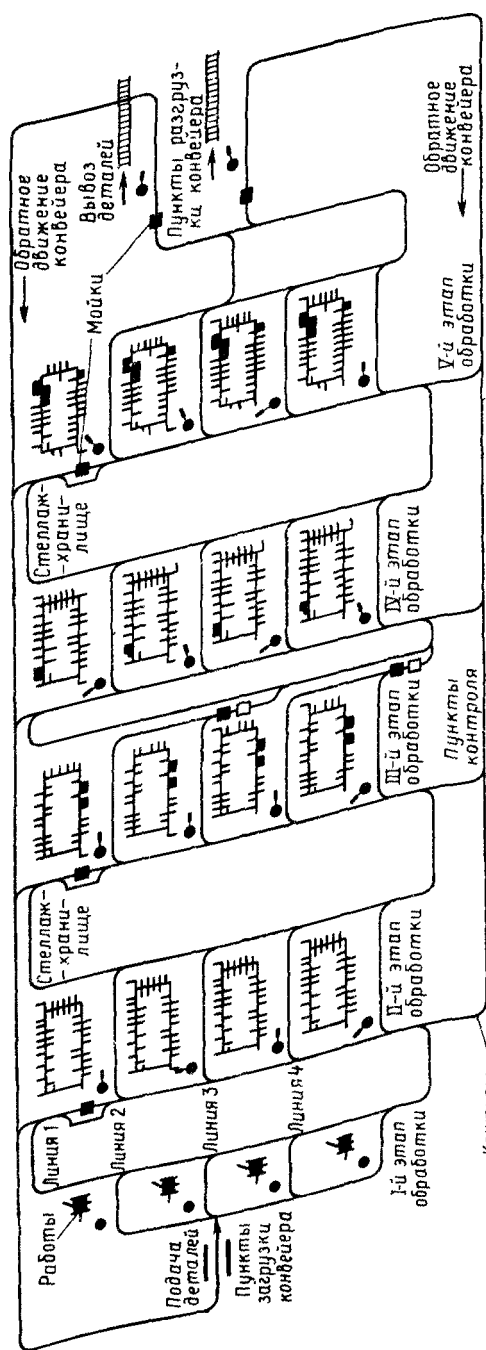


Рис. 15.1
ГПС на заводе фирмы «Дженерал Моторс»: 22 робота обслуживают четыре линии станков, связанные конвейером

благодаря дешевизне таких роботов ожидалось, что они окупят себя всего за один год.

Ради получения экономической выгоды компания «Дженерал Электрик» не пожалела 16 млн. долл., вложенных ею в реконструкцию электро-возостроительного завода в г. Эри, шт. Пенсильвания [23]. Система (рис. 15.2) предназначалась для транспортирования корпусов электродвигателей — крупногабаритных, массивных (от 70 кг до 1 т) металлических отливок вдоль довольно протяженной последовательности обрабатывающих станков.

ГПС компании «Дженерал Электрик» была создана на основе челночной тележки, перемещающейся по колее длиной 64 м. Тележка управляется с помощью ПК под общим наблюдением центрального компьютера. При движении тележка отклоняет специальные планки, каждой из которых присвоен двоичный номер. Фотоэлектрические датчики фиксируют эти номера и таким образом проверяют положение тележки. Сообщается, что данная ГПС позволила на 87,5 % сократить время обработки корпуса электродвигателя при общем росте производительности труда на 240 %. Одновременно на 38 % был увеличен выпуск продук-

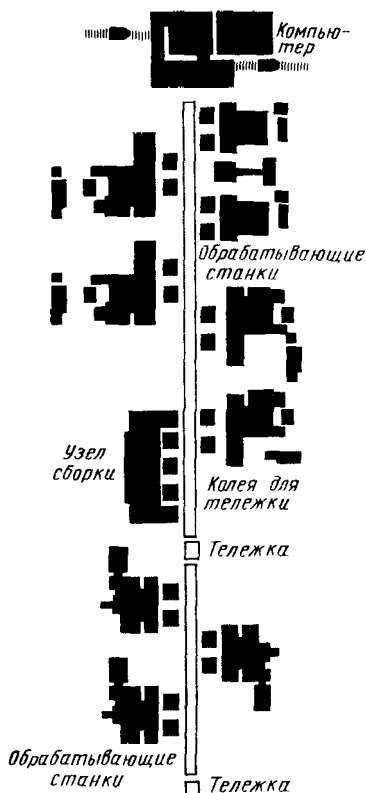
Рис. 15.2

ГПС компании «Дженерал Электрик»: обработка корпусов электродвигателей обеспечивается с помощью челночной тележки для транспортирования изделий

ции при сокращении рабочих площадей на 25 %.

Стохастическое производство. Гибкая автоматизация приняла совершенно неожиданный поворот, при котором сама задача стала частью ее решения. Напомним, что поводом для создания ГПС было стремление приспособить одну и ту же систему для производства широкого ассортимента изделий, выпускаемых мелкими партиями. Такое стремление заставляет выбирать очень гибкие приспособления для обрабатывающих станков, которые сами по себе имеют различную степень гибкости. Однако, если ГПС вступила в строй, то смена выпускаемых изделий и беспорядочное чередование моделей могут, как это ни покажется странным, даже *повысить* эффективность системы. Проиллюстрируем это на примере.

Пример 15.1. Эффективность стохастического производства с ГПС. В данном упрощенном примере некоторая гибкая производственная система может получать заказы торговых предприятий и изготавливать любое из трех изделий А, В и С в любом требуемом порядке. Система состоит из фрезеровального станка, токарного обрабатывающего центра с ЧПУ и пункта автоматического кон-



15.1

Параметр	Изделия		
	А	В	С
Предполагаемый объем производства, шт.	100	50	25
Время обработки (включая транспортирование), мин:			
фрезеровальный станок	8	30	6
токарный центр с ЧПУ	22	3	16
пункт контроля	6	5	25

троля. Время обработки каждого изделия на каждом месте (станке) приведено в табл. 15.1. Время транспортирования материалов и заготовок можно считать ничтожно малым по сравнению с временами обработки, указанными в таблице.

а) Сколько времени потребуется для выпуска запланированного количества изделий при традиционной организации производства (т. е. сначала выпускаются все 100 изделий *A*, затем все 50 изделий *B* и, наконец, все 25 изделий *C*)?

б) Предположим, что три изделия выпускаются в смешанном порядке согласно последовательности $[A-B-A-C-A-B-A] \dots [A-B-A-C \dots]$ и т. д. Сколько времени потребуется в этом случае для выпуска запланированного количества изделий *A*, *B* и *C*?

Решение. Логично предположить, что все три станка в системе могут работать одновременно. Однако каждый станок в данный момент может обрабатывать только одно изделие. При этих условиях суммарное время производства всех изделий по традиционной схеме (а) определяется следующими расчетами, где через A_i обозначен *i*-й экземпляр изделия *A*:

	Время (мин)
A_1 : фрезеровальный станок	8
A_1-A_{99} : токарный центр с ЧПУ 99×22	2178
B_1-B_{50} : фрезеровальный станок 50×30	1500
C_1 : фрезеровальный станок	6
C_1 : токарный центр с ЧПУ	16
C_1-C_{25} : пункт контроля 25×25	625
Итого:	4333

Заметим, что здесь были опущены следующие операции, как выполняемые параллельно с остальными и, следовательно, не влияющие на общее время производства: A_2-A_{100} : фрезеровальный станок; A_{100} : токарный центр с ЧПУ; A_1-A_{100} : пункт контроля; B_1-B_{50} : пункт контроля; C_2-C_{25} : фрезеровальный станок; C_2-C_{25} : токарный центр с ЧПУ.

Производство тех же изделий в смешанном порядке по схеме (б) должно привести к более эффективному использованию ГПС благодаря сбалансированной загрузке станков. На рис. 15.3 представлена временная диаграмма, позволяющая спланировать машинные операции для чередования изделий в порядке $A-B-A-C-A-B-A$. Заметим, что производственный цикл начинает повторяться с момента $t = 136$. Таким образом, время выполнения цикла $A-B-A-C-A-B-A$ при непрерывном производстве будет равно 136 мин. Чтобы выпустить все изделия в полном объеме, потребуется 25 таких циклов. В заключительном цикле дополнительно потребуется еще 14 мин для выполнения завершающих операций над последним экземпляром изделия *A* (A_{100}); таким образом, время этого цикла увеличится до 150 мин. Общее время производства по схеме (б) составит, мин,

$$24 \times 136 + 150 = 3414.$$

Экономия времени по сравнению с выпуском продукции партиями будет равна

$$4333 - 3414 = 919 = 15,5 \text{ ч.}$$

Это соответствует повышению производительности на $(919/4333) \times 100 \% = 21\%$. В рассмотренном примере есть одна тонкость, на которую следует обратить внимание. Надо помнить, что в производственных системах данного типа очередное изделие не может поступить на станок до тех пор, пока он не закончит обработку предыдущего изделия и не передаст его на другую операцию. В случаях, когда станок еще занят обработкой изделия, это очевидно, но даже если обработка уже закончена, предыдущее изделие не обязательно будет передано на следующую операцию, так как там еще могут вестись работы. Читатель легко узнает здесь ситуацию, описанную в гл. 4 и охарактеризованную понятием «блокировка оборудования». С блокировкой оборудования необходимо считаться при проектировании ГПС без промежуточных складов продукции между отдель-

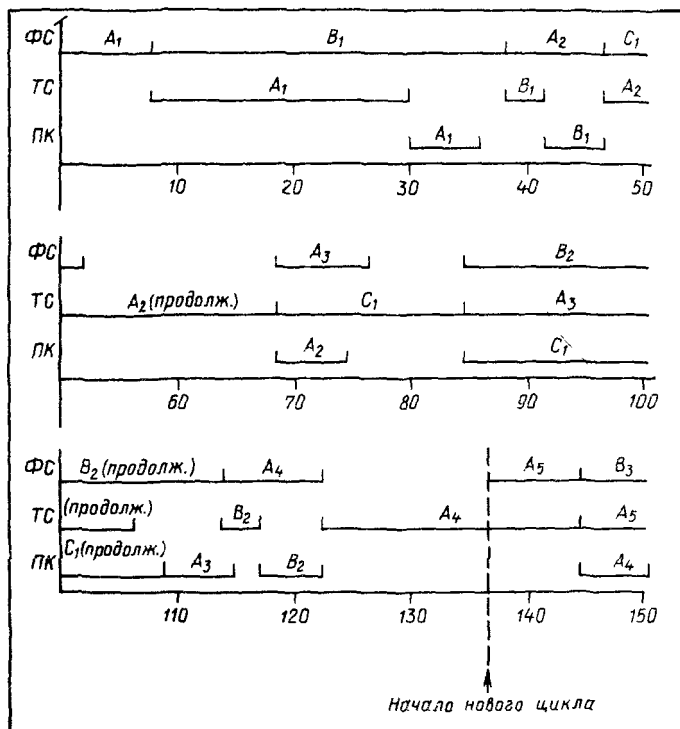


Рис. 15.3

Планирование работы ГПС при выпуске изделий в последовательности [A—B—A—C—A—B—A]—[A—B—A—C—...] и т. д. (пример 15.1):

ФС — фрезерный станок; ТЦ — токарный центр; ПК — пункт контроля

ными установками. В примере 15.1 фрезерный станок заблокирован от $t = 52$ до $t = 68$ и затем от $t = 76$ до $t = 84$. Следовательно, операций фрезерования изделий A_3 и B_2 производятся с задержкой, несмотря на то, что обработка предыдущего изделия на этом станке закончена. Указанным периодам простоя на рис. 15.3 противопоставлен интервал времени от $t = 122$ до $t = 136$, когда фрезерный станок не заблокирован. Фрезерование изделия A_5 с успехом можно было бы начать при $t = 122$, но на рис. 15.3 так не сделано, чтобы показать начало «нового цикла». К тому же, начало обработки изделия A_5 раньше $t = 136$ все равно не привело бы к сокращению времени цикла, поскольку изделие B_3 так или иначе должно было бы ожидать момента $t = 144$.

В реальных производственных условиях, как правило, не существует такой свободы выбора последовательности обрабатываемых изделий, как в случае (б) примера 15.1. Но вся прелесть гибкой производственной системы заключается в том, что для нее стохастическое чередование изделий обычно даже лучше, чем организация их партиями, как в случае (а) того же примера. Поэтому, согласно Нейсбитту [49], с экономической точки зрения следует стремиться к созданию таких производственных мощностей, при которых гибкость не только возможна, но даже желательна.

тельна. И насколько же отличается это новое техническое мышление от философии традиционного массового производства, господствовавшей всего какое-то десятилетие назад.

15.2. ГРУППОВАЯ ТЕХНОЛОГИЯ

Для сохранения эффективности производства при обработке на одних и тех же станках однотипных, но не идентичных изделий был предложен метод, известный под названием *групповая технология* (ГТ). Он впервые был применен в Советском Союзе в 50-е гг. [44], а впоследствии обратил на себя внимание остального мира.

Схема групповой технологии. В групповой технологии акцент делается на сходство изделий, которым присваивается обобщающее понятие «семейство». Станки могут быть скомпонованы таким образом, чтобы производство этих семейств было наиболее экономичным. В итоге можно добиться существенного выигрыша в эффективности над традиционной схемой процесса, когда станки определенного типа сгруппированы в участки. Резонно задать вопрос, в чем же разница между групповой технологией и старым понятием *планирование выпуска изделия* — улучшенным вариантом планирования технологического процесса. Планирование выпуска изделия осуществляется в расчете на массовое производство какого-то одного образца или модели, что не характерно для современного рынка с его разнородными запросами. Групповая технология действительно тесно связана с планированием выпуска изделия; ее можно считать попыткой использовать все экономические преимущества схемы, ориентированной на единственное изделие, когда в действительности выпускаемые изделия не идентичны. В результате появляется схема производства, состоящая из нескольких поточных линий, которые могут иметь общие элементы.

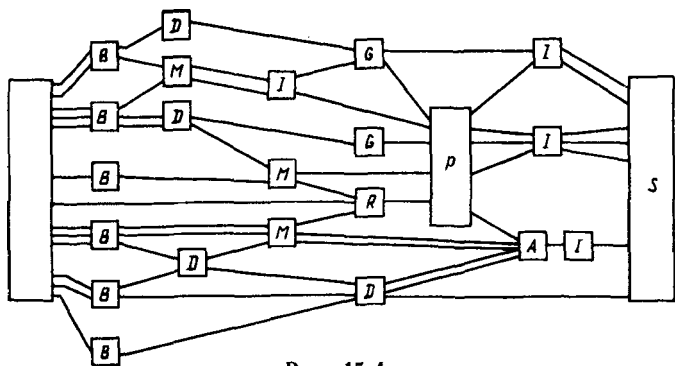


Рис. 15.4

Схема групповой технологии: 13 семейств изделий проходят 24 различные операции по их обработке:

ГР — гальваническое покрытие; S — упаковка и догрузка; A — сборка; B — зачистка; D — сверление; F — литейное производство; G — шлифование; M — фрезерование; I — контроль; R — роботизированная сварка

Семейство изделий	Последовательность операций	Число операций
1	$F-B-D-G-I-S$	6
2	$F-B-M-I-G-P-I-S$	8
3	$F-B-M-I-P-I-S$	7
4	$F-B-D-G-P-I-S$	7
5	$F-B-D-M-P-I-S$	7
6	$F-B-M-$	3
7	$F-$	3
8	$F-B-M-$	3
9	$F-B-M-$	3
10	$F-B-D-M-$	7
11	$F-B-D-D-$	4
12	$F-B-D-$	3
13	$F-B-D-S$	4
Всего		65

На рис. 15.4 изображена схема групповой технологии для тринадцати семейств изделий, проходящих 24 различные операции по их обработке. Эта схема не ориентирована ни на выпуск изделия, ни на организацию технологического процесса. Схема с планированием выпуска изделия должна была бы состоять из тринадцати независимых линий, не содержащих общих позиций, за исключением сварочных роботов и сборочных операций, на которых отдельные детали объединяются в узлы и агрегаты. Схема организации технологического процесса физически должна была бы состоять из станков, сгруппированных в участки, как, например, участок сверлильных станков или участок фрезеровальных станков. Единственными операциями, которые можно было бы назвать подобными участками на рис. 15.4, являются литейное производство, гальваническое покрытие и упаковка-погрузка.

Некоторые операции встречаются на пути нескольких семейств, другие же являются специализированными. Кроме того, из схемы следует, что только семь семейств изделий достигают операции упаковки и погрузки, тогда как из литейного производства выходят тринадцать семейств. Это объясняется тем, что на пунктах сварки и сборки происходит объединение деталей в узлы и агрегаты. Число станков здесь значительно меньше, чем в соответствующей схеме, ориентированной на выпуск изделий, благодаря комбинированию рабочих мест для изделий со схожими характеристиками. Это видно из табл. 15.2, где приведено количество пунктов обработки (включая литье, гальваническое покрытие и упаковку-погрузку), которое имела бы схема произ-

водства, ориентированная на выпуск изделий. В табл. 15.2 подсчитано число пунктов (операций) для всех семейств, начиная с самого верхнего пути на рис. 15.4. Ясно, что их число (65) значительно больше, чем 24, как это имеет место в схеме групповой технологии на рис. 15.4. Реально от групповой технологии вряд ли можно было бы ожидать такой экономии, поскольку операции литья, гальванического покрытия и упаковки-погрузки скорее всего не сгруппированы так, как в данной схеме. Тем не менее на 19 позициях можно было бы сэкономить за счет группировки семейств изделий, проходящих через другие операции.

Распознавание изделий. Еще одним аспектом групповой технологии является классификация изделий по признакам, что делается обычно с помощью нумерации. В процессе конструирования, особенно если оно осуществляется с применением ЭВМ, каждому разряду номера изделия полезно придать определенный логический смысл. Тогда конструктор может извлечь чертеж изделия из базы данных ЭВМ, сославшись на номер этого изделия, даже если он не знает его устройства и назначения. Но, задавая логический номер изделия, конструктор должен быть уверен, что извлекаемый чертеж по крайней мере похож на то, что ему нужно. В результате этого чертеж может быть скопирован в том виде, как он есть, или же с небольшими изменениями.

Расплатой за присвоение логического смысла каждому разряду номера изделия являются очень большие номера. Грейвз [24] отмечает, что в некоторых системах групповой технологии для нумерации изделий используются 36-разрядные числа. В таких системах изделие практически полностью описывается всего одним числом, но число это столь велико, что его невозможно запомнить; оно имеет реальную пользу только в случае хранения в памяти ЭВМ.

Разрешающая способность системы нумерации изделий зависит от числа разрядов и количества возможных значений каждого разряда. Так, 36-разрядная система нумерации изделий обладает следующей разрешающей способностью:

1. Разряды только цифровые: 10^{36} .
2. Разряды алфавитно-цифровые: 36^{36} .

36-разрядная система имеет огромную степень разрешения и позволяет почти полностью описать конструкцию изделия одним числом. Но 36-разрядные числа очень громоздки и неудобны в практическом применении.

В групповой технологии используются две системы цифровой идентификации: иерархическая и цепного типа. В иерархических системах интерпретация последующих разрядов зависит от выбора предыдущих, старших, разрядов. В системах цепного типа смысл каждого разряда жестко определен. Рассмотрим, например, систему классификации промышленных роботов. Первый разряд может интерпретировать осевую конфигурацию, второй — источник питания, третий — нагрузочную способность, а четвертый

и последующие разряды — прочие характеристики робота. В иерархической системе гидравлический робот грузоподъемностью 10 кг может быть представлен кодом

311XXX ... X.

Третий разряд здесь равен единице, потому что грузоподъемность 10 кг может, скажем, соответствовать нижней границе диапазона при кодировании гидравлических роботов. Однако в системе цепного типа данный робот может быть представлен кодом

317XXX.X.

В системе цепного типа интерпретация третьего разряда жестко определена и не зависит от выбора первого и второго разрядов. Следовательно, разряд «нагрузка» должен принимать определенное значение для любой конфигурации робота и для любого источника питания — от самых маленьких настольных электрических моделей до огромных гидравлических установок. Поэтому в системе цепного типа цифра 7 выбрана для обозначения грузоподъемности 10. кг, что соответствует примерно середине диапазона грузоподъемности роботов. Из этого примера ясно, что иерархическая система имеет преимущество в точности описания предметов. Но система цепного типа гораздо проще в интерпретации благодаря однозначному декодированию разрядов. Даже при использовании ЭВМ кодирование и декодирование номеров изделий в иерархической системе может представлять собой очень сложную задачу.

Групповую технологию можно сравнить с известной игрой в двадцать вопросов. Смысл игры в том, чтобы отгадать предмет, задав 20 вопросов, ответы на которые могут быть только «да» или «нет»; при этом заранее оговаривается, что заданный предмет является растением, животным или минералом. Это эквивалентно описанию предмета 21-разрядным кодом, в котором первый разряд является троичным (растение, животное или минерал), а остальные 20 — двоичными (да или нет). Система является иерархической, так как отгадывающий может произвольно формулировать каждый последующий вопрос в зависимости от ответов на предыдущие. Разрешающая способность такой системы равна 3×2^{20} . Такое разрешение явно проигрывает в сравнении с 30-разрядной классификацией, принятой в групповой технологии; и все же любой, кто играл в двадцать вопросов, мог убедиться, насколько глубокое впечатление оставляет разрешающая способность этой простой игры.

Экономическая сторона групповой технологии. Групповая технология выгодна не только благодаря разумной организации производства, упорядочению данных и маркировке изделий, она снижает затраты на оборудование. В этом смысле одним из основных выигрышных моментов является уменьшение количества

инструментов и приспособлений, необходимых в производственном процессе. Групповая технология привлекательна тем, что два изделия, совершенно отличные по остальным показателям, могут быть смонтированы с помощью одних и тех же приспособлений, если коды их геометрических параметров идентичны.

Хотя основные идеи групповой технологии находят применение уже не одно десятилетие, ее потенциальные возможности еще не исчерпаны, и здесь есть над чем подумать. Постоянно открываются новые направления, где можно получить экономическую выгоду, и задача исследований в области групповой технологии — получить количественную оценку этой выгоды.

15.3. ПРОИЗВОДСТВО С КОМПЛЕКСНЫМ УПРАВЛЕНИЕМ ОТ ЭВМ

С ГПС и групповой технологией тесно связано еще одно понятие: производство с комплексным управлением от ЭВМ. Это понятие заменяет собой более известное словосочетание «комплексная система автоматизированного проектирования и изготовления с использованием ЭВМ» (английская аббревиатура CAD/CAM — *прим. пер.*). Этот новый термин подчеркивает тесную взаимосвязь между проектированием, производством и даже контролем изготовления и качества и другими сторонами управления производственной сферой. Поэтому понятие ПКУ является более общим, нежели ГПС, и его можно распространить на некоторые производственные системы, которые не являются полностью автоматизированными.

В ПКУ акцент делается на связь, осуществляемую ЭВМ между различными подсистемами программного обеспечения, управляющими процессом производства и вспомогательными операциями. Это не такая простая задача, ибо подобных подсистем всегда столько же, сколько и обслуживаемых ими областей производства. Так, в области технического проектирования имеется одно программное обеспечение (САПР), в области ЧПУ — другое, в области материально-технического снабжения — третье, в области планирования технологических процессов — четвертое и т. д.

Естественно, что проектирование и изготовление явились первыми областями, которые было признано необходимым связать каналом обмена данных. И в той и в другой требуется скрупулезное кодирование для полного описания изделий. Объединение процессов проектирования и изготовления привело к значительному выигрышу, так как позволило для кодирования изделий использовать общую базу данных.

Достижения в области комплексного проектирования и изготовления стимулировали компьютеризацию таких областей, как планирование технологических процессов, контроль производства, транспортирование материалов, складирование, контроль качества. В начале 80-х гг. было предпринято немало усилий по

стандартизации баз данных различных областей с целью их совместимости. Отсутствие такой совместимости в 70-е гг. даже на одном и том же предприятии часто подрывало доверие к ЭВМ со стороны администрации, хотя все вычисления осуществлялись в пакетном режиме вне контура управления производством. Преодоление подобной несовместимости является необходимым условием достижения успеха в сфере производства с комплексным управлением от ЭВМ. В свою очередь, и комплексное управление способствует развитию стандартизации в направлении устранения несовместимости баз данных.

Проблему объединения баз данных различных областей производства ни в коем случае не следует преуменьшать. В 1984 г. Садовский [62] высказал такую мысль: «Хотя нам известны многочисленные примеры систем управления производством и материально-техническим снабжением с объединенными базами данных, очень немногие из них удачно сопрягаются с системами, автоматически вырабатывающими списки запрашиваемых материалов; и еще меньше таких, которые включают в себя системы автоматизации нисходящего планирования технологических процессов и административных функций. Эти и многие другие проблемы должны быть решены, прежде чем появятся комплексно автоматизированные производства в их истинном понимании».

Производство с комплексным управлением от ЭВМ является целью, имеющей большие последствия; немногие предприятия достигли этой цели, а многие, возможно, никогда не достигнут ее. Но в выборе роботов, станков с ЧПУ, промышленных логических систем управления, технических средств и программного обеспечения систем нет ничего противоречащего генеральной линии на создание полностью интегрированных систем комплексной автоматизации. Создавая системы с комплексным управлением от ЭВМ, мы приближаемся к высшей степени автоматизации производства. Некоторые специалисты по робототехнике заглядывают еще дальше, и речь об этом пойдет в следующем разделе.

15.4. САМОВОСПРОИЗВОДЯЩИЕСЯ СИСТЕМЫ

В представлении большинства людей высшей степенью автоматизации является завод, целиком управляемый роботами без какого-либо участия человека. Но это не так. Бутов и его коллеги [11] рассматривают возможность создания «самовоспроизводящихся систем», в которых часть производственной мощности станков используется не для выпуска продукции, а для изготовления аналогичных станков. Эта идея проиллюстрирована на рис. 15.5.

Чтобы роботы производили роботов, как люди рожают людей, — это, конечно, из области научной фантастики. То же самое можно сказать про заводы, рожающие другие заводы, хотя это, возможно, не так странно, как в случае с роботами. Но между

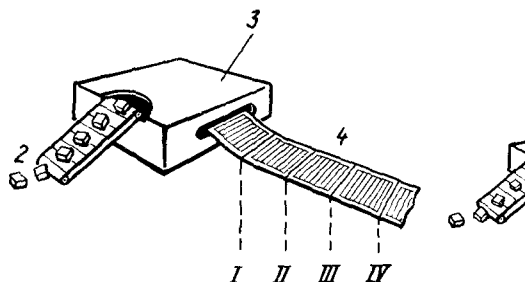
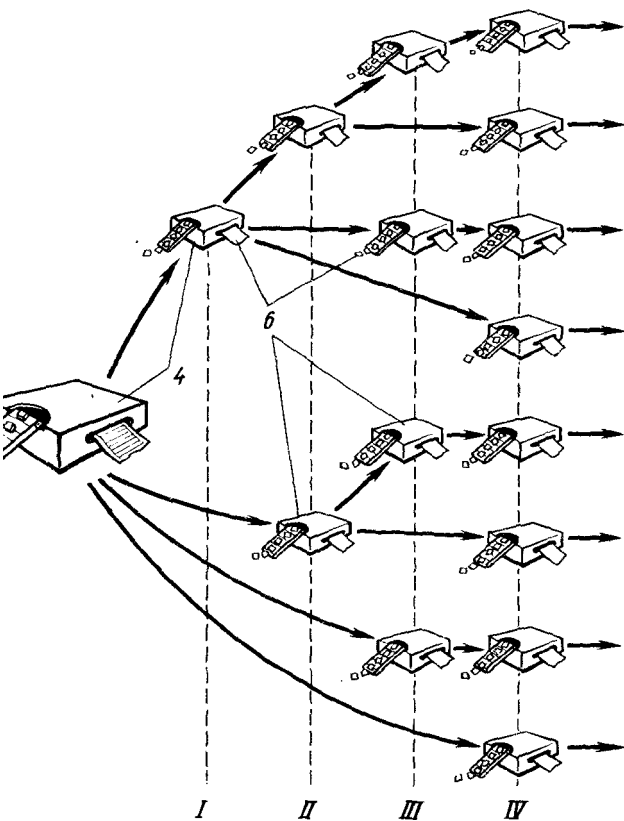


Рис. 15.5

Саморазмножающиеся системы часть времени и энергии используют для сборки дополнительного количества станков:

слева — пропорциональная система; справа — саморазмножающаяся система; 2 — сырье; 3 — технологический процесс; 4 — продукция; 6 — копии технологического процесса; I—IV — годы



фантастикой и реальностью есть различие, и заключается оно в том, что в саморазмножающейся системе действительно станки можно частично использовать для производства других станков того же типа. На рис. 15.5 показано, что каждая установка создает одну целую подобную себе установку. Но в ряде случаев вполне возможно и целесообразно и частичное воспроизводство.

Исследуем эффект размножения в системе станков. Обозначим P_k — выпуск продукции за период k ; N_k — число станков, созданных за период k ($k = 1, 2, \dots$); N_0 — число станков в начале производства; p — часть станка, предназначенная для выпуска продукции; r — часть станка, предназначенная для воспроизводства себе подобных ($p + r = 1$); K_p — производительность одного станка при выпуске продукции; K_R — производительность одного станка при воспроизводстве себе подобных.

В табл. 15.3 приведены выражения, характеризующие число станков и общее количество продукции, накопленное к концу каждого последующего периода. И то и другое пропорционально зависит от начального числа станков N_0 . Выбор коэффициентов p и r зависит от принятой стратегии: большие значения p гарантируют быструю отдачу, но для длительного периода времени выгоднее задавать большие значения r , так как в этом случае получается больше продукции и больше станков. Результаты подытожены следующими формулами:

общее количество станков N

$$N = \sum_{j=0}^n N_j = (rK_R + 1)^n N_0, \quad (15.1)$$

общее количество продукции P

$$P = \sum_{j=0}^n P_j = pK_p N_0 \left[\frac{(1 + rK_R)^n - 1}{rK_R} \right]. \quad (15.2)$$

Пример 15.2. Саморазмножающаяся система роботов. Предположим, что робот был бы способен с минимальной помощью за один год либо создавать себе подобного, если бы он занимался только этим, либо строить дом для одного человека, если бы все время посвящал строительству. Сколько времени потребовалось бы этому роботу и его потомкам, чтобы построить дома для 100 человек, если бы 75 % времени роботы занимались строительством, а остальные 25 % — воспроизводством себе подобных? Сколько времени понадобилось бы для решения той же задачи при распределении деятельности роботов в соотношении 50 — 50? 25 — 75? Предполагается, что роботы, в отличие от людей, не подвержены усталости и старению при выполнении ими любой работы.

Решение. $K_p = K_R = 1$, $N_0 = 1$. Для $p = 0,75$ и $r = 0,25$ по (15.2) для $P = 100$ получим

$$100 = 0,75 \left[\frac{(1,25)^n - 1}{0,25} \right];$$

$$n = 15,85 \text{ лет.}$$

k	N_k	$\sum_{j=0}^k N_j$	P_k	$\sum_{j=0}^k P_j$
0	N_0	N_0	0	0
1	$rK_R N_0$	$rK_R N_0 + N_0 =$ $= (rK_R + 1) N_0$	$pK_p N_0$	$0 + pK_p N_0 = pK_p N_0$
2	$rK_R (N_1 + N_0)$	$rK_R (rK_R + 1) N_0 +$ $+ (rK_R + 1) N_0 =$ $= (rK_R + 1)^2 N_0$	$pK_p (N_1 + N_0)$	$pK_p (rK_R + 1) N_0 + pK_p N_0 =$ $= pK_p N_0 (rK_R + 1 + 1) = pK_p N_0 (rK_R + 2)$
3	$rK_R (N_2 + N_1 +$ $+ N_0)$	$rK_R (rK_R + 1)^2 N_0 +$ $+ (rK_R + 1)^2 N_0 =$ $= (rK_R + 1)^3 N_0$	$pK_p (N_2 + N_1 +$ $+ N_0)$	$pK_p (rK_R + 1)^2 N_0 + pK_p N_0 (rK_R + 2) =$ $= pK_p N_0 [(rK_R + 1)^2 + (rK_R + 2)] =$ $= pK_p N_0 (r^2 K_R^2 + 3rK_R + 3) =$ $= pK_p N_0 \left[\frac{(r^3 K_R^3 + 3r^2 K_R^2 + 3rK_R + 1) - 1}{rK_R} \right] =$ $= pK_p N_0 \left[\frac{(rK_R + 1)^3 - 1}{rK_R} \right]$
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
n	$rK_R \sum_{j=0}^n N_j$	$(rK_R + 1)^n N_0$	$pK_p \sum_{j=0}^n N_j$	$pK_p N_0 \left[\frac{(1 + rK_R)^n - 1}{rK_R} \right]$

Для $p = 0,50$ и $r = 0,50$:

$$100 = 0,5 \left[\frac{1,5^n - 1}{0,5} \right] = 1,5^n - 1;$$

$$n = 11,38 \text{ лет.}$$

Для $p = 0,25$ и $r = 0,75$:

$$100 = 0,25 \left[\frac{1,75^n - 1}{0,75} \right];$$

$$n = (\ln 301)/(\ln 1,75) = 10,20 \text{ лет.}$$

Хотя пример 15.2 может показаться неестественным, давайте, все же, согласимся с тем фактом, что уже сейчас роботы фактически используются отчасти и для производства других роботов. Коэффициенты r и K_R очень малы, и дочерние роботы идут *на продажу*, а не остаются на заводе с целью наращивания производства. Но этот пример заставляет нас задуматься о том, куда же мы движемся, и поднимает вопросы этического свойства. Этические проблемы автоматизации будут рассмотрены в этой главе чуть позже.

В перспективе интересной сферой приложения саморазмножающихся установок, особенно роботов, являются космические станции. Космическое пространство является средой, опасной для человека, и использование роботов в космосе даже более многообещающе, чем на земле. Сложности возникнут только с посещением станции и доставкой на нее роботов. Разумеется, это не полностью решает вопрос, потому что в космосе всегда существует проблема сырья. Тем не менее, саморазмножающиеся системы представляют собой интересную экспериментальную модель производства как на земле, так и в космическом пространстве.

15.5. ДАТЧИКИ ТИПА КОШАЧЬИХ УСОВ

Техническое осязание является интригующей областью исследований в роботостроении, и один из оригинальных методов осязания заключается в установке на захватном устройстве робота гибких щупов, имитирующих кошачьи усы. Этот метод был пред-

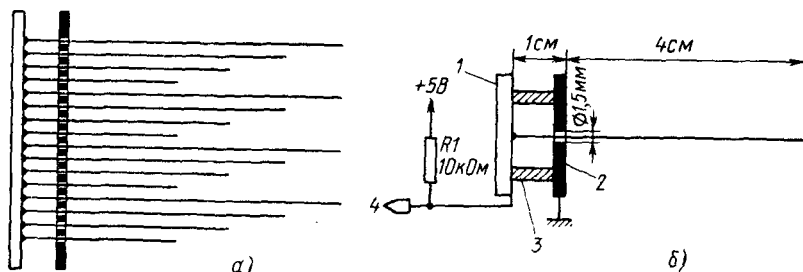
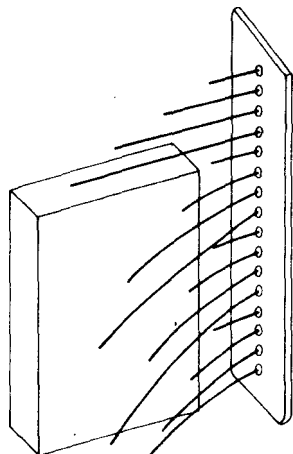


Рис. 15.6

Датчик типа кошачьих усов для устройства технического осязания робота:
 а — датчик в виде ряда щупов; б — датчик в виде одного щупа; 1 — фольгированная пластинка; 2 — медная пластинка; 3 — пластиковая трубка; 4 — выходной сигнал (1 — есть касание, 0 — нет касания)

Рис. 15.7

Датчик типа кошачьих усов, блок прямоугольной формы и соответствующая ему двоичная матрица



0	0	0	0
1	1	1	0
1	1	1	0
1	1	1	0

ложен Р. Эндрю Расселом из Уоллонгонгского университета в Австралии [61]. На рис. 15.6 схематически изображены один такой щуп и целый ряд щупов, напоминающих кошачьи усы. Отклонение щупа в любом направлении приводит к его электрическому контакту с краем отверстия диаметром 1,5 мм в заземленной медной пластине; в результате такого контакта формируется выходной сигнал (двоичная единица), указывающий на то, что щуп чего-то коснулся.

Составной датчик типа кошачьих усов, размеры которого не позволяли установить его на руке робота в опыте Рассела, был использован в качестве неподвижного устройства для определения формы проходящих мимо него предметов. Для формирования матрицы из шестнадцати двоичных разрядов в четырех уровнях были расположены группы из четырех отличающихся по длине щупов.

На рис. 15.7 изображено применение составного датчика для обнаружения блока прямоугольной формы. Блоки иной формы могут быть представлены следующими двоичными матрицами:

Восьмиугольник					Треугольник				
1	0	0	0	0	0	0	0	0	
1	1	0	0	0	1	0	0	0	
1	1	0	0	0	1	1	0	0	
1	0	0	0	0	1	1	1	0	

При более сложной форме предметов или при наличии у них длинных выступающих частей будут возникать определенные трудности. Кроме того, если щуп отклонился так, что он коснулся другого щупа, то оба они зарегистрируют контакт с предметом.

Одиночный щуп, закрепленный на руке робота, может оказаться очень полезным при выполнении операции, изображенной на рис. 15.8. Точное положение цилиндрического объекта роботу не известно, он определяет его по отклонению щупа. Однако роботу известно положение объекта относительно оси y . Следовательно, захват робота движется вдоль оси x до заданного положения. После этого робот может с помощью щупа определить координату y объекта, установить захват на соответствующем расстоянии над объектом, открыть его и подобрать объект. Робот Рассела [61] обладал способностью найти на столе два цилиндра и поставить один из них на другой.

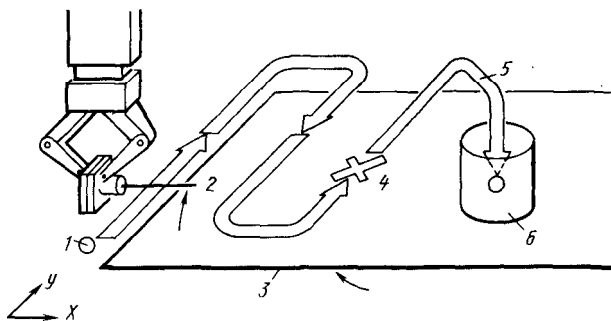


Рис. 15.8

Рука робота с датчиком в виде одиночного щупа использует координатный метод поиска объекта на столе, определяет его точное положение, а затем подбирает: 1 — начало поиска; 2 — щуп; 3 — область поиска; 4 — объект обнаружен; 5 — захват над объектом; 6 — захват сжимается и подбирает объект

К другим интересным разработкам в области робототехники относятся постоянно усложняющиеся системы технического зрения, подвижные (шагающие) роботы и искусственный интеллект. Поскольку промышленные роботы становятся все более и более искусными и похожими на людей, приобретают особую актуальность этические проблемы автоматизации.

15.6. ЭТИЧЕСКИЕ ПРОБЛЕМЫ АВТОМАТИЗАЦИИ

Рано или поздно каждый из нас должен начать задавать вопросы этического свойства:

Существует ли предел для производства, которое мы можем поручить роботам и автоматам?

Освободят ли промышленные роботы людей от работы?

Если роботы начнут создавать себе подобных, найдется ли в конце концов для них столько работы?

Не приведет ли автоматизация в конечном счете к возможности уничтожения ею тех, кто ее усиленно развивает?

Не получится ли когда-нибудь так, что люди будут служить машинам, а не наоборот, как было задумано?

Безработица. Основной вопрос этического свойства заключается в том, не приводит ли автоматизация к безработице. В представлении общественности автоматизация обычно выступает в роли злодея, олицетворяющего безработицу. Однако на этот счет существуют Теория X и Теория Y, столь же противоположные по смыслу, как и соответствующие оригинальные теории Дугласа — Макгрегора в приложении к управлению производством [29]. В отношении *автоматизации* Теория X утверждает, что она *вызывает* безработицу, а Теория Y говорит о том, что она *предотвращает* ее.

Теория X. Совершенно очевидно, что принципиальное назначение роботов и автоматизации производства состоит в том, чтобы

сделать технологические процессы более эффективными. Роботы обычно экономят трудовые затраты, а это значит, что они выполняют все то, что раньше производилось рабочими. Даже в тех случаях, когда какой-то новый процесс идет без участия людей, внедрение робота означает, что один или более рабочих не будут заняты на этом производстве. Причем замена здесь производится не в отношении один к одному. Роботы могут выполнять операции со скоростью, равной или даже большей, чем человек-оператор, они могут работать по три смены в день, по семь дней в неделю, что эквивалентно труду четырех рабочих. Даже с учетом времени на текущий ремонт и профилактику робот часто заменяет более чем одного рабочего.

Противники Теории X считают роботы работодателями, а не работоустранителями, заявляя, что рабочие необходимы для создания и обслуживания роботов. Хотя уровень мастерства и типы работ могут претерпевать изменения, сторонники роботов доказывают, что число операций остается прежним или даже увеличивается. Но этот аргумент противоречит принципиальному назначению роботов и автоматизации: повышать эффективность производства. Приверженцы Теории X говорят: если верно, что для создания и обслуживания роботов требуется столько же или даже больше людей, сколько раньше было необходимо для выполнения той же работы, которую теперь делают роботы, то где же здесь повышение эффективности производства?

Конечно, кроме повышения эффективности производства имеется много других причин для применения роботов и автоматизации. Среди них такие, как улучшение качества, повышение безопасности, устранение монотонных, утомительных или малопривлекательных работ, но стоит ли ради этого оставлять рабочих без средств к существованию? Современный инженер должен считаться с этими аргументами, будучи подверженным либо угрызениям собственной совести, либо нападкам противников автоматизации. Столкнувшись с такой проблемой, он должен задать себе вопрос: как объяснить рабочему, которого собираются заменить роботом, что его работа «нежелательна» и от нее надо будет отказаться.

Итак, противники технического прогресса относятся к роботам и автоматизации, как к средствам уничтожения рабочих мест, и доводы, что взамен старых появляются новые рабочие места или что необходимо исключить нежелательные ручные операции, не изменяют их убежденности в том, что автоматизация порождает безработицу.

Теория Y. Но существует и другой взгляд на автоматизацию, как на средство, *увеличивающее* или, по крайней мере, *сохраняющее* занятость. Однако возможно ли в условиях общемировой конкуренции бесконечно поддерживать занятость, если ту же самую работу более эффективно делают машины? Без автоматизации конкуренция может привести к исчезновению целых отраслей

промышленности и всех рабочих мест в этих отраслях. В примере 15.3 все внимание уделяется именно этому вопросу.

Пример 15.3. Автоматизация в сталелитейной промышленности. В 1951 г. корпорация «Юнайтед Стэйтс Стил» насчитывала 301 328 рабочих. В 1983 г. число рабочих уменьшилось почти на 67 % и составляло всего 98 722. За этот же период количество произведенной и отгруженной продукции снизилось с 24,6 млн. т до 11,0 млн. т [74]. Если не принимать во внимание другую продукцию, выпускаемую корпорацией, то каково было производство стали в расчете на одного рабочего P , т/чел, в 1951 г. и в 1983 г.?

Решение:

$$1951 \text{ г.: } \frac{24,6 \cdot 10^6}{301\,328} = 81\,638;$$

$$1983 \text{ г.: } \frac{11,0 \cdot 10^6}{98\,722} = 111\,424.$$

Эти результаты довольно приближенны, поскольку корпорация «Ю. С. Стил» в 1983 г. была более многопрофильной, чем в 1951 г. Поэтому рост производительности при выпуске стали был бы более значительным, если бы было известно число рабочих, занятых выпуском другой продукции, и это число учитывалось при вычислениях. Такой рост производительности за последние три десятилетия позволил предотвратить крах «Ю. С. Стил» и многих других компаний в условиях сильной иностранной конкуренции. Экономия за счет роста производительности превысила общую прибыль, и если бы не эта экономия, то компании давно бы уже не существовали.

Если посмотреть шире и провести аналогичные расчеты по сталелитейной промышленности в мировом масштабе, то мы, бесспорно, обнаружим, что автоматизация уменьшает количество рабочих, требуемых для поддержания выпуска стали на заданном уровне. Но раз здесь замешана автоматизация, то любая компания, не применяющая ее, почти наверняка обречена на банкротство и, следовательно, на увольнение всех рабочих.

Хотя в качестве примера была выбрана сталелитейная промышленность, нетрудно обнаружить аналогию и в других отраслях. Хорошим примером является текстильная промышленность. Без автоматизации текстильная промышленность в таких странах, как США, имела бы мало шансов на выживание в сравнении с другими странами, где применяется дешевый ручной труд.

Большинство людей рассматривают профсоюзы в качестве противников автоматизации на том основании, что она приводит к сокращению рабочих мест. Но еще в середине 60-х гг. стало очевидным, что в экстремальных ситуациях профсоюзы поддерживают автоматизацию. В 1966 г. в статье Макмануса [35], помещенной в сборнике «Железный век», было написано: «21 марта 1966 г. профсоюз потребовал от компании АЛКОА (Aluminum Company of America) автоматизировать завод в Нью-Кенсингтоне, шт. Пенсильвания. АЛКОА приостановила программу с финансированием в 4,5 млн. долл. и угрожала закрыть завод. Руководство профсоюза нашло способ, как предотвратить закрытие пред-

приятия. Оно согласилось продолжить работу при условии перерасчета трудового стажа, перевода на другие операции и повсеместного повышения эффективности производства... Короче говоря, профсоюзы настаивают на автоматизации, а не борются с ней.»

В 80-е гг. привычными стали такие явления, как борьба за существование в условиях напряженной конкуренции. И похоже, что автоматизация является тем оружием, которое обеспечивает сохранение занятости и выживаемость какой-либо компании в соперничестве с другими фирмами.

Приведенный выше аргумент служит намеком на то, что автоматизация — это неизбежное зло, потому что и другие применяют ее. Это наталкивает на мысль: а не договориться ли об ограничении автоматизации в мировом масштабе подобно тому, как сверхдержавы ведут переговоры о сокращении вооружений. Подобное предложение вряд ли осуществимо, но даже если бы оно было выдвинуто, то, по-видимому, не нашло бы поддержки, потому что вывод в пользу автоматизации как средства обеспечения занятости выглядит сильнее, чем представление о ней как только об оружии в борьбе с конкурентами. Высвобожденные за счет автоматизации производственные мощности могут быть использованы либо для выпуска других изделий, либо для увеличения производства тех же самых изделий. Предположим, например, что бум автоматизации позволил бы вдруг снизить стоимость автомобилей наполовину; большинство семей решило бы, что они могут позволить себе приобрести еще один автомобиль. Или, если им достаточно одного автомобиля, они могли бы вложить средства в покупку более комфортабельного жилья или других товаров, а это способствовало бы увеличению числа рабочих мест в соответствующих отраслях производства.

В начале нашего столетия автомобиль был таким же предметом роскоши, каким сейчас является персональный самолет. Но сборочные линии Генри Форда и автоматизация на манер Детройта сделали автомобили доступными для рядового потребителя и привели к огромному спросу на них в 20-е гг. В связи с этим возникли миллионы рабочих мест в автомобильной промышленности и смежных отраслях. Сегодняшняя волна автоматизации увеличивает объем продажи других товаров и, как следствие, приводит к росту числа рабочих мест подобно тому, как это было в автомобильной промышленности в 20-е гг.

Что касается автоматизации, то ни Теория X, ни Теория Y не дают полного ответа на поставленный вопрос. Да, автоматизация приводит к сокращению рабочих мест, но из этого еще не следует, что она создает безработицу. В действительности логика свидетельствует об обратном.

Автоматизация без прикрас. Наибольший вред приносит преувеличение возможностей автоматизации. Очень часто инженеры в порыве энтузиазма преувеличивают эффективность разрабаты-

ваемых ими проектов роботов и систем автоматизации производства. В результате оказываются выброшенными на ветер средства, вложенные в создание оборудования, которое на деле не способно дать экономический эффект, оправдывающий исходные затраты. Печальнее всего то, что при этом приносятся в жертву рабочие места, на которых ранее были заняты люди. Последствия очевидны: сокращение рабочих мест без реального выигрыша в эффективности производства и снижения себестоимости продукции. В этой ситуации все оказываются в проигрыше, и при разработке последующих проектов доверие к автоматизации может быть сильно подорвано.

Закон Мерфи. «Если какая-нибудь неприятность может случиться, она случается». Почти каждый знаком с этим законом Мерфи, но вот происхождение его неизвестно. Все же есть достаточно оснований полагать, что Мерфи был инженером по автоматике. Проекты автоматизации как нельзя лучше подпадают под действие закона Мерфи по нескольким причинам. Прежде всего это постоянное и сильное стремление, а возможно и *требование*, найти какой-то способ повышения производительности и снижения стоимости. Успех в этом деле сулит огромные прибыли, а за ошибки приходится жестоко расплачиваться. Следовательно, будут перепробованы многие идеи, и неизбежно некоторые из них окажутся забракованными. Во-вторых, занятие автоматизацией почти всегда сродни погружению в неизвестность. Когда испытываются новые процессы и оборудование, всегда возникают какие-то неожиданности. С подобными примерами мы встречались в гл. 8, когда рассматривали процесс создания робота.

Вероятность появления неожиданных ситуаций связана с проблемами этического характера. При разработке и обосновании нового проекта роботизации обязательно надо допускать возможность возникновения таких ситуаций, чтобы избежать поспешного утверждения проекта, который впоследствии не даст ожидаемых результатов. Кроме того, надо считаться с возможными отказами оборудования. Безопасность всегда является предметом особого внимания на стадии предварительных испытаний и отладки. Но она может также стать серьезной проблемой и на этапе эксплуатации новой автоматизированной системы, поскольку рабочие могут быть не подготовлены к аварийным режимам и потенциальным отказам.

Отказы оборудования с точки зрения права. Общество не останется безучастным в случаях, когда из-за отказов оборудования кто-то получает повреждения или увечья. Представление о промышленном роботе, наносящем повреждение или даже убивающем рабочего, просто не укладывается в сознании, настолько редко это случается. Тем не менее, отмечены фатальные исходы, имевшие место как в США, так и в Японии. В одном случае в США суд присяжных постановил выплатить семье потерпевшего компенсацию в размере 10 млн. долл. [28]. Этот рабочий взбирался

на стеллаж склада, который обслуживался механической рукой. Адвокат семьи погибшего сказал: «Я полагаю, следует задать вопрос: «Кто же кого обслуживает?» Мы обязаны сделать все возможное, чтобы не скатиться назад к мнению, бытовавшему в эпоху промышленной революции, что люди являются обычным расходуемым материалом».

Если закон Мерфи является недостаточным стимулом соблюдения осторожности при разработке проектов автоматизации, то достаточными должны быть социальные последствия и угроза судебного иска в 10 млн. долл.

В ы в о д ы

В этой главе показано, к чему приводят внедрение роботов и автоматизация производства. Кроме того, рассмотрен небольшой пример, иллюстрирующий порождаемые этим процессом направления научных исследований и этические проблемы, связанные с автоматизацией. Гибкая автоматизация и различные типы описанных в этой книге автоматизированных систем привели к изменению образа жизни и запросов потребителей. Эти тенденции подмечены Джоном Нейсбитом [49]. Потребитель больше не скован выбором товаров из ограниченного числа моделей или типов. Новая свобода выбора, ставшая возможной благодаря гибкому производству, подталкивает его к еще большей гибкости и приводит к созданию ГПС. Однако, как подчеркнуто в предыдущих главах, ГПС невозможна без автоматизации отдельных рабочих мест.

Естественным продуктом гибкой автоматизации и гибкого производства является групповая технология. Она дает экономический эффект как на этапе конструирования, так и в стадии производства.

В ПКУ акцент делается на связь между программным обеспечением собственно производства и общей базой данных. Хотя ПКУ на данный момент является в большей степени идеей, чем реальностью, оно представляет собой тот идеал, к которому должны стремиться инженеры, занимающиеся разработкой и внедрением роботов и подсистем автоматизации производства.

При изучении саморазмножающихся систем подчеркнута особенность роботизации, основанная на самой замечательной способности человека — способности к воспроизводству. Представление о роботах, порождающих другие роботы, выглядит неестественно, но уже сегодня некоторые роботы используются для этой цели. Даже системы с частичным воспроизводством можно считать саморазмножающимися, имея в виду использование машин и станков для изготовления дополнительного количества таких же установок.

Еще одной областью исследований является техническое осязание. В качестве примера в этой главе рассмотрен эксперимент

по обнаружению объектов с помощью датчика с гибкими щупами, изготовленного по образцу и подобию кошачьих усов. Другими быстро развивающимися областями исследований являются техническое зрение, технический слух и искусственный интеллект.

Проследивая развитие робототехники и автоматизации производства, нельзя избежать их оценки с этической точки зрения. Принципиальный вопрос этического свойства здесь таков: приводит ли автоматизация к безработице или нет? На этот счет есть две теории — Теория X и Теория Y. Статистика говорит в пользу Теории Y, утверждающей, что автоматизация сохраняет рабочие места, которые могли бы быть утрачены в результате всеобщей конкуренции.

Легко впасть в чрезмерный оптимизм по поводу успехов робототехники или других направлений автоматизации, если не считаться с законом Мерфи, безопасностью, простоями оборудования и другими потенциальными негативными последствиями. В результате многих проектов, выполняемых с большим оптимизмом, появляются доставляющие массу хлопот дорогие, но неэффективные, опасные в эксплуатации или ненадежные системы. Это приводит к увольнению части рабочих, однако производственный процесс не улучшается, и в проигрыше оказываются все.

Роботы и автоматизация производства вырабатывают определенное отношение к ним со стороны общественности. Насколько велико здесь проявление эмоций, можно судить по большому числу судебных решений, вынесенных по поводу несчастных случаев с участием роботов.

Огромные возможности роботов и автоматизации производства таят в себе соблазн в погоне за сенсациями игнорировать этические проблемы. Роботы сейчас находятся в центре внимания, но то, как их примет общественность, в значительной степени зависит от этических взглядов инженеров, занимающихся их разработкой и внедрением.

Упражнения и вопросы

15.1. В групповой технологии классификация изделий производится с помощью номеров, состоящих из шести цифровых и четырех алфавитных разрядов. Какова разрешающая способность такой системы нумерации изделий?

15.2. В иерархической системе идентификации изделий использовано шесть цифровых разрядов. Сколько признаков изделий можно закодировать таким образом? Сколько потребовалось бы признаков изделий, если бы вместо иерархической системы применялась система с фиксированными разрядами?

15.3. Какова разрешающая способность системы идентификации изделий, в которой использовано восемь цифровых разрядов? Какова была бы разрешающая способность такой системы, если бы разряды были не цифровыми, а алфавитными? Какова была бы разрешающая способность в случае алфавитно-цифровых разрядов?

15.4. Саморазмножающаяся система имеет производительность при собственном воспроизводстве (K_D) 20 %, а при выпуске продукции (K_P) — 50 единиц. Производственные мощности системы распределены так: для воспроизводства (r) используется 10 %, а для выпуска продукции (p) — 90 %. Если в начале

эксплуатации имеется всего одна установка, то сколько продукции сможет дать система за пять лет? Сколько продукции она даст за тот же период, если значения коэффициентов r и p поменять местами?

15.5. Космическая станция оснащена саморазмножающимся оборудованием в виде роботов, производительность которых при собственном воспроизводстве (K_D) и при выпуске продукции (K_P) составляет соответственно 10 % и 5000 единиц. Если соотношение r/p равно 50/50 и в начале эксплуатации на станции имеется пять роботов, то сколько роботов будет там через 10 лет? Каков будет годовой объем производства системы на десятом году эксплуатации? Сколько всего единиц продукции будет выпущено к началу десятого года эксплуатации, а также к его концу?

15.6. Что представляют собой Теория X и Теория Y применительно к автоматизации?

15.7. Вызывает ли автоматизация безработицу или, наоборот, препятствует ее возникновению? Аргументируйте Ваш ответ.

15.8. Укажите, какие виды деятельности инженера, занимающегося автоматизацией, можно отнести к неэтическим. Каковы последствия этих действий?

15.9. Тактильный датчик типа кошачьих усов, изображенный на рис. 15.6, образует двоичную матрицу следующего вида:

0	0	0	0
1	0	0	0
1	1	1	1
1	1	1	1

Изобразите по крайней мере две возможные формы предмета, проходящего мимо датчика.

15.10. В 1980 г. на одной из фирм, занимающейся выпуском определенного изделия, было занято 350 рабочих и произведено 56 000 единиц продукции. Годовой доход от реализации этой продукции составил 29 млн. долл., из которых 3,2 млн. долл. — это чистая прибыль. К 1985 г. фирма реализовала программу модернизации технологического процесса, установив роботы и системы автоматизации. В результате этого в 1985 г. выпуск изделий увеличился до 68 000 единиц, а число рабочих сократилось до 280. Годовой доход от реализации продукции составил 46 млн. долл., чистая прибыль — 4,1 млн. долл. Привело ли внедрение роботов и автоматизация к безработице на этой фирме? В обоснование Вашего ответа приведите любые аргументы, которые считаете разумными. Подтвердите Вашу точку зрения вычислениями.

ПРИЛОЖЕНИЕ А

ПРОГРАММА НА ЯЗЫКЕ ВАЛ ДЛЯ ПРОМЫШЛЕННОГО РОБОТА «ПУМА 560», ПРИМЕНЯЕМОГО НА УЧАСТКЕ ПОТОЧНОЙ СБОРКИ

Эта программа на языке ВАЛ была написана аспирантами Арканзасского университета Джоном Фьюджи и Рэнди Боэнном в университетском Центре робототехники и автоматизации.

Описание установки. Назначение установки заключается в том, чтобы помещать штифт в любое из двух отверстий в деревянном бруске во время движения этого бруска по ленточному конвейеру, как показано на рис. А.1. Поскольку инженера в первую очередь всегда беспокоит производительность (а ее желательно иметь

Рис. А.1

Установка с роботом «Пума» для помещения штифта в отверстие в бруске во время движения последнего по ленточному конвейеру

максимальной), то процесс должен быть организован таким образом, чтобы все операции совершались за минимальное время.

В начале ленты на конвейере установлен фотоэлектрический датчик. Лента движется с постоянной скоростью 1,0 м/с. На конвейере дополнительно смонтирован направляющий желоб, обеспечивающий горизонтальную ориентацию бруска на ленте (рис. А.2).

Самое главное в этой задаче — сообщить роботу о положении бруска на конвейере. Горизонтальная ориентация устанавливается с помощью направляющего желоба, поэтому в программе основное внимание сосредоточено на определении положения бруска в направлении движения конвейера. Фотоэлектрический датчик сообщает роботу, что брусок прошел мимо него. Для того чтобы дать возможность контроллеру робота знать точное положение бруска, робот перемещает свой захват таким образом, чтобы не пропустить брусок, пока он еще находится в желобе. Этот прием называется «калибровкой касания».

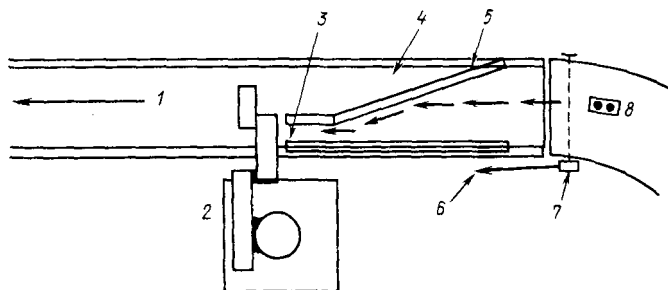


Рис. А.2

Схема, иллюстрирующая действие направляющего желоба, обеспечивающего горизонтальную ориентацию бруска при его движении по ленточному конвейеру: 1 — направление движения; 2 — робот; 3 — место «калибровки касания»; 4 — лента конвейера; 5 — направляющий желоб; 6 — сигнал прерывания к роботу; 7 — фотоэлектрический датчик; 8 — брусок

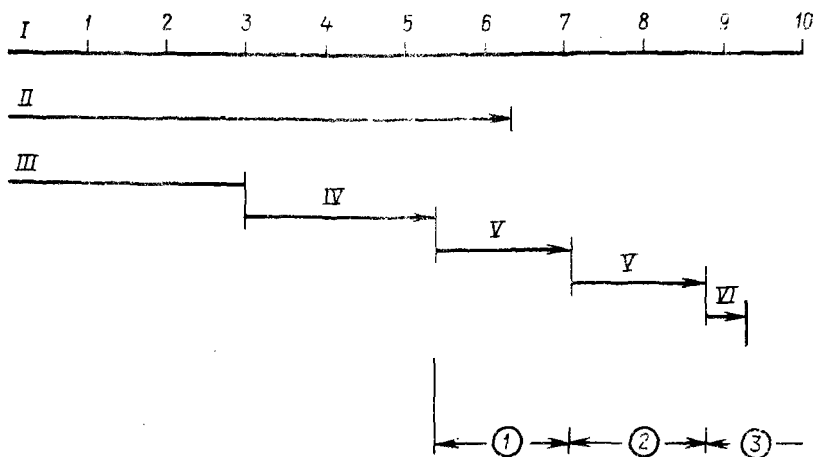


Рис. А.3

Временная диаграмма, связывающая движение бруска и манипуляции робота с использованием калибровки касания:

I — появление сигнала от датчика; *II* — движение бруска к направляющему желобу (6,3 с); *III* — задержка (3,0 с); *IV* — рука робота занимает положение на пути бруска (2,4 с); *V* — пауза 1,7 с (в этом интервале брусок входит в контакт с рукой); *VI* — подъем и опускание руки, установка штифта в отверстие (1,7 с); *VII* — возврат (0,5 с); *V–VII* — движение относительно оси робота; 1–3 — интервалы обработки прерывания

Пояснение программы. Для облегчения анализа программу можно разделить на следующие сегменты:

Шаги программы	Сегмент	Задача
[1–3]	A	Установка робота в состояние готовности
[4–7]	B	Задержка
[8–13]	C	Извлечение штифта из кассеты и подведение его к точке встречи с бруском
[14–16]	D	Пауза для определения точного положения бруска и синхронизации действий робота; опрос датчика (интервал 1)
[17–25]	E	Подъем штифта, позволяющий бруску продолжить движение по ленте, установка штифта в выбранное отверстие; опрос датчика (интервал 2) (Высота, на которую поднимается штифт относительно бруска, настраивается так, чтобы можно было выбрать любое из двух отверстий)
[26–29]	F	Возврат в состояние готовности; опрос датчика (интервал 3)
[30–33]	G	Повторение программы (по сигналу от датчика)

Время, необходимое для прохождения бруска от датчика до точки встречи с ним робота, составляет 6,3 с. Поскольку роботу требуется всего 2,4 с для выполнения сегмента *C*, то в программу введен сегмент *B*, покрывающий эту разницу во времени. Благодаря этому может быть разрешен запрос на прерывание от следующего бруска, пока робот выполняет операции над предыдущим бруском. Так как время необходимой задержки определяется временем выполнения роботом текущей задачи, то сегмент *G* передает управление в различные пункты сегмента *B* (или не-

посредственно сегменту С) в соответствии с тем, на каком интервале поступает очередной запрос на прерывание от датчика.

Листинг программы снабжен построчным комментарием. Временная диаграмма (рис. А.3) иллюстрирует выполнение отдельных сегментов программы.

Программа на языке ВАЛ.

1. SPEED 100 ALWAYS	Установить скорость
2. OPENI	Открыть захват
3. APPRO PIN, 85	Привести робот в состояние готовности
4. 10 WAIT—1	Ожидание прерывания
5. 15 DELAY 0.5	
6. 20 DELAY 0.7	
7. 25 DELAY 1.8	
8. 30 MOVE PIN	Подвести захват к штифту
9. SETI FLAG=0	Сбросить флаг на нуль
10. CLOSEI	Захватить штифт
11. DEPART 85	Отвести захват со штифтом
12. MOVE ADJPOS	Отрегулировать положение руки
13. MOVE HOLDBL	Подвести руку к точке встречи с брусом
14. REACT —1, INT1	Разрешить обработку прерывания
15. DELAY 1.7	Ожидать встречи с брусом
16. IGNORE 1 ALWAYS	Отменить обработку прерывания
17. IF FLAG NE 0 THEN 100	При появлении запроса на прерывание не разрешать его обработку
18. REACT —1, INT2 ALWAYS	Разрешить обработку прерывания
19. 100 SPEED 82 ALWAYS	Отрегулировать скорость установки штифта в отверстие
20. DEPART 800	Разрешить бруску продолжить движение
21. MOVES INSERT	Установить штифт в отверстие
22. OPENI	Открыть захват для следующей установки
23. MOVES RELEASE	Держать захват открытым во время манипуляций
24. SPEED 100 ALWAYS	Вернуться к нормальной скорости
25. IGNORE 1 ALWAYS	Отменить обработку прерывания
26. IF FLAG NE 0 THEN 200	При появлении запроса на прерывание не разрешать его обработку
27. REACT —1, INT3	Разрешить обработку прерывания
28. 200 APPRO PIN, 85	Привести робот в состояние готовности
29. IGNORE 1 ALWAYS	Отменить обработку прерывания
30. IF FLAG EQ 1 THEN 30	Если запрос на прерывание появляется на интервале 1, приступить непосредственно к выполнению программы
31. IF FLAG EQ 2 THEN 25	Если запрос на прерывание появляется на интервале 2, то задать задержку 1,8 и затем приступить к выполнению программы
32. IF FLAG EQ 3 THEN 20	Если запрос на прерывание появляется на интервале 3, то задать задержку 2,5 и затем приступить к выполнению программы
33. GO TO 10	Ожидать запроса на прерывание, если его пока нет
Подпрограмма INT1	
1. SETI FLAG=1	
2. RETURN	
Подпрограмма INT2	
1. SETI FLAG=2	
2. RETURN	
Подпрограмма INT3	
1. SETI FLAG=3	
2. RETURN	

ПРИЛОЖЕНИЕ В

ПРОГРАММА НА ЯЗЫКЕ АРМБЕЙСИК ДЛЯ ДЕМОНСТРАЦИОННОЙ МОДЕЛИ РОБОТА «МИНИМУВЕР-5»

Эта программа на языке АРМБейсик и выставочная модель робота были подготовлены студентом Арканзасского университета Дэвидом Ренни в университетском Центре робототехники и автоматизации.

Описание установки. Выставочный экспонат «Минимувер-5» представляет собой комплекс технических и программных средств, спроектированных с целью продемонстрировать способность простой системы манипулятор — микроЭВМ принимать решения.

Когда перед захватывающим устройством экспоната помещается предмет, фотодатчик обнаруживает его присутствие и сообщает об этом микроЭВМ. Манипулятор, управляемый микроЭВМ, с помощью хватающих движений измеряет толщину предмета. Если этот предмет оказывается монетой, то манипулятор помещает ее в копилку в виде поросенка. Если предмет оказывается деревянным брусом, то манипулятор кладет его в мусорное ведро.

После выполнения этих операций манипулятор возвращается в исходное положение и остается в нем до тех пор, пока перед ним не появится новый предмет.

Каждый раз, когда открывается или закрывается захватывающее устройство, раздается звуковой сигнал. Кроме того, если предпринимается попытка сбить систему с толку ложными действиями, то раздается звуковой сигнал протеста, и манипулятор, забавно покрутив своим захватом, возвращается в исходное положение.

На телевизионный экран выводится сообщение о числе собранных монет, выброшенных брусков и ложных попыток. Выводится также информация о размерах предмета (в дюймах и шагах шагового двигателя), с которым в данный момент работает манипулятор.

Принцип действия. Техническими устройствами экспоната управляет программный модуль. В состав этого модуля входят три программы: 1) программа генерации звуковых сигналов, 2) АРМБейсик и 3) программа управления сбором монет, или ПУСМ.

Программа генерации звуковых сигналов помещает в верхней части ЗУПВ подпрограмму в машинном коде, которая вызывается из ПУСМ каждый раз, когда необходим соответствующий звуковой эффект. Эта программа заставляет центральный процессор выдавать цифровые сигналы на каскадный порт вывода. В управляющем устройстве эти сигналы усиливаются и подаются на громкоговоритель.

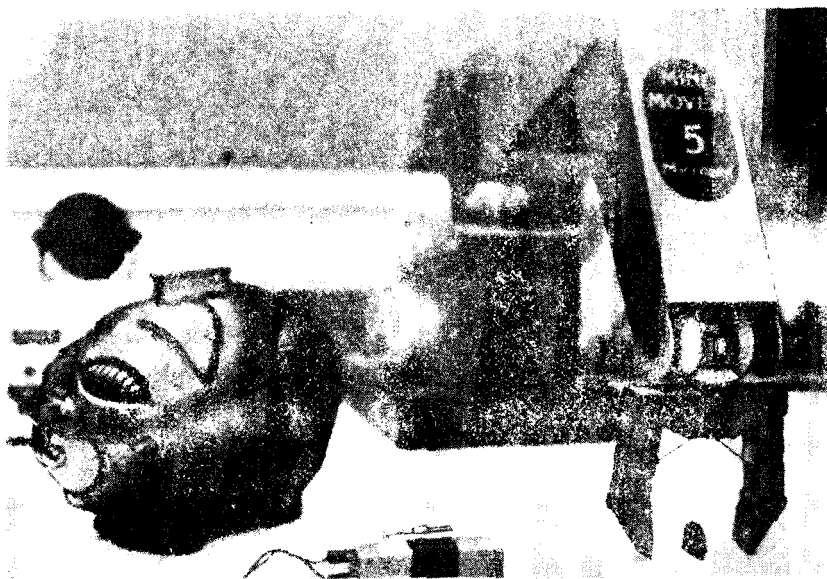


Рис. В.1

Демонстрационный образец робота «Минимувер-5» для сбора монет, созданный в Арканзасском университете

АРМБейсик представляет собой драйвер на машинном языке, который управляет шаговыми двигателями манипулятора и обслуживает счетчики каждого привода. АРМБейсик является неотъемлемой частью робота «Минимувер-5».

Основные функции управления системой выполняет ПУСМ. После того как все три программы загружены в память, производится обучение робота действиям с монетой и бруском, для чего манипулятору вручную задаются соответствующие движения. Последовательность операций записывается в память, и система приводится в состояние готовности.

В режиме готовности ПУСМ циклически опрашивает порт ввода 159. Выход фотодатчика соединен с шиной *D4* порта 159. Когда перед захватывающим устройством помещается предмет, фотодатчик выдает на шину *D4* порта 159 низкий потенциал, и светодиоды сигнализируют о наличии предмета. Затем ПУСМ приказывает манипулятору приблизиться к предмету и измерить его толщину. В ПУСМ происходит подсчет числа шагов, затрачиваемых при сжатии захвата до момента контакта с предметом. В зависимости от результата измерения управление передается одной из трех подпрограмм: манипуляций с монетой, с бруском или протеста. Установка показана на рис. В.1, программа для робота на языке АРМБейсик приведена ниже.

COIN DEMO CONTROL PROGRAM LISTING

```

10 REM *****
20 REM *
30 REM *          MINIMOVER-5
40 REM *
45 REM *          COIN DEMO VER.2
50 REM *          PICK - AND - PLACE
60 REM *
70 REM *          MANUAL TEACH PROGRAM
80 REM *
90 REM *****
91 REM ***** INITIALIZE ROUTINE *****
92 CC=0:BC=0:F=0:S=50:ZX=USR(-250):CLS
93 PRINT:PRINT:PRINT"WAIT FOR GRIPPER TO INITIALIZE
94 @CLOSE:GOSUB7000
95 CLS:PRINT:PRINT:PRINT:PRINT"MANUALLY POSITION GRIPPER 1/4 INCH DIRECTLY
ABOVE THE COIN SLOT":PRINT:PRINT:PRINT"TYPE 0 WHEN DONE
96 @SET
97 CLS:PRINT:PRINT:PRINT:PRINT"WAIT FOR GRIPPER TO OPEN
100 @STEP S,0,0,0,0,100
103 @CLOSE
105 @RESET
110 GOSUB6000:@STEP S,0,0,0,0,750:CLS
130 REM ***** COIN GRASP ROUTINE *****
140 PRINT:PRINT:PRINT:PRINT"PLACE A COIN IN THE COIN SLOT
145 PRINT:PRINT:PRINT"NOW TYPE 0
160 @SET
170 @READ A1,A2,A3,A4,A5,GO
180 @CLOSE:GOSUB 7000: REM CLOSE HAND AND MEASURE PART
190 @READ A1,A2,A3,A4,A5,GC
200 G = GO-GC: REM GRIP SIZE OPEN LESS GRIP SIZE CLOSED
205 CLS:PRINT:PRINT:PRINT
206 PRINT"(GRIPPER SHOULD NOW BE GRASPING THE COIN) SIZE: "GC
207 PRINT:PRINT
210 PRINT"USE 2 & 3 KEYS TO MANUALLY POSITION COIN ABOVE PICKUP SITE":PRINT:PRINT:PRINT
0 WHEN DONE
220 @SET
230 @READ B1,B2,B3,B4,B5
235 CLS:PRINT:PRINT:PRINT
240 PRINT"USE THE 1 KEY TO POSITION COIN ABOVE THE PIGGY BANK"
250 @SET
260 @READ C1,C2,C3,C4,C5

```

```

265 CLS:PRINT:PRINT:PRINT
270 PRINT"USE THE W & E KEYS TO PLACE THE COIN INTO THE BANK SLOT":PRINT:PRINT:PRINT"tY
O WHEN DONE"
280 @SET
290 @READ D1,D2,D3,D4,D5
300 REM
310 REM      RELEASE PART AND RETURN TO B
320 REM
330 GOSUB 6000:@STEP S,0,0,0,0,0,G
340 @STEP S,C1-D1,C2-D2,C3-D3,C4-D4,C5-D5
350 @STEP S,B1-C1,B2-C2,B3-C3,B4-C4,B5-C5
360 GOSUB 2100
370 REM      WAIT UNTIL READY
380 REM
385 CLS:PRINT"          ***** DEMONSTRATION READY *****":PRINT:PRINT"
PLACE COIN OR BLOCK AT PICKUP SITE FOR DEMONSTRATION"
405 PRINT@ 448, "          NUMBER OF UNITS PROCESSED:" FALSE
406 PRINT@576, "          COINS          B LOCKS          FALSE
STARTS":GOSUB 3000
407 PRINT@ 832, "SIZE OF OBJECT BEING PROCESSED---    INCHES:          STEPS:"
408 M=1
410 REM ***** LOOP HERE UNTIL TARGET PRESENT *****
411 L=INP(159)
413 OUT158,M:GOSUB 26000:M=M*2:IF M=16 THEN 408
415 IF L=31 THEN GOSUB 25000 ELSE 410
420 REM ***** DETERMINE SIZE OF TARGET *****
430 REM
1000 @STEP S,A1-B1,A2-B2,A3-B3,A4-B4,A5-B5
1001 @READ X1,X2,X3,X4,X5,G0
1005 @CLOSE:ZX=USR(20)
1010 @READ X1,X2,X3,X4,X5,G0
1011 G=GO-GC
1012 IF GC < 5 THEN GOSUB 5000 ELSE 1015
1013 GOTO 410
1015 IF GC < 100 THEN GOSUB 1020 ELSE GOSUB 21000
1019 GOTO 410
1020 REM ***** COIN PLACEMENT SUBROUTINE *****
1021 GOSUB 4000
1022 @STEP S,B1-A1,B2-A2,B3-A3,B4-A4,B5-A5
1025 G=GO-GC
1030 @STEP S,C1-B1,C2-B2,C3-B3,C4-B4,C5-B5
1040 @STEP S,D1-C1,D2-C2,D3-C3,D4-C4,D5-C5
1050 GOSUB 6000:@STEP S,0,0,0,0,0,G
1060 @STEP S,C1-D1,C2-D2,C3-D3,C4-D4,C5-D5
1070 @STEP S,B1-C1,B2-C2,B3-C3,B4-C4,B5-C5
1073 CC=CC+1
1074 GOSUB 3000
1080 GOTO 410
1085 RETURN
1090 END
2000 REM ***** THIS LOOP TESTS INPUT PORT 158 *****
2001 A=INP(159):PRINTA
2002 GOTO2000
2100 REM ***** BLOCK TEACH SUBROUTINE *****
2105 CLS
2106 PRINT:PRINT:PRINT:PRINT "PLACE WOOD BLOCK ON PICKUP SITE " : PRINT:PRINT:PRINT"TYI
O WHEN DONE"
2109 @READ X1,X2,X3,X4,X5
2110 @SET
2140 @STEP S,A1-B1,A2-B2,A3-B3,A4-B4,A5-B5
2150 @CLOSE:
2160 @STEP S,B1-A1,B2-A2,B3-A3,B4-A4,B5-A5
2170 CLS
2230 PRINT:PRINT:PRINT:PRINT"(GRIPPER SHOULD NOW HAVE BOCK ABOVE PICKUP
SITE) SIZE "GC
2240 PRINT:PRINT:PRINT "USE THE Q KEY TO POSITION BLOCK ABOVE RAMP ":PRINT:PRINT:PRINT"
O WHEN DONE"

```



```

2250 @SET
2260 @READ H1,H2,H3,H4,H5
2270 CLS:PRINT:PRINT:PRINT:PRINT "USE THE W E & T KEYS TO POSITION
THE BLOCK ONTO THE RAMP":PRINT:PRINT:PRINT:PRINT "TYPE Q WHEN DONE"
2280 @SET
2290 @READ I1,I2,I3,I4,I5
2330 GOSUB 6000:@STEP S,0,0,0,0,0,G
2340 @STEP S,H1-I1,H2-I2,H3-I3,H4-I4,H5-I5
2350 @STEP S, B1-H1,B2-H2,B3-H3,B4-H4,B5-H5
2390 RETURN
3000 REM ***** PRINT TABULATIONS SUBROUTINE *****
3010 PRINT@710,CC;
3020 PRINT@732,BC;
3030 PRINT@756,F;
3050 RETURN
4000 REM ***** PRINT MEASUREMENT SUBROUTINE *****
4010 PRINT@ 998," ";
4020 PRINT@ 998,GC/361;:PRINT@ 1017,GC;
4050 RETURN
5000 REM ***** FALSE ALARM SUBROUTINE *****
5001 GOSUB 4000
5005 @STEP 50,0,0,0,0,0,G
5006 GOSUB 22000
5010 @STEP 50,0,0,0,50,50,0
5020 FOR CN=1TO5
5022 @STEP 50,0,0,0,-100,-100,0
5030 @STEP 50,0,0,0,100,100,0
5036 NEXT
5040 @STEP 50, 0,0,0,-50,-50,0
5041 @STEP 50, B1-A1,B2-A2,B3-A3,B4-A4,B5-A5
5045 F=F+1
5046 GOSUB 3000
5060 RETURN
6000 REM ***** RELEASE SOUND SUBROUTINE *****
6010 ZX=USR(40):RETURN
7000 REM ***** GRASP SOUND SUBROUTINE *****
7010 ZX=USR(20):RETURN
21000 REM ***** BLOCK PLACEMENT SUBROUTINE *****
21005 GOSUB 4000
21015 @READ X1,X2,X3,X4,X5,GC
21022 @STEP S,B1-A1,B2-A2,B3-A3,B4-A4,B5-A5
21030 @STEP S,H1-B1,H2-B2,H3-B3,H4-B4,H5-B5
21040 @STEP S,I1-H1,I2-H2,I3-H3,I4-H4,I5-H5
21050 GOSUB 6000:@STEP 50,0,0,0,0,0,G
21060 @STEP S,H1-
21070 @STEP S,B1-H1,B2-H2,B3-H3,B4-H4,B5-H5
21081 RETURN
21090 END
22000 REM ***** RASPBERRIES SUBROUTINE *****
22001 ZX=USR(-5000):ZX=USR(60):ZX=USR(5)
22100 ZX=USR(-5000):ZX=USR(150):ZX=USR(200)
22110 ZX=USR(-250)
22200 RETURN
23000 REM ***** SOUND INITIALIZE *****
23010 FOR HZ=1TO100:NEXT: ZX=USR(-250)
23100 ZX=USR(20):FORNN=1TO50:NEXT:ZX=USR(40)
23300 RETURN
25000 REM ***** TIMER SUBROUTINE *****
25001 FOR N=1TO200:NEXT:RETURN
26000 REM *** DELAY SUBROUTINE ***
26001 FOR N=1TO150:NEXT:RETURN

```

ПРИЛОЖЕНИЕ С

ПРОГРАММА НА ЯЗЫКЕ ВАЛ ДЛЯ МАНИПУЛИРОВАНИЯ С КУБИКОМ РУБИКА ПРИ ПОМОЩИ ПРОМЫШЛЕННОГО РОБОТА «ПУМА 560»

Эта программа была написана аспирантом Арканзасского университета Дэвидом М. Ренни в университетском Центре робототехники и автоматизации. Здесь приведено также описание установки, в которой робот управляется от микроЭВМ верхнего уровня. Материал использован с разрешения Дэвида М. Ренни.

Введение. Промышленный робот «Юнимэйт Пума 560 Марк II» был запрограммирован для решения задачи на сборку кубика Рубика. ПК робота воспринимает команды от микроЭВМ верхнего уровня и выполняет программу манипуляций, необходимых для сборки кубика из произвольного начального состояния. МикроЭВМ верхнего уровня «Зенит Z-89» с помощью подсистемы технического зрения получает информацию о начальной конфигурации цветов граней кубика и вычисляет последовательность действий, которую сообщает ПК робота через последовательный интерфейс.

Кубик Рубика — это всемирно известная головоломка. Каждая из его шести граней разделена на девять равных ячеек, в свою очередь являющихся гранями меньших кубиков. В «нетронутом» состоянии каждая из граней кубика имеет вполне определенный однородный цвет. 26 маленьких видимых кубиков соединены таким образом, что каждые девять из них, образующие основные грани, можно вращать как единую группу. Поворот любой из граней кубика приводит к частичному перераспределению ячеек на четырех прилегающих гранях. Таким образом можно быстро нарушить упорядоченное распределение цветов ячеек на гранях кубика. Цель головоломки — вернуть кубик в исходное упорядоченное состояние (собрать кубик).

Кубик Рубика особенно привлекателен тем, что последовательность манипуляций, необходимых для его сборки, различна для каждого из приблизительно $43 \cdot 10^{18}$ возможных начальных состояний. Компьютер, каждую микросекунду генерирующий новую конфигурацию цветов граней кубика, для образования всех возможных конфигураций затратил бы более одного миллиона лет.

Несмотря на это, разработано несколько алгоритмов сборки кубика, с помощью которых отдельные люди ухитряются решить головоломку менее чем за минуту. Разработаны также многочисленные программы для микроЭВМ, предназначенные для имитации кубика Рубика и решения головоломки.

В Иллинойском университете и в лабораториях «Пасифик Нортвест» в Беттеле созданы специализированные машины, способные логически и физически собирать кубик [19, 59]. В данной

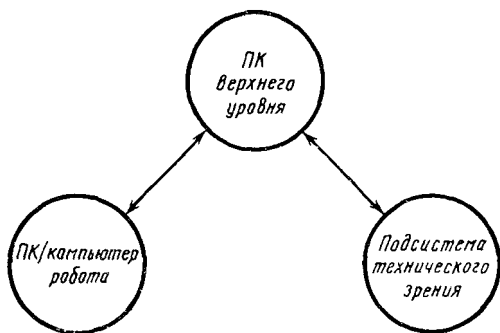


Рис. С.1

Три основные подсистемы, образующие систему для сборки кубика Рубика

работе описано, как решается та же задача с помощью промышленного робота общего назначения, сопряженного с микроЭВМ.

Описание системы.

Система включает в себя ПК верхнего

уровня вместе с подсистемами робота и технического зрения (рис. С.1). Подсистема технического зрения определяет начальную конфигурацию цветов на гранях кубика. Манипулятор робота физически устанавливает кубик в такие положения, чтобы подсистема технического зрения могла осмотреть каждую из его шести граней. Манипулятор также осуществляет поворот граней кубика, чтобы привести его в упорядоченное состояние. ПК верхнего уровня использует входные данные для вычисления последовательности операций по сборке кубика в соответствии с алгоритмом решения на языке Бейсик [14]. Другие программы ПК вырабатывают набор командных инструкций, которые затем выдаются в подсистему робота. Ниже более детально рассматривается каждая из подсистем.

Подсистема робота. Подсистема робота — это промышленная модель «Юнимэйт Пума 560 Марк II». Действиями робота управляет ЭВМ, использующая язык ВАЛ, разработанный специально для моделей этого семейства. Помимо манипулятора подсистема включает в себя ПК, дисковый накопитель для хранения программ, обучающий пульт, терминал с телевизионным дисплеем и модуль ввода-вывода (для сопряжения с внешними устройствами).

Фиксирующее устройство (рис. С.2) жестко удерживает два нижних слоя кубика так, чтобы его верхний слой (грань) можно было поворачивать. О правильной установке кубика в фиксирующем устройстве сигнализирует специальный контактный датчик. Сигнал от этого датчика через модуль ввода-вывода поступает в ПК и используется для контроля правильности действий манипулятора. Специальное приспособление позволяет роботу захватить и повернуть одну грань кубика или поднять его и затем установить в фиксирующем устройстве в новое положение (рис. С.3).

Для того чтобы повернуть грань кубика, манипулятор сначала помещает его в фиксирующее устройство, причем грань, которая подлежит вращению, должна быть верхней, как показано на рис. С.2. После этого манипулятор может повернуть эту грань

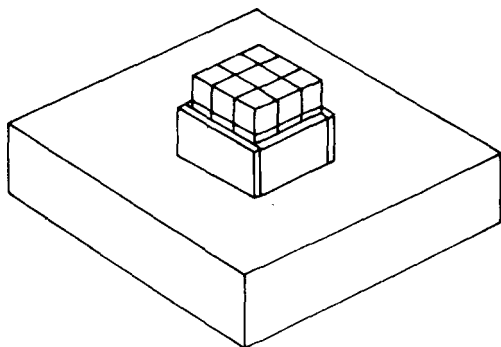


Рис. С.2

Фиксирующее устройство жестко удерживает два нижних слоя кубика, так что манипулятор может вращать только верхнюю грань

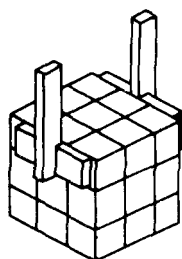


Рис. С.3

Простой Т-образный захват для установки кубика в фиксирующее устройство, извлечения его оттуда и вращения верхней грани

на 90° по часовой стрелке, на 90° против часовой стрелки или на 180° , в зависимости от выполняемой команды, которая вычисляется в соответствии с ВАЛ-программами, хранящимися в памяти ПК. Например, каждый раз, когда необходимо повернуть верхнюю грань кубика на 90° по часовой стрелке, вызывается программа FPLUS:

```
PROGRAM FPLUS
1. SETI NEED=RFR
2. SETI DIR =1
3. GOSUB GFT
4. RETURN
```

В строке 1 этой программы параметр NEED устанавливается равным RFR, что соответствует конфигурации цветов на той грани кубика, которая оказалась перед видеокамерой при инициализации его начального состояния (фасад кубика). Во время манипуляций с кубиком его фасад в действительности может занять и какое-то иное положение. В строке 2 параметру DIR придается значение, соответствующее требуемому направлению вращения фасада: 1 — на 90° по часовой стрелке, 2 — на 90° против часовой стрелки, 3 — на 180° . После этого вызывается подпрограмма GFT (поворот фасада).

В строках с 1 по 6 происходит определение фактического положения фасада кубика с помощью переменных TOP, BOT, LT, RT, FR и POS. Например, если фасад оказался внизу, то переменная BOT в строке 2 имеет значение NEED (BOT = NEED = RFR). Следовательно, управление будет передано метке 10. Подпрограмма BUP заставит манипулятор перевернуть кубик в фиксирующем устройстве вверх ногами, чтобы фасад

.PROGRAM GFT

```
1.      IF TOP EQ NEED THEN 60
2.      IF BOT EQ NEED THEN 10
3.      IF LT EQ NEED THEN 20
4.      IF RT EQ NEED THEN 30
5.      IF FR EQ NEED THEN 40
6.      IF POS EQ NEED THEN 50
7.      10 GOSUB BUP
8.      GOTO 60
9.      20 GOSUB LUP
10.     GOTO 60
11.     30 GOSUB RUP
12.     GOTO 60
13.     40 GOSUB FUP
14.     GOTO 60
15.     50 GOSUB PUP
16.     GOTO 60
17.     60 IF DIR EQ 1 THEN 70
18.         IF DIR EQ 2 THEN 80
19.         IF DIR EQ 3 THEN 90
20.     70 GOSUB CW
21.     RETURN
22.     60 GOSUB CW
23.     RETURN
24.     90 GOSUB CW2
25.     RETURN
```

.END

оказался наверху. Подпрограмма BUP производит также изменение значений переменных TOP, BOT, LT, RT, FR и POS в соответствии с новой ориентацией кубика. В строках с 17 по 19 происходит определение требуемого направления поворота верхней грани, далее вызывается соответствующая подпрограмма, определяющая действия манипулятора, после чего осуществляется возврат в основную программу.

ПК верхнего уровня. В качестве ПК верхнего уровня используется микроЭВМ «Зенит Z-89», работающая под управлением операционной системы CP/M. В стандартной конфигурации системы команды ПК робота сообщаются пользователем с терминала, оснащенного телевизионным дисплеем (рис. С.4).

В этом эксперименте ЭВМ верхнего уровня сопрягается с портом ввода ПК робота и по сути заменяет собой терминал последнего. Полностью схема всей системы изображена на рис. С.5.

ПК верхнего уровня действует следующим образом: кубик Рубика, выведенный из упорядоченного состояния, помещается в фиксирующее устройство, контактный датчик которого инициирует работу ПК. Затем выполняется программа VIDEO. BAS, заставляющая манипулятор повернуть каждую из шести граней кубика к видеокамере, чтобы определить на них цвет каждой из девяти ячеек. После того как осмотрены все шесть граней кубика, его начальное состояние становится известным ПК. В па-

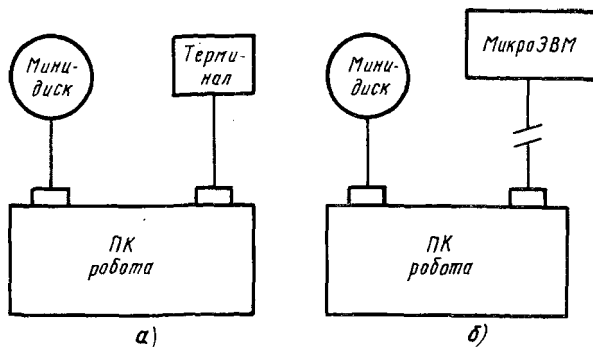


Рис. С.4

Сопряжение микроЭВМ с ПК робота:
а — стандартная схема; б — сопряжение с микроЭВМ

мать с диска загружается программа RUBIK. BAS, реализующая алгоритм решения задачи, и начинается ее выполнение. Фрагмент листинга типичной последовательности операций имеет вид

$L + F2 \quad B - T + L - T - B +$

Эта последовательность означает «повернуть левую грань по часовой стрелке, фасад на 180° , нижнюю грань против часовой стрелки, верхнюю грань по часовой стрелке, ... и т. д.». Каждой команде, вырабатываемой алгоритмом решения, соответствует определенная программа ПК на языке ВАЛ.

После того как определена последовательность операций по сборке кубика, выполняется завершающая программа MAIN. BAS

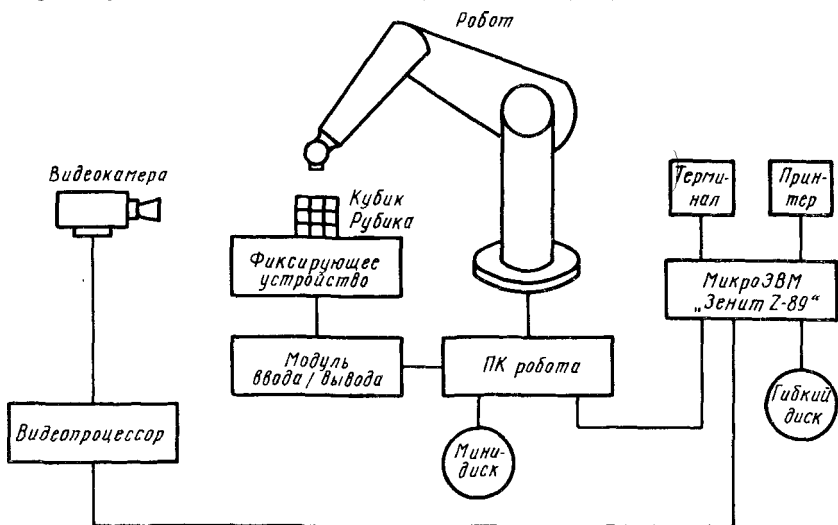


Рис. С.5

Функциональная схема системы

которая преобразует эту последовательность в командные строки на языке ВАЛ и посылает их через последовательный порт ПК робота. Прежде чем послать следующую команду, программа MAIN. BAS ждет поступления от ПК робота подтверждения об успешном выполнении предыдущей программы. Кубик будет собран после выполнения всей рассчитанной последовательности команд.

Подсистема технического зрения. Проектом не предусматривалась разработка видеопроцессора, поэтому в эксперименте вход от подсистемы технического зрения имитировался заданием начальной конфигурации цветов граней с клавиатуры терминала.

Алгоритм решения. Приведенный алгоритм решения был выбран ввиду его доступности, простоты отладки и пригодности для работы с операционной системой Зенит-СР/М.

Выводы. При предварительных испытаниях система успешно справлялась с задачей по сборке кубика, затрачивая на это 30 мин и менее, без учета времени, необходимого для вычисления решающей последовательности и оценки начального состояния. Улучшение временных параметров, траекторий и скоростей рабочих органов манипулятора должно привести к ускорению решения задачи. Кроме того, вовсе нельзя утверждать, что использованный алгоритм решения является оптимальным; другие алгоритмы могут дать лучшие результаты.

Данный эксперимент демонстрирует возможность успешного сопряжения промышленного робота общего назначения с микрокомпьютерной системой с целью решения сложной задачи. Он иллюстрирует также потенциальные возможности современных робототехнических устройств; подобный тип искусственного интеллекта в сочетании с различными методами сопряжения и обработкой распределенных данных может быть с успехом распространен на такие прикладные производственные задачи, как контроль, сортировка и сборка изделий, дистанционная обработка и оперативное телеуправление процессами.

Приведенная программа на языке ВАЛ позволяет роботу «Пума» собрать кубик Рубика в соответствии с командами, вырабатываемыми ЭВМ верхнего уровня. Для студента она является исчерпывающим примером программирования на языке ВАЛ. Однако это лишь та часть программного обеспечения, которая использует язык ВАЛ; в листинг не включены программы на Бейсике или Ассемблере для связи между ЭВМ верхнего уровня и ПК робота, для ввода данных о начальном состоянии кубика и для реализации алгоритма сборки кубика. Чтобы воспроизвести эксперимент, потребуется написать эти программы на Бейсике или Ассемблере для конкретной машины, выбранной для управления роботом.

TYPE C:RUBIK.PRM

LISTP

.PROGRAM AREM

```
1.    REMARK *****
2.    REMARK
3.    REMARK          RUBIK'S CUBE MANIPULATION ROUTINES
4.    REMARK
5.    REMARK          WRITTEN BY DAVID M. RANNEY
6.    REMARK
7.    REMARK          UNIVERSITY OF ARKANSAS EXPERIMENT STATION
8.    REMARK
9.    REMARK          SPRING SEMESTER 1984
10.   REMARK
11.   REMARK *****
.END
```

.PROGRAM BUP

```
1.    REMARK *****
2.    REMARK  THIS ROUTINE PUTS THE PRESENT BOTTOM FACE UP
3.    REMARK *****
4.    GOSUB CLOSE
5.    MOVES ABOVE
6.    MOVES CLEARCAM
7.    MOVES NEWCUBE
8.    OPEN , 0.00
9.    DELAY 0.50
10.   MOVES CLEARCAM
11.   MOVE INSIDESO
12.   MOVES NEWCUBESO
13.   CLOSE , 0.00
14.   DELAY 0.50
15.   MOVES INSIDESO
16.   MOVE CLEARCAM
17.   MOVES NEWCUBE
18.   OPEN , 0.00
19.   DELAY 0.50
20.   MOVES CLEARCAM
21.   GOSUB TURNCUBEUP
22.   SETI BUFFER = TOP
23.   SETI BUFFER2 = FR
24.   SETI TOP = BOT
25.   SETI BOT = BUFFER
26.   SETI FR = POS
27.   SETI POS = BUFFER2
28.   RETURN 0
.END
```

.PROGRAM BPLUS

```
1.    REMARK ***** BACK FACE *****
2.    SETI NEED = RPOS
3.    SETI DIR = 1
4.    GOSUB GFT
5.    RETURN 0
.END
```

.PROGRAM BMINUS

```
1.    REMARK ***** BACK FACE *****
2.    SETI NEED = RPOS
3.    SETI DIR = 2
4.    GOSUB GFT
5.    RETURN 0
.END
```

PROGRAM BZ

```
1.    REMARK ***** BACK FACE *****
2.    SETI NEED = RPOS
3.    SETI DIR = 3
4.    GOSUB GFT
5.    RETURN 0
.END
```



```

.PROGRAM CLOSE
1.      REMARK *****
2.      REMARK *****CLOSE IF CUBE HOME...ELSE PRINT MESSAGE *****
3.      REMARK *****
4.      IFSIG -2, , , THEN 50
5.      30 CLOSE , 0.00
6.      DELAY 0.30
7.      RETURN 0
8.      50 TYPE THE CUBE IS NOT IN THE HOME POSITION !!!!
9.      80 IFSIG 2, , , THEN 80
10.     GOTO 60
11.     80 DELAY 3.00
12.     GOTO 30
.END

```

```

.PROGRAM CW
1.      REMARK *****
2.      REMARK THIS ROUTINE TURNS ACTUAL UP FACE CW
3.      REMARK *****
4.      GOSUB CLOSE
5.      MOVES CW
6.      OPEN , 0.00
7.      DELAY 0.50
8.      MOVES ABOVECW
9.      MOVES ABOVE
10.     MOVES CUBE
11.     RETURN 0
.END

```

```

.PROGRAM CCW
1.      REMARK *****
2.      REMARK THIS ROUTINE TURNS THE PRESENT UP FACE CCW
3.      REMARK *****
4.      GOSUB CLOSE
5.      MOVES CCW
6.      OPEN , 0.00
7.      DELAY 0.50
8.      MOVES ABOVECCW
9.      MOVES ABOVE
10.     MOVES CUBE
11.     RETURN 0
.END

```

```

.PROGRAM CW2
1.      REMARK *****
2.      REMARK THIS ROUTINE TURNS THE ACTUAL TOP FACE 180 DEGREES
3.      REMARK *****
4.      GOSUB CLOSE
5.      MOVES CUBE180
6.      OPEN , 0.00
7.      DELAY 0.50
8.      MOVES ABOVE180
9.      MOVES ABOVE
10.     MOVES CUBE
11.     RETURN 0
.END

```

```

.PROGRAM CPLUB
1.      REMARK *****
2.      REMARK REDEFINES REFERENCED FACES
3.      REMARK *****
4.      SETI BUFFER = RTOP
5.      SETI RTOP = RLT
6.      SETI RLT = RDOT
7.      SETI RDOT = RRT
8.      SETI RRT = BUFFER
9.      RETURN 0
.END

```

```

PROGRAM CMINUS
1.    REMARK *****
2.    REMARK REDEFINES REFERENCED FACES
3.    REMARK *****
4.    SETI BUFFER = RTOP
5.    SETI RTOP = RRT
6.    SETI RRT = RBOT
7.    SETI RBOT = RL1
8.    SETI RL1 = BUFFER
9.    RETURN 0
.END

PROGRAM C2
1.    REMARK *****
2.    REMARK REDEFINE REFERENCED FACES
3.    REMARK *****
4.    GOSUB CPLUS
5.    GOSUB CPLUS
6.    RETURN 0
.END

PROGRAM DPLUS
1.    REMARK ***** DOWN FACE *****
2.    SETI NEED = RBOT
3.    SETI DIR = 1
4.    GOSUB GFT
5.    RETURN 0
.END

PROGRAM DMINUS
1.    REMARK ***** DOWN (BOTTOM) FACE *****
2.    SETI NEED = RBOT
3.    SETI DIR = 2
4.    GOSUB GFT
5.    RETURN 0
.END

PROGRAM D2
1.    REMARK *****DOWN (BOTTOM) FACE *****
2.    SETI NEED = RBOT
3.    SETI DIR = 3
4.    GOSUB GFT
5.    RETURN 0
.END

PROGRAM END
1.    REMARK *****
2.    REMARK MOVE TO CLEARCAM POSITION
3.    REMARK *****
4.    MOVE CLEARCAM
5.    RETURN 0
.END

PROGRAM FUP
1.    REMARK *****
2.    REMARK THIS ROUTINE PLACES THE PRESENT FRONT FACE UP
3.    REMARK *****
4.    GOSUB CLOBE
5.    MOVES ABOVE
6.    MOVES CLEARCAM
7.    MOVES NEWCUBE
8.    OPEN , 0.00
9.    DELAY 0.50
10.   MOVES CLEARCAM
11.   GOSUB TURNcubeUP
12.   SETI BUFFER = TOP
13.   SETI TOP = FR
14.   SETI FR = BOT
15.   SETI BOT = POS
16.   SETI POS = BUFFER
17.   RETURN 0
.END

```

```

.PROGRAM FPLUS
  1.    REMARK ***** FRONT FACE *****
  2.    SETI NEED = RFR
  3.    SETI DIR = 1
  4.    GOSUB GFT
  5.    RETURN 0
.END

.PROGRAM FMINUS
  1.    REMARK ***** FRONT FACE *****
  2.    SETI NEED = RFR
  3.    SETI DIR = 2
  4.    GOSUB GFT
  5.    RETURN 0
.END

.PROGRAM F2
  1.    REMARK ***** FRONT FACE *****
  2.    SETI NEED = RFR
  3.    SETI DIR = 3
  4.    GOSUB GFT
  5.    RETURN 0
.END

.PROGRAM FPOS
  1.    REMARK *****
  2.    REMARK MOVES FRONT FACE TO POSTERIOR POSITION
  3.    REMARK *****
  4.    GOSUB CLOSE
  5.    MOVES ABOVE
  6.    MOVES ABOVE180
  7.    MOVES CUBE180
  8.    OPEN , 0.00
  9.    DELAY 0.50
 10.    MOVES ABOVE180
 11.    MOVES ABOVE
 12.    MOVES CUBE
 13.    RETURN 0
.END

.PROGRAM GFT
  1.    REMARK *****
  2.    REMARK ***** GET-FACE-TURN ROUTINE *****
  3.    REMARK *****
  4.    IF TOP EQ NEED THEN 60
  5.    IF BOT EQ NEED THEN 10
  6.    IF LT EQ NEED THEN 20
  7.    IF RT EQ NEED THEN 30
  8.    IF FR EQ NEED THEN 40
  9.    IF POS EQ NEED THEN 50
 10.    10 GOSUB BUP
 11.    GOTO 60
 12.    20 GOSUB LUP
 13.    GOTO 60
 14.    30 GOSUB RUP
 15.    GOTO 60
 16.    40 GOSUB FUP
 17.    GOTO 60
 18.    50 GOSUB PUP
 19.    GOTO 60
 20.    60 IF DIR EQ 1 THEN 70
 21.        IF DIR EQ 2 THEN 80
 22.        IF DIR EQ 3 THEN 90
 23.    70 GOSUB CW
 24.    RETURN 0
 25.    80 GOSUB CCW
 26.    RETURN 0
 27.    90 GOSUB CW2
 28.    RETURN 0
.END

```

```

PROGRAM INITIAL
1.  REMARK *****
2.  REMARK ***** INITIALIZATION SUBROUTINE *****
3.  REMARK *****:
4.  BASE 0.00, 0.00, 0.00, 0.00
5.  SPEED 30.00
6.  DEPARTS 100.00
7.  HERE ABOVE
8.  MOVES CUBE
9.  GOSUB CLOSE
10. MOVES ABOVE
11. SHIFT ABOVE BY 100.00, 0.00, 0.00
12. SET CLEARCAM = ABOVE
13. SHIFT ABOVE BY -100.00, 0.00, 0.00
14. MOVES CLEARCAM
15. DRIVE B, 50.00, 50.00
16. HERE CLEARCAMCW
17. DRIVE B, 90.00, 50.00
18. HERE CLEARCAM180
19. MOVES CLEARCAM
20. DRIVE B, -50.00, 50.00
21. HERE CLEARCAMCCW
22. MOVES CLEARCAM
23. DEPARTS -100.00
24. HERE NEWCUBE
25. DRIVE B, 50.00, 50.00
26. HERE NEWCUBECH
27. DRIVE B, 50.00, 50.00
28. HERE NEWCUBE180
29. MOVES NEWCUBE
30. DRIVE B, -50.00, 50.00
31. HERE NEWCUBECCW
32. MOVES NEWCUBE
33. SHIFT NEWCUBE BY 166.00, 0.00, 0.00
34. SET INSIDE = NEWCUBE
35. SHIFT NEWCUBE BY -166.00, 0.00, 0.00
36. OPEN , 0.00
37. DELAY 0.50
38. MOVES INSIDE
39. DRIVE S, 52.000, 50.00
40. HERE INSIDE90
41. SHIFT INSIDE90 BY 0.00, 0.00, -150.00
42. SET NEWCUBE90 = INSIDE90
43. SHIFT INSIDE90 BY 0.00, 0.00, 190.00
44. MOVES NEWCUBE90
45. CLOSE , 0.00
46. DELAY 0.50
47. MOVES INBIDE90
48. MOVE CLEARCAM
49. MOVEB ABOVE
50. MOVES CUBE
51. OPEN , 0.00
52. DELAY 3.00
53. GOSUB CLOSE
54. DRIVE B, 50.00, 50.00
55. HERE CW
56. DRIVE B, 50.00, 80.00
57. HERE CUBE180
58. MOVES CUBE
59. DRIVE B, -50.00, 80.00
60. HERE CCW
61. MOVES CUBE
62. DEPARTS 100.00
63. DRIVE B, 90.00, 80.00
64. HERE ABOVECH
65. DRIVE B, 90.00, 80.00
66. HERE ABOVE180
67. MOVES ABOVE
68. DRIVE B, -50.00, 80.00
69. HERE ABOVECCW
70. MOVES ABOVE
71. MOVES CUBE
72. OPEN , 0.00
73. DELAY 0.50
74. SPEED 100.00
.END

```

```

PROGRAM LUP
1.    REMARK *****
2.    REMARK *** THIS SUBROUTINE PUTS THE PRESENT LEFT FACE UP ***
3.    REMARK *****
4.    GOSUB CLOSE
5.    MOVES ABOVE
6.    MOVES ABOVECCW
7.    MOVES CLEARCAMCCW
8.    MOVES NEWCUBECCW
9.    OPEN , 0.00
10.   DELAY 0.50
11.   MOVES CLEARCAMCCW
12.   GOSUB TURNCUBEUP
13.   SETI BUFFER = TOP
14.   SETI BUFFER2 = RT
15.   SETI TOP = LT
16.   SETI LT = POB
17.   SETI POB = BUFFER
18.   SETI RT = FR
19.   SETI FR = BOT
20.   SETI BOT = BUFFER2
21.   RETURN 0
.END

.PROGRAM LPLUS
1.    REMARK ***** LEFT FACE *****
2.    SETI NEED = RLT
3.    SETI DIR = 1
4.    GOSUB GFT
5.    RETURN 0
.END

.PROGRAM LMINUS
1.    REMARK ***** LEFT FACE *****
2.    SETI NEED = RLT
3.    SETI DIR = 2
4.    GOSUB GFT
5.    RETURN 0
.END

.PROGRAM L2
1.    REMARK ***** LEFT FACE 180 *****
2.    SETI NEED = RLT
3.    SETI DIR = 3
4.    GOSUB GFT
5.    RETURN 0
.END

.PROGRAM LPOS
1.    REMARK *****
2.    REMARK THIS ROUTINE TURNS THE LEFT FACE TO POB FACE
3.    REMARK *****
4.    GOSUB CLOSE
5.    MOVES ABOVE
6.    MOVES ABOVECW
7.    MOVES CW
8.    OPEN , 0.00
9.    DELAY 0.50
10.   MOVES ABOVECW
11.   MOVES ABOVE
12.   MOVES CUBE
13.   RETURN 0
.END

.PROGRAM ORIENT
1.    REMARK *****
2.    REMARK ASSIGNS FACE COLORS TO ACTUAL POSITIONS AND INITIAL
3.    REMARK FACE COLORS TO REFERENCED POSITIONS
4.    REMARK *****
5.    SETI TOP = 1
6.    SETI RTOP = TOP
7.    SETI BOT = 2
8.    SETI RBOT = BOT
9.    SETI LT = 5

```

```

10.      SETI .RLT = LT
11.      SETI RT = R
12.      SETI RRT = RT
13.      SETI FR = F
14.      SETI RFR = FR
15.      SETI POS = P
16.      SETI RPOS = POS
17.      RETURN 0
.END

,PROGRAM PUP
1.      REMARK *****
2.      REMARK THIS ROUTINE PUTS THE PRESENT POSTERIOR FACE UP
3.      REMARK *****
4.      GOSUB CLOSE
5.      MOVES ABOVE
6.      MOVES CLEARCAM180
7.      MOVES NEWCUBE180
9.      OPEN , 0.00
9.      DELAY 0.50
10.     MOVES CLEARCAM180
11.     MOVE INSIDES0
12.     MOVES NEWCUBES0
13.     CLOSE , 0.00
14.     DELAY 0.50
15.     MOVES INSIDES0
16.     MOVE CLEARCAM
17.     MOVES ABOVE
18.     MOVES CUBE
19.     OPEN , 0.00
20.     DELAY 0.50
21.     SETI BUFFER = TOP
22.     SETI BUFFER2 = LT
23.     SETI BUFFER3 = BOT
24.     SETI TOP = POS
25.     SETI POS = BUFFER
26.     SETI LT = RT
27.     SETI RT = BUFFER2
28.     SETI BOT = FR
29.     SETI FR = BUFFER3
30.     RETURN 0
.END

,PROGRAM P7
1.      REMARK *****
2.      REMARK P7 COMMAND -- RUBIK'S MANEUVER
3.      REMARK *****
4.      GOSUB YMINUS
5.      GOSUB UPLUS
6.      GOSUB YMINUS
7.      GOSUB UPLUS
8.      GOSUB YMINUS
9.      GOSUB U2
10.     GOSUB YPLUS
11.     GOSUB UPLUS
12.     GOSUB YPLUS
13.     GOSUB UPLUS
14.     GOSUB YPLUS
15.     GOSUB U2
16.     RETURN 0
.END

,PROGRAM P0
1.      REMARK ***** P0 COMMAND *****
2.      GOSUB UPLUS
3.      GOSUB F2
4.      GOSUB UMINUS
5.      RETURN 0
.END

```

```

.PROGRAM P1
1.    REMARK ***** P1 COMMAND *****
2.    GOSUB RMINUS
3.    GOSUB F2
4.    GOSUB RPLUS
5.    RETURN 0
.END

.PROGRAM P2
1.    REMARK ***** P2 COMMAND *****
2.    GOSUB F2
3.    GOSUB RMINUS
4.    GOSUB FPLUS
5.    GOSUB RPLUS
6.    GOSUB FMINUS
7.    RETURN 0
.END

.PROGRAM P3
1.    REMARK ***** P3 COMMAND *****
2.    GOSUB XMINUS
3.    GOSUB FPLUS
4.    GOSUB XPLUS
5.    RETURN 0
.END

.PROGRAM P4
1.    REMARK ***** P4 COMMAND *****
2.    GOSUB FPLUS
3.    GOSUB XMINUS
4.    GOSUB F2
5.    GOSUB XPLUS
6.    RETURN 0
.END

.PROGRAM P5
1.    REMARK ***** P5 COMMAND *****
2.    GOSUB LMINUS
3.    GOSUB ZMINUS
4.    GOSUB L2
5.    GOSUB ZPLUS
6.    GOSUB LMINUS
7.    RETURN 0
.END

.PROGRAM P6
1.    REMARK ***** P6 COMMAND *****
2.    GOSUB R2
3.    GOSUB YPLUS
4.    GOSUB R2
5.    GOSUB YMINUS
.END

.PROGRAM P8
1.    REMARK ***** P8 COMMAND *****
2.    GOSUB UMINUS
3.    GOSUB F2
4.    GOSUB UPLUS
5.    GOSUB LPLUS
6.    GOSUB UPLUS
7.    GOSUB LPLUS
8.    GOSUB UMINUS
9.    GOSUB LMINUS
10.   GOSUB UMINUS
11.   GOSUB F2
12.   GOSUB UPLUS
13.   GOSUB F2
14.   RETURN 0
.END

```

```

.PROGRAM PS
1.    REMARK ***** PS COMMAND *****
2.    GOSUB UMINUS
3.    GOSUB F2
4.    GOSUB UPLUS
5.    GOSUB FPLUS
6.    GOSUB UMINUS
7.    GOSUB FPLUS
8.    GOSUB UPLUS
9.    GOSUB F2
10.   RETURN 0
.END

.PROGRAM RUP
1.    REMARK *****
2.    REMARK THIS ROUTINE PUTS THE PRESENT RIGHT FACE UP
3.    REMARK *****
4.    SPEED 30.00
5.    GOSUB CLOSE
6.    MOVES ABOVE
7.    MOVES CLEARCAMCW
8.    MOVES NEWCUBECH
9.    OPEN , 0.00
10.   DELAY 0.50
11.   MOVE CLEARCAMCW
12.   GOSUB TURNcubeUP
13.   SETI BUFFER = TOP
14.   SETI BUFFER2 = LT
15.   SETI TOP = RT
16.   SETI RT = POS
17.   SETI POS = BUFFER
18.   SETI LT = FR
19.   SETI FR = BOT
20.   SETI BOT = BUFFER2
21.   RETURN 0
.END

.PROGRAM RMINUS
1.    REMARK ***** RIGHT FACE *****
2.    SETI NEED = RRT
3.    SETI DIR = 2
4.    GOSUB GFT
5.    RETURN 0
.END

.PROGRAM RPLUS
1.    REMARK ***** RIGHT FACE *****
2.    SETI NEED = RRT
3.    SETI DIR = 1
4.    GOSUB GFT
5.    RETURN 0
.END

.PROGRAM R2
1.    REMARK ***** RIGHT FACE *****
2.    SETI NEED = RRT
3.    SETI DIR = 3
4.    GOSUB GFT
5.    RETURN 0
.END

.PROGRAM RPOS
1.    REMARK *****
2.    REMARK MOVES RIGHT FACE TO POSTERIOR POSITION
3.    REMARK *****
4.    GOSUB CLOSE
5.    MOVES ABOVE
6.    MOVES ABOVECCW
7.    MOVES CCW
8.    OPEN , 0.00
9.    DELAY 0.50
10.   MOVES ABOVECCW
11.   MOVES ABOVE
12.   MOVES CUBE
13.   RETURN 0
.END

```



```

.PROGRAM START
1.  REMARK *****
2.  REMARK MOVES TO CUBE POSITION
3.  REMARK *****
4.  MOVE CUBE
5.  RETURN 0
.END

.PROGRAM TURNCUBEUP
1.  REMARK *****
2.  REMARK *** THIS ROUTINE GETS THE CUBE FROM THE NEWBENCH POSITION
3.  REMARK AND PLACES THE INSIDE FACE UP AT THE CUBE POSITION -----
4.  REMARK GRIPPER SHOULD BE AT ANY CLEARCAM POSITION UPON ENTRY *****
5.  REMARK *****
6.  MOVE INSIDES0
7.  MOVES NEWCUBES0
8.  CLOSE , 0.00
9.  DELAY 0.50
10. MOVES INSIDES0
11. MOVE CLEARCAM
12. MOVES ABOVE
13. MOVES CUBE
14. OPEN , 0.00
15. DELAY 0.50
16. RETURN 0
.END

.PROGRAM TYPE
1.  REMARK *****
2.  REMARK TYPES OUT ALL REFERENCED AND ACTUAL FACES (FOR DEBUGGING)
3.  REMARK *****
4.  TYPEI TOP
5.  TYPEI BOT
6.  TYPEI LT
7.  TYPEI RT
8.  TYPEI FR
9.  TYPEI POS
10. TYPEI RTOP
11. TYPEI RBOT
12. TYPEI RLT
13. TYPEI RRT
14. TYPEI RFR
15. TYPEI RPOS
16. RETURN 0
.END

.PROGRAM TEST
1.  GOSUB ORIENT
2.  GOSUB ZMINUS
3.  GOSUB FPLUS
4.  GOSUB ZMINUS
5.  GOSUB FPLUS
6.  GOSUB ZMINUS
7.  GOSUB F2
8.  GOSUB ZPLUS
9.  GOSUB FPLUS
10. GOSUB ZPLUS
11. GOSUB FPLUS
12. GOSUB ZPLUS
13. GOSUB F2
14. RETURN 0
.END

.PROGRAM UPLUS
1.  REMARK ***** UP FACE *****
2.  SETI NEED RTOP
3.  BETI DIR = 1
4.  GOSUB GFT
5.  RETURN 0
.END

```

```

.PROGRAM U2
1.    REMARK ***** UP FACE *****
2.    SETI NEED = RTOP
3.    SETI DIR = 3
4.    GOSUB GFT
5.    RETURN 0
.END

.PROGRAM UMINUS
1.    REMARK ***** UP FACE *****
2.    SETI NEED = RTOP
3.    SETI DIR = 2
4.    GOSUB GFT
5.    RETURN 0
.END

.PROGRAM XMINUS
1.    REMARK *****
2.    REMARK 'X' SLICE NEGATIVE AND REDEFINE REFERENCED FACES
3.    REMARK *****
4.    GOSUB LMINUS
5.    GOSUB RPLUS
6.    SETI BUFFER = RTOP
7.    SETI RTOP = RPOS
8.    SETI RPOS = RBOT
9.    SETI RBOT = RFR
10.   SETI RFR = BUFFER
11.   RETURN 0
.END

.PROGRAM XPLUS
1.    REMARK *****
2.    REMARK 'X' SLICE POSITIVE AND REDEFINE REFERENCED FACES
3.    REMARK *****
4.    GOSUB LPLUS
5.    GOSUB RMINUS
6.    SETI BUFFER = RTOP
7.    SETI RTOP = RFR
8.    SETI RFR = RBOT
9.    SETI RBOT = RPOS
10.   SETI RPOS = BUFFER
11.   RETURN 0
.END

.PROGRAM X2
1.    REMARK *****
2.    REMARK 'X' SLICE TWICE
3.    REMARK *****
4.    GOSUB L2
5.    GOSUB R2
6.    SETI BUFFER = RTOP
7.    SETI BUFFER2 = RFR
8.    SETI RTOP = RBOT
9.    SETI RBOT = BUFFER
10.   SETI RFR = RPOS
11.   SETI RPOS = BUFFER2
12.   RETURN 0
.END

.PROGRAM YMINUS
1.    REMARK *****
2.    REMARK 'Y' SLICE NEGATIVE AND REDEFINE REFERENCED FACES
3.    REMARK *****
4.    GOSUB FPLUS
5.    GOSUB BMINUS
6.    SETI BUFFER = RTOP
7.    SETI RTOP = RRT
8.    SETI RRT = RBOT
9.    SETI RBOT = RLT
10.   SETI RLT = BUFFER
11.   RETURN 0
.END

```

```

.PROGRAM YPLUS
1.  REMARK *****
2.  REMARK 'Y' SLICE POSITIVE AND REDEFINE REFERENCED FACES
3.  REMARK *****
4.  GOSUB FMINUS
5.  GOSUB BPLUS
6.  SETI BUFFER = RTOP
7.  SETI RTOP = RLT
8.  SETI RLT = RBOT
9.  SETI RBOT = RRT
10. SETI RRT = BUFFER
11. RETURN 0
.END

```

```

.PROGRAM YZ
1.  REMARK ***** 'Y' SLICE *****
2.  GOSUB FZ
3.  GOSUB BZ
4.  SETI BUFFER = RTOP
5.  SETI BUFFERZ = RRT
6.  SETI RTOP = RBOT
7.  SETI RBOT = BUFFER
8.  SETI RRT = RLT
9.  SETI RLT = BUFFERZ
10. RETURN 0
.END

```

```

.PROGRAM ZMINUS
1.  REMARK *****
2.  REMARK 'Z' SLICE AND REDEFINE REFERENCED FACES
3.  REMARK *****
4.  GOSUB UMINUS
5.  GOSUB DPLUS
6.  SETI BUFFER = RFR
7.  SETI RFR = RRT
8.  SETI RRT = RPOS
9.  SETI RPOS = RLT
10. SETI RLT = BUFFER
11. RETURN 0
.END

```

```

.PROGRAM ZPLUS
1.  REMARK *****
2.  REMARK 'Z' SLICE POSITIVE AND REDEFINE REFERENCED FACES
3.  REMARK *****
4.  GOSUB UPLUS
5.  GOSUB DMINUS
6.  SETI BUFFER = RFR
7.  SETI RFR = RLT
8.  SETI RLT = RPOS
9.  SETI RPOS = RRT
10. SETI RRT = BUFFER
11. RETURN 0
.END

```

```

.PROGRAM ZZ
1.  REMARK *****
2.  REMARK 'Z' SLICE TWICE AND REDEFINE REFERENCED FACES
3.  REMARK *****
4.  GOSUB UZ
5.  GOSUB DZ
6.  SETI BUFFER = RFR
7.  SETI BUFFERZ = RLT
8.  SETI RFR = RPOS
9.  SETI RPOS = BUFFER
10. SETI RLT = RRT
11. SETI RRT = BUFFERZ
12. RETURN 0
.END

```

A>

ОТВЕТЫ К УПРАЖНЕНИЯМ

1.7.	Нет; 4,4 %	6.15.	188
1.8.	50 %	6.16.	≈50 000 долл. в год;
2.2.	a) — 699		≈219 000 долл. в год (зависит от принятых допущений)
2.4.	Три полюса, две цепи	6.19.	96
2.5.	202,5°	9.2.	В 15 раз (время горения дуги для ручной сварки 30 %)
2.7.	49/499 ≈ 10 %	9.5.	98,4 %; 3936 ч
2.8.	2	9.6.	2400
2.9.	0,998; 0,002	9.7.	2222
2.10.	Сократится с 0,002 (2 на 1000) до 0,8 · 10 ⁻⁸ (8 на миллиард)	9.8.	12 000
2.12.	40	9.9.	11 808
3.4.	Рабочего диапазона нет	9.10.	10,6 мин; 5,66 изделий в час
3.5.	100 %; 100 %; 100 %; 100 %; 71,4 %; 92,9 %	9.11.	10,77 мин; 5,57 изделий в час
3.6.	Да; нет; нет; а; да	10.3.	Истинно; ложно; истинно; истинно; истинно
3.7.	37,5; 79 %	10.5.	Наиболее жесткое ограничение: А; наиболее свободно от ограничений: $A + \bar{A}$
3.8.	38,75 %; 93,94 %	10.9.	$P_1 = A_1 A_2 B_1 B_2$; $P_2 = A_1 B_1 (\bar{A}_2 + \bar{B}_2)$; $P_3 = A_1 B_2 (A_2 \bar{B}_1) +$ $+ A_2 B_1 (A_1 \bar{B}_2)$; $P_4 = A_2 B_2$
3.9.	Нет	13.6.	а) 0,9375; в) 0,46875
3.11.	Эффективность 37 %	13.8.	39; 38; 39; 38
3.12.	а; 31 %; 20 %	13.9.	Интегральное; дифференциальное
3.13.	0,000256; 0,248	13.11.	Пропорциональное; дифференциальное
3.14.	Да	14.2.	Снижение стоимости до 1,875 % относительно первоначальной (снижение стоимости более чем на 98 %)
4.1.	200 изделий в час; 3,6 мин	14.4.	65 536; 36 865; 28 671
4.5.	41,2 изделий в час; 11,44 %	14.6.	FF; 8 бнт; 524 288 бит; 65 536 байт; 131 072 полубайт
4.6.	Новый коэффициент использования = 21,15 % (почти удвоился)	14.7.	8; 1
4.7.	1,0 мин; 8,7 мин	14.9.	7
4.8.	3,33 с на изделие; 19,98 с	14.11.	4 FFF
5.4.	Адрес слова; в приращениях; буква Z	14.12.	CHK; 2480; TEST; 146; CHK; TEST; ROBOT, TOOL
5.5.	Материал толщиной 0,5 дюйма (12,7 мм); 0,1 дюйма (2,54 мм); 2 дюйма (50,8 мм); 1,5 дюйма (38,1 мм)	15.1.	1.456.976
5.9.	3 дюйма (76,2 мм) по вертикали и 2 дюйма (50,8 мм) по горизонтали; из-за радиуса фрезы	15.2.	1.000.000; 60
5.13.	Буква X		
6.3.	2		
6.14.	Четыре или пять рабочих в зависимости от принятых допущений		

1. Advanced Robotics Corporation Form No. 00099-001, 777 Manor Park Drive, Columbus, Ohio 43228.
2. Aidlin, S. S. and Aidlin, S. H. "The Developing Art of Feeding Plastic Bottles." *Automation* (now *Production Engineering*), Vol. 17, No. 1, January 1970.
3. Amstead, B. H.; Ostwald, Philip F.; and Begeman, Myron L. *Manufacturing Processes*, 7th ed. New York: Wiley, 1979.
4. Asfahl, C. Ray. "Curriculum Dilemmas for Robotics Educators," *T-H-E Journal*, Vol. 11, No. 7, April 1984, p. 98.
5. Aumiaux, M. *The Use of Microprocessors*. Translated by Anne Hutt. Chichester: Wiley, 1979.
6. Bailey, J. Ronald. "Product Design for Robotic Assembly." *Proceedings*, 13th International Symposium on Industrial Robots and Robots 7, Vol. 1. Dearborn, Mich.: Society of Manufacturing Engineers, 1983.
7. Baker, James A. "Factory Automation: Industry's Survival Kit." *High Technology*, February 1984.
8. Baker, James A. "Robots: A General Electric Perspective." Paper presented to Society of Manufacturing Engineering executives, Chicago, Ill., September 30, 1981.
9. Boothroyd, G. and Dewhurst, P. *Design for Automatic and Manual Assembly*. Amherst, Mass.: Boothroyd and Dewhurst Manufacturing Consultants, 1983.
10. Boothroyd, G.; Poli, C.; and Murch, L. E. *Automatic Assembly*. New York: Marcel Dekker, 1982.
11. Butow, Steve; Kent, Stan; Major, Janet; and Matthews, Anthony. "The Get Away Special." *Robotics Age*, Vol. 5, No. 5, September–October 1983.
12. Cameron, Arnold C. "Machines for Driving Screws." *Automation* (now *Production Engineering*), Vol. 19, No. 1, January 1972.
13. Cannon, D. L. and Luecke, G. *Understanding Microprocessors*. Dallas: Texas Instruments, 1979.
14. Cooper, K. W., Jr. "Your Computer vs. Rubik's Cube." *Microcomputing*, October 1982.
15. Correale, Anthony. "Physical Design of a Customer 16-Bit Microprocessor." *IBM Journal of Research and Development*. Vol. 26, No. 4, July 1982.
16. Dario, P., et al. "Piezoelectric Polymers: New Sensor Materials for Robotic Applications." *Proceedings*, 13th International Symposium on Industrial Robots and Robots 7, Vol. 2. Dearborn, Mich.: Society of Manufacturing Engineers, 1983.

17. den Hamer, H. E. *Interordering, a New Method of Component Orientation*. New York: Elsevier Scientific Publishing Company, 1980.
18. Dodd, John. "Robots, the New Steel Collar Workers." *The Personnel Journal*, September 1981.
19. Dyer, R. D., et al. "Robot to Solve Rubik's Cube." *Robotics and Industrial Inspection, Proceedings of SPIE*, Vol. 360, August 1982.
20. Emerson, C. Robert. "Cost Justification of Industrial Robots." Paper submitted for publication in *Industrial Engineering*, April 1983.
21. Engelberger, Joseph F. *Robotics in Practice: Management and Applications of Industrial Robots*. New York: AMACOM Division of the American Management Associations, 1980.
22. "Flexible Automation Saves Us Time and Effort." *Modern Material Handling*, Vol. 38, No. 14, October 6, 1983, p. 52.
23. "GE's New FMS—Regaining a Competitive Edge." *Modern Material Handling*, Vol. 38, No. 12, September 6, 1983, p. 48.
24. Graves, Gerald R. *Some Considerations for Group Technology Manufacturing in Production Planning*. Doctoral dissertation, Oklahoma State University, 1984.
25. Groover, Mikell P. *Automation, Production Systems, and Computer-Aided Manufacturing*. Englewood Cliffs, N.J.: Prentice-Hall, 1980.
26. *Introduction to Microelectronics*. Grand Rapids, Mich.: Industrial Training, 1981.
27. Jablonowski, Joseph. "Aiming for Flexibility in Manufacturing Systems." *American Machinist*, Vol. 124, No. 3, March 1980.
28. "Jury Awards Family \$10 Million." *Northwest Arkansas Times*. Fayetteville, Ark (Associated Press article), August 14, 1983.
29. Kinnucan, Paul. "How Smart Robots Are Becoming Smarter." *High Technology*, Vol. 1, No. 1, September–October 1981.
30. Kwei, Edwin C. S. "Evaluation of Process Improvement Alternatives for a Frozen Okra Process." Unpublished masters degree research report, University of Arkansas, Fayetteville, 1971.
31. Lancaster, M. Carol. "Robotics—The New Industrial Revolution." Unpublished student paper, University of Arkansas, Fayetteville, November 1981.
32. Lane, Jack E. and Mariotti, John J. "Workplace Handling System Design Targets on Productivity as Main Objective." *Industrial Engineering*, Vol. 13, No. 4, April 1981.
33. Lapedes, Daniel N., ed. *Dictionary of Scientific and Technical Terms*, 2nd ed. New York: McGraw-Hill, 1978.
34. MacGregor, Douglas. *The Human Side of Enterprise*. New York: McGraw-Hill, 1960.

35. McManus, G. J. "Labor Eases Automation Stance." *Iron Age*, March 1966.
36. McWhorter, Gene. *Understanding Digital Electronics*. Dallas: Texas Instruments Learning Center, 1978.
37. *Machine Guarding—Assessment of Need* (NIOSH 75-173), U.S. Department of Health and Human Services (formerly HEW), June 1975.
38. *Making Our Robots and Your Process Work as One*. Columbus, Ohio: Advanced Robotics Corporation, 1983.
39. Marbach, William D.; Rogers, Michael; and Conant, Jennet. "Working at the Wafer's Edge." *Newsweek*, August 22, 1983.
40. Miller, Irwin and Freund, John E. *Probability and Statistics for Engineers*, 2nd ed. Englewood Cliffs, N.J.: Prentice-Hall, 1977.
41. Miller, Richard K. *Robots in Industry: Applications for Metal Fabrication*. Madison, Ga.: SEAI Institute, 1982.
42. *Minimover-5 User Reference and Applications Manual*. Mountain View, Calif.: Microbot, 1980.
43. Mitrofanov, S. P. *Scientific Principles of Group Technology*, (English translation from Russian). National Lending Library for Science and Technology, 1966. (Original text published in 1959.)
44. *Model 510 Programmable Controller User's Manual*. Johnson City, Tenn.: Texas Instruments, February 1981.
45. Moder, Joseph J.; Phillips, Cecil R., and Davis, Edward W. *Project Management with CPM and PERT*, 3rd ed. New York: Van Nostrand Reinhold, 1984.
46. Molander, Thony. "Routing and Drilling with an Industrial Robot." *Proceedings*, 13th International Symposium on Industrial Robots and Robots 7, Vol. 1. Dearborn, Mich.: Society of Manufacturing Engineers, 1983.
47. *MottonMate Robot System Installation and Operation Manual*. Akron, Ohio: Schrader-Bellows Division, Scovill, December 1981.
48. Mutter, Randolph F. "Effective Interfacing Through End Effectors." *Proceedings*, 13th International Symposium on Industrial Robots and Robots 7, Vol. 1. Dearborn, Mich.: Society of Manufacturing Engineers, 1983.
49. Naisbitt, John. *Megatrends*. New York: Warner Books, 1982.
50. Ottinger, Lester V. "Robotics for the I.E.: The Automated Factory." *Industrial Engineering*, Vol. 14, No. 9, September 1982.
51. Prenting, Theodore. "Why Automatic Assembly Needs IE's." *Automation* (now *Production Engineering*), Vol. 15, No. 12, December 1970.
52. Pressman, Roger S. and Williams, John E. *Numerical Control and Computer-Aided Manufacturing*. New York: Wiley, 1977.
53. Quinlan, Joseph C. "Welding with Robots—Is It for You?" *Tooling and Production*, March 1983.

54. Remick, Carl. "Robots: New Faces on the Production Line." *Management Review*, American Management Association, May 1979.
55. *Results with Robots* Product literature, Prab Robots, 1982.
56. Rodgers, Robert C. "What's News in PCs?" *Production Engineering*, Vol. 28, No. 11, November 1981.
57. Rogers, P. F. and Boothroyd, G. "Designing Slot Orienting Devices for Vibratory Bowl Feeders." *Automation* (now *Production Engineering*), Vol. 19, No. 1, January 1972.
58. Rosato, Pat John. "Robotic Implementation—Do It Right." *Proceedings*, 13th International Symposium on Industrial Robots and Robots 7. Dearborn, Mich.: Society of Manufacturing Engineers, April 1983.
59. "Rubik's Robot." *Science* 82, Vol. 3, No. 6, July–August 1982.
60. Russell, Marvin, Jr. "Odex I: The First Functionoid." *Robotics Age*, Vol. 5, No. 5, September–October 1983.
61. Russell, R. A. "Closing the Sensor-Robot Control Loop." *Robotics Age*, Vol. 6, No. 4, April 1984.
62. Sadowski, Randall P. "History of Computer Use in Manufacturing Shows Major Need Now Is for Integration." *Industrial Engineering*, Vol. 16, No. 3, March 1984, p. 34.
63. Schmitt, Neil M. and Farwell, Robert F. *Understanding Electronic Control of Automation Systems*. Dallas: Texas Instruments Learning Center, 1983.
64. *SMART™ I Programmer's Manual*. Pittsburgh: Aerotech, February 1982.
65. Smith, Randall C. and Nitzan, David. "A Modular Programmable Assembly Station." *Proceedings*, 13th International Symposium on Industrial Robots and Robots 7. Dearborn, Mich.: Society of Manufacturing Engineers, 1983.
66. Standard EIA RS-267-B, Electronic Industries Association. Standard available in its entirety from Electronic Industries Association, 2001 Eye Street NW, Washington, D.C. 20006, (202) 457-4900.
67. Stauffer, Robert N. "An Exercise in Robot Press Loading." *Manufacturing Engineering*, Vol. 86, No. 2, February 1981.
68. Susnjara, Ken. *A Manager's Guide to Industrial Robots*. Englewood Cliffs, N.J.: Prentice-Hall, 1982.
69. Tanner, William R. "Industrial Robots Today." *Machtime and Tool Bluebook*, March 1980.
70. Tanner, William R., ed. *Industrial Robots*, Vol. 1 and 2. Dearborn, Mich.: Society of Manufacturing Engineers, 1979.
71. Ullrich, Robert A. *The Robotics Primer*. Englewood Cliffs, N.J.: Prentice-Hall, 1983.
72. *Unimate Industrial Robot System Planbook*. Product literature. Unimation-Westinghouse, (no date).

73. *Unimate Puma™ Mark II Robot-500 Series, Vol. II—User's Guide to VAL™* Danbury, Conn.: Unimation-Westinghouse, 1983.
74. U.S. Steel Corporation. Annual Report, 1983.
75. Vanderbrug, Gordon; Wilt, Donald; and Davis, Jim. "Robotic Assembly of Keycaps to Keyboard Arrays." *Proceedings, 13th International Symposium on Industrial Robots and Robots 7*. Dearborn, Mich.: Society of Manufacturing Engineers, 1983.
76. Waldman, Harry. "The Programmable Controller: Continuing to Grow in Use." *Manufacturing Engineering*, Vol. 91, No. 2, August 1983.
77. Warnecke, H. J., et al. "Application of Industrial Robots for Assembly Operations in the Automotive Industry." *Proceedings, 13th International Symposium on Industrial Robots and Robots 7*. Dearborn, Mich.: Society of Manufacturing Engineers, 1983.
78. *Western Electric Engineer*, Vol. 27, No. 2, 1983.

Предисловие	5
1. Подготовка к автоматизации	8
1.1. Влияние конкуренции	8
1.2. Конструирование для автоматизации	10
1.3. Стабильность процессов	17
1.4. Обеспечение надежности оборудования	19
1.5. Серийность производства	20
Выводы	23
Упражнения и вопросы	24
2. Элементы автоматизации	26
2.1. Датчики	27
2.2. Анализаторы	38
2.3. Исполнительные устройства	45
2.4. Приводы	50
Выводы	54
Упражнения и вопросы	55
3. Механизация манипулирования деталями	57
3.1. Подача изделий	57
3.2. Исследование процесса подачи деталей	66
Выводы	78
Упражнения и вопросы	78
4. Автоматическое производство и автоматическая сборка	81
4.1. Сборочные установки	81
4.2. Производительность и время производства	84
4.3. Буферный склад	89
Выводы	92
Упражнения и вопросы	93
5. Числовое программное управление, автоматизированное проектирование и изготовление с помощью ЭВМ	94
5.1. Преимущества числового программного управления	98
5.2. Машинные коды	100
5.3. Оси координат	107
5.4. Программирование в приращениях и в абсолютной системе координат	107
5.5. Контроль перемещений	110
5.6. Программирование для ЧПУ	110
5.7. Прямое числовое программное управление	117
5.8. Числовое программное управление от ЭВМ	119
5.9. Иерархическое программное управление	121
5.10. Комплексная система автоматизированного проектирования и изготовления с помощью ЭВМ	122
Выводы	123
Упражнения и вопросы	127
6. Промышленные роботы	127
6.1. Геометрия робота	129
6.2. Приводы роботов	137

6.3. Контроль перемещений	143
6.4. Оснащение робота	145
6.5. Программирование	148
6.6. Возможности оцувствления	151
6.7. Технические характеристики	155
6.8. Области применения роботов и обоснование экономической це- лесообразности их использования	157
Выводы	161
Упражнения и вопросы	162
7. Обучение робота выполнению работ	164
7.1. Мир роботов и мир реальности	164
7.2. Методы программирования	167
7.3. Язык ВАЛ	173
7.4. Язык АРМБейсик	186
Выводы	192
Упражнения и вопросы	193
8. Введение роботов	195
8.1. Оценка проекта роботизации	196
8.2. Планирование	197
8.3. Дооснащение	205
8.4. Моделирование и испытание	208
8.5. Установка	214
8.6. Отиошение руководства и рабочих	217
8.7. Техническое обучение	218
Выводы	219
Упражнения и вопросы	220
9. Использование роботов в промышленности	222
9.1. Сварка	222
9.2. Загрузка технилогического оборудования	224
9.3. Окраска распылением	240
9.4. Сборка	245
Выводы	251
Упражнения и вопросы	252
10. Промышленные логические системы управления	254
10.1. Таблицы истинности	256
10.2. Булева алгебра	259
Выводы	267
Упражнения и вопросы	268
11. Графическая интерпретация логических функций	270
11.1. Логические схемы	270
11.2. Лестничные логические схемы	273
11.3. Таймеры	277
11.4. Временные диаграммы	278
Выводы	289
Упражнения и вопросы	289
12. Программируемые контроллеры	291
12.1. Что представляет собой ПК?	292
12.2. Специальные устройства ПК	296
12.3. Программирование ПК	303
12.4. Программы пользователя	312
12.5. Управление промышленными роботами	314
12.6. Преимущества ПК	321
12.7. Недостатки ПК	326
Выводы	327
Упражнения и вопросы	328
13. ЭВМ для управления в реальном времени	330
13.1. Уровни реализации	332
13.2. Концепции управления	339

13.3. Интерфейс производственного оборудования	345
13.4. Прерывания	352
13.5. Программирование управляющей ЭВМ	357
Выводы	358
Упражнения и вопросы	359
14. Микропроцессоры	361
14.1. Преимущества микропроцессоров	362
14.2. Микропроцессоры и микроЭВМ	363
14.3. Интегральные схемы	365
14.4. Структура микропроцессора	369
14.5. Программирование	375
14.6. Применение микропроцессоров	380
Выводы	383
Упражнения и вопросы	384
15. Перспективы и этические проблемы автоматизации.	387
15.1. Гибкие производственные системы (ГПС)	387
15.2. Групповая технология	394
15.3. Производство с комплексным управлением от ЭВМ.	398
15.4. Самовоспроизводящиеся системы	399
15.5. Датчики типа кошачьих усов	403
15.6. Этические проблемы автоматизации	405
Выводы	410
Упражнения и вопросы	411
Приложение А	412
Приложение В	416
Приложение С	418
Ответы к упражнениям	439
Список литературы	440