

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**



ПОБЕДИТЕЛЬ КОНКУРСА ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВУЗОВ

В.А. Валетов, А.А. Орлова, С.Д. Третьяков
**ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ
ИЗГОТОВЛЕНИЯ ПРИБОРОВ И УСТРОЙСТВ**

Учебное пособие



Санкт-Петербург

2008

Валетов В.А., Орлова А.А., Третьяков С.Д. Интеллектуальные технологии производства приборов и систем. Учебное пособие, - СПб: СПб ГУИТМО, 2008. – 134 с.

Рассматриваются вопросы применения современных интеллектуальных технологий в приборостроении. Излагаются теоретические предпосылки использования элементов искусственного интеллекта для управления интеллектуальными производственными комплексами. Рассматриваются вопросы использования современных робототехнических производственных модулей и интеллектуальных систем управления ими. Приводятся примеры создания структур управления интеллектуальными производственными комплексами с использованием элементов искусственного интеллекта.

Пособие предназначено для студентов старших курсов и магистров, изучающих дисциплину «Интеллектуальные технологии производства приборов и систем» на кафедре Технологии приборостроения СПб ГУИТМО.

Рекомендовано УМО по образованию в области приборостроения и оптотехники в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки 200100 – Приборостроение.



В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

©Санкт-Петербургский государственный университет информационных технологий, механики и оптики, 2008

© В.А. Валетов, С.Д. Третьяков, А.А. Орлова 2008

ВВЕДЕНИЕ	5
1. Современные интеллектуальные технологии, применяемые при производстве приборов и систем	6
1.1 Основные этапы развития интеллектуальных систем и предпосылки их использования в промышленности.....	6
1.2 Признаки интеллекта производственных систем	8
1.3 Современные тенденции развития интеллектуальных производственных систем	13
1.4. Робототизированные системы искусственного интеллекта	18
2. Особенности использования интеллектуальных систем и систем искусственного интеллекта в приборостроении	21
2.1 Элементы искусственного интеллекта, применяемые в приборостроении	21
2.2 Особенности представления знаний в интеллектуальных производственных системах.....	24
2.3 Языки и модели представления знаний	28
2.4. Семантические сети	36
3. Методы поиска решений технологических задач и системы представления знаний	49
3.1 Фреймы.....	49
3.2 Семантические и когнитивные сети	53
3.3 Продукционные правила или продукционные системы	60
3.4 Методы поиска решений технологических задач	64
3.4.1 Структуры данных для поиска в пространстве состояний	64
3.4.2 Представление задачи в пространстве состояний.....	65
3.4.3 Поиск на основе данных и от цели	66
3.5 Поиск в глубину и в ширину	68
3.5.1. Поиск в ширину	68
3.5.2 Поиск в глубину.....	70
3.5.3 Поиск в глубину с итерационным заглублением	71
3.6 Процедура поиска с возвратами. BACKTRACK.....	72
3.7 Алгоритмы эвристического поиска.....	75
3.7.1 Алгоритм наискорейшего спуска для дерева решений	75
3.7.2. Алгоритмы оценочных (штрафных) функций.....	78
3.7.3 Алгоритм минимакса	80
3.7.4 Альфа-бета-процедура	80
3.8 Поиск решений на основе исчисления предикатов	83
4. Использование экспертных систем для решения технологических задач производства деталей приборов	86
4.1 Экспертные системы: определение и классификация.....	86
4.2 Технологические задачи, решаемые при помощи экспертных систем	88
4.3 Этапы проектирования экспертных систем.....	89
4.4. Инструментальные средства создания экспертных систем	91
4.5. Методика создания экспертных систем	92
4.6. Примеры экспертных систем	94

5. Использование робототехнических систем с элементами искусственного интеллекта в приборостроении	98
5.1. Область робототехники	98
5.2. Системы искусственного интеллекта в робототехнике	100
5.3. Современное робототехническое технологическое оборудование с элементами искусственного интеллекта	101
5.4. Структура и состав интеллектуальной робототехнической системы.....	103
5.5. Обобщённый пример устройства робота-станка	106
6. Применение в приборостроении современного технологического оборудования с элементами искусственного интеллекта и интеллектуальных систем	107
6.1. Особенности современного технологического оборудования	107
6.2. Особенности разработки технологических процессов для интеллектуальных робототехнических систем	110
7. Управление производственным робототехническим комплексом с использованием элементов интеллектуального управления	113
7.1 Основные функции и особенности интеллектуальных систем управления	113
7.2. Состав системы управления и функциональные характеристики ее элементов.....	114
7.3. Пример расчёта параметров встроенной оптической системы контроля на примере абразивной обработки поверхности.....	116
8. Воспроизведение сложных поверхностей деталей приборов при помощи систем искусственного интеллекта	120
9. Контроль работы производственного комплекса с использованием интеллектуальной системы управления качеством изделия	123
10. Использование встраиваемых технологических систем контроля работы интеллектуальных производственных систем	126
11. Моделирование технологического комплекса позиционно-силового управления в системе Simulink	128
СПИСОК ЛИТЕРАТУРЫ	131

ВВЕДЕНИЕ

Одним из основных факторов, оказывающих сильное влияние на социально-экономическое развитие общества, является уровень развития промышленного производства. В странах с развитой экономической системой уровень производственной культуры обеспечивается за счёт не прекращающихся исследований в области новых технологических разработок. Одной из основных тенденций последних десятилетий является стремление автоматизировать как можно больше производственных процессов в различных отраслях промышленного производства. Последние достижения науки в области развития систем искусственного интеллекта позволяют вывести процессы автоматизации производства на качественно новый уровень. Наряду с уже используемыми системами роботизированного производства, внедрение искусственного интеллекта в управление производственными процессами значительно расширяет возможности технологических систем. Такой симбиоз автоматизированных производственных систем и элементов искусственного интеллекта получил название интеллектуальных технологий. Область применения таких технологий довольно широка. Производственные комплексы с элементами интеллектуальных технологий можно встретить в различных отраслях промышленности, но очевидно, что наибольшее распространение такие комплексы получили в сфере производства высоко точных приборов и систем.

В данном учебном пособии рассматриваются вопросы применения интеллектуальных технологий в области приборостроения. Описываются основы систем искусственного интеллекта, алгоритмы их работы в комплексе с современным высокоточным оборудованием. Рассматриваются вопросы моделирования работы интеллектуальных систем, а также возможности интеллектуальных систем управления.

1. Современные интеллектуальные технологии, применяемые при производстве приборов и систем

1.1 Основные этапы развития интеллектуальных систем и предпосылки их использования в промышленности

В последние десятилетия основная тенденция в развитии промышленного производства заключалась в разработке систем комплексной автоматизации. Здесь стоит отметить, что под комплексной автоматизацией понимается постепенный переход к интегрированным научно-производственным комплексам, базирующимся на широком применении гибких средств автоматизации и вычислительной техники на протяжении всего производственного цикла - от научных исследований до выпуска готовой продукции. При этом автоматизация собственно производства заключается в создании робототехнологических комплексов и на их основе - гибких автоматических производств. В рамках гибкого автоматизированного производства все робототехнические комплексы и обслуживающие их системы управляются от сети ЭВМ. Это придает им необходимую гибкость по отношению к возможным изменениям номенклатуры или типоразмеров выпускаемой продукции. В свою очередь, бурное развитие в последние годы информационных технологий позволило осуществить более тесную интеграцию систем автоматизированного производства, робототехнических комплексов и электронно-вычислительных систем. В первую очередь это связано с бурным развитием систем искусственного интеллекта. Включение в производственную цепочку элементов искусственного интеллекта (ИИ) позволяет добиться максимальной автономности производственного комплекса и ввести в автоматизированную систему новый уровень управления. Здесь стоит более подробно рассказать об основах систем искусственного интеллекта и их возможностях, особенно в сфере промышленного производства.

В 1950 году британский математик Алан Тьюринг опубликовал в журнале «Mind» свою работу «Вычислительная машина и интеллект», в которой описал тест для проверки программы на интеллектуальность. Он предложил поместить исследователя и программу в разные комнаты и до тех пор, пока исследователь не определит, кто за стеной - человек или программа, считать поведение программы разумным. Это было одно из первых определений интеллектуальности, то есть А. Тьюринг предложил называть интеллектуальным такое поведение программы, которое будет моделировать разумное поведение человека. В настоящее время существует множество определений интеллектуальных систем и искусственного интеллекта. Сам термин «искусственный интеллект» (AI - Artificial Intelligence) был предложен в 1956 году на семинаре в Дартмутском колледже (США). Ниже приведены некоторые из этих определений. Так Д. Люгер в своей книге [11] определяет искусственный интеллект как область компьютерных наук, занимающуюся

исследованием и автоматизацией разумного поведения».

В учебнике по искусственному интеллекту [12] дается такое определение: «Искусственный интеллект - это одно из направлений информатики, целью которого является разработка аппаратно-программных средств, позволяющих пользователю-непрограммисту ставить и решать свои, традиционно считающиеся интеллектуальными задачи, общаясь с ЭВМ на ограниченном подмножестве естественного языка».

В любом случае предметом ИИ является изучение интеллектуальной деятельности человека, подчиняющейся заранее неизвестным законам. ИИ это все то, что не может быть обработано с помощью алгоритмических методов. Таким образом в цепочку - система автоматизированного производства, робототехнический комплекс и электронно-вычислительная система – добавляется новое звено – система искусственного интеллекта. Но в отличие от традиционного метода управления автоматизированным производством посредством электронно-вычислительного комплекса, который обрабатывает информацию по известным заранее законам и алгоритмам, ввод в эту цепочку системы искусственного интеллекта есть не что иное, как попытка сделать автоматизированную производственную систему как можно более автономной и адаптируемой. При этом под системой здесь понимается множество элементов, находящихся в отношениях друг с другом и образующих причинно-следственную связь, а под адаптивной системой такая система, которая сохраняет работоспособность при непредвиденных изменениях свойств управляемого объекта, целей управления или окружающей среды путем смены алгоритма функционирования, программы поведения или поиска оптимальных, в некоторых случаях просто эффективных, решений и состояний. Традиционно, по способу адаптации различают самонастраивающиеся, самообучающиеся и самоорганизующиеся системы.

Таким образом, под интеллектуальной системой понимается адаптивная система, позволяющая выполнять программы целесообразной деятельности по решению поставленных перед ней задач на основании конкретной ситуации, складывающейся на данный момент в окружающей их среде. Очевидно, что применение таких систем при автоматизированном производстве приборов и устройств наиболее целесообразно как с технологической точки зрения, так и с экономической. К области решаемых интеллектуальными системами задач относятся задачи, обладающие, как правило, следующими особенностями.

1. В этих задачах неизвестен алгоритм их решения (такие задачи называют интеллектуальными);

2. В этих задачах используется, помимо традиционных данных в числовом формате, информация в виде изображений, рисунков, знаков, букв, слов, звуков;

3. В таких задачах предполагается наличие выбора.

Так как для решения интеллектуальных задач не существует алгоритма, то предполагается сделать выбор между многими вариантами в условиях неопределённости. Таким образом, очевидно, что свобода выбора является существенным признаком интеллектуальных задач.

Интеллектуальные технологические системы производства содержат переменную, настраиваемую модель внешнего мира и реальной исполнительской системы с объектом управления. Цель и управляющие воздействия формируются в таких системах на основе знаний о внешней среде, объекте управления и на основе моделирования ситуаций в реальной системе.

1.2 Признаки интеллекта производственных систем

Когда идёт речь об интеллектуальной системе, то в первую очередь задаётся вопрос, о том, какие признаки интеллекта должны характеризовать такую систему. Очевидно, что интеллектуальная система должна уметь в наборе фактов распознать наиболее существенные, а также быть способной из имеющихся фактов и знаний сделать выводы не только с использованием дедукции, но и с помощью аналогии, индукции и т.д. Кроме того, интеллектуальная система должна обладать умением давать самооценку, т.е. обладать рефлексией – средствами для оценки результатов собственной работы. Интеллектуальная система, при помощи неких подсистем, должна также уметь отвечать на вопросы о том, каким образом получен тот или иной результат. Обязательным условием для функционирования интеллектуальной системы является её способность обобщать данные, обнаруживать сходство между имеющимися в её распоряжении фактами.

Например, производственный модуль с использованием электронного управления, который при изготовлении детали допускает неточность одного и того же параметра до тех пор, пока не вмешается оператор, интеллектуальной системой называться не может.

Центральным понятием в интеллектуальной системе является знание. Существует множество определений знания. Например, знание есть результат, полученный познанием окружающего мира и его объектов или система суждений с принципиальной и единой организацией, основанная на объективной закономерности. Для области приборостроения самым точным является понятие знания как совокупности фактов и правил. Фрагмент знания представляет некое правило, которое в общем случае имеет вид *если <условие> то <действие>*.

На рис. 1.1 показан процесс логического вывода формализованной информации, которую используют в процессе вывода. Также важным представляется деление знаний на статические и динамические. Под статическими обычно понимают такие знания, которые были внедрены в интеллектуальную систему на этапе её проектирования. Соответственно, динамические знания интеллектуальная система получает в процессе функционирования в масштабе реального времени.

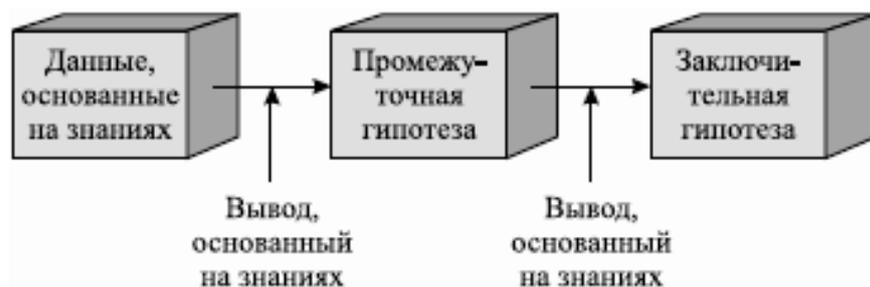


Рис. 1.1 Процесс логического вывода в интеллектуальной системе

В свою очередь, знания можно разделить на факты и правила. Под фактами подразумеваются знания типа «А это А», они характерны для баз данных. Под правилами (их ещё называют продукциями) понимаются знания вида «ЕСЛИ-ТО». Кроме этих знаний существуют так называемые метазнания, т.е. знания о знаниях. Создание продукционных систем для представления знаний позволило разделить знания и управление в компьютерной программе, обеспечить модульность продукционных правил, т.е. отсутствие синтаксического взаимодействия между правилами. При создании моделей представления знаний следует учитывать такие факторы, как однородность представления и простота понимания. Выполнить это требование в равной степени для простых и сложных задач довольно сложно.

Представленная на рис. 1.2 простейшая структурная схема управления интеллектуальным робототехническим модулем позволяет более наглядно усвоить данный материал.

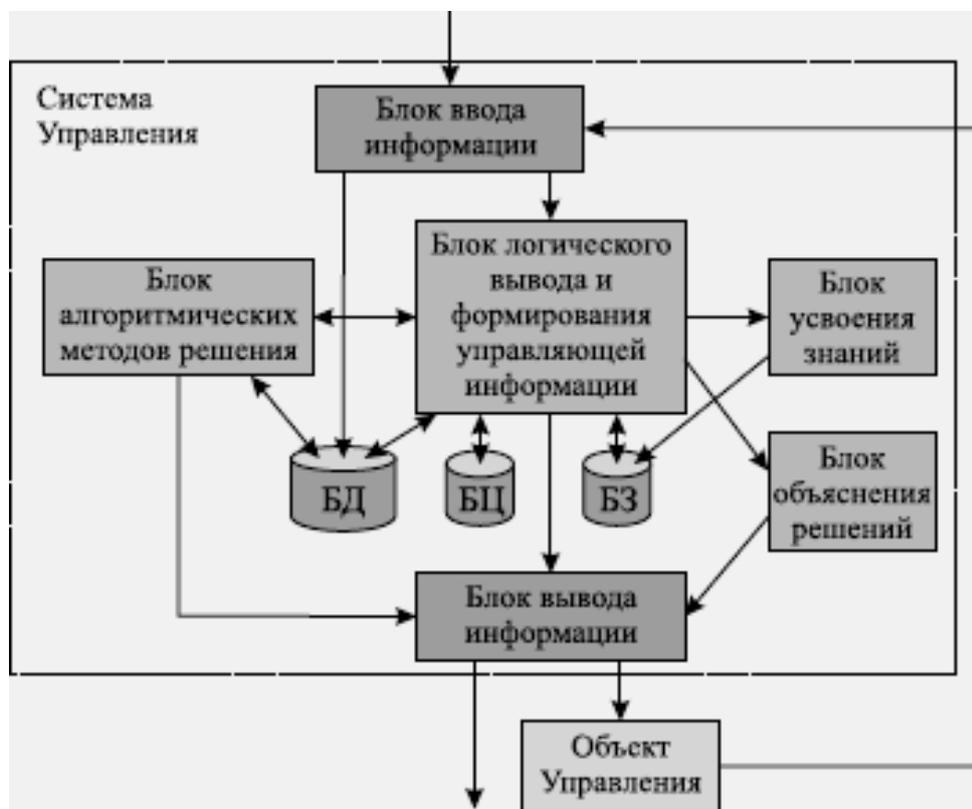


Рис. 1.2 Структурная схема управления интеллектуальным робототехническим модулем, где БД – база данных, БЦ – блок целей и БЗ – база знаний

На этом рисунке стрелками обозначено направление движения информации, двунаправленными стрелками обозначено взаимодействие типа «запрос-ответ» и «действие-подтверждение», весьма распространенное в информационных системах. Входом системы является «Блок ввода информации», предназначенный для ввода числовых данных, текста, речи, распознавания изображений. Информация на вход системы может поступать (в зависимости от решаемой задачи) от пользователя, внешней среды, объекта управления. Далее входная информация поступает в блок логического вывода, либо сразу в базу данных - совокупность таблиц, хранящих, как правило, символьную и числовую информацию об объектах предметной области. Блок логического вывода и формирования управляющей информации обеспечивает нахождение решений для нечетко формализованных задач интеллектуальной системы, осуществляет планирование действий и формирование управляющей информации для пользователя или объекта управления на основе базы знаний, базы данных, базы целей и блока алгоритмических методов решений. Под базой знаний обычно понимается совокупность знаний, например, система продукционных правил, о закономерностях предметной области, например информация о режимах резания при механической обработке заготовок. Блок целей - это множество локальных целей системы, представляющих собой совокупность знаний, активизированных в конкретный момент и в конкретной ситуации для достижения глобальной цели. Блок алгоритмических методов решений содержит программные модули решения задач предметной области по жестким алгоритмам. Блок усвоения знаний осуществляет анализ динамических знаний с целью их усвоения и сохранения в базе знаний. Блок объяснения решений интерпретирует пользователю последовательность логического вывода, примененную для достижения текущего результата. На выходе системы блок вывода информации обеспечивает вывод данных, текста, речи, изображений и другие результаты логического вывода пользователю или объекту управления. Контур обратной связи позволяет реализовать свойства адаптивности и обучения интеллектуальной системы. На этапе проектирования эксперты и инженеры по знаниям наполняют базу знаний и базу целей, а программисты разрабатывают программы алгоритмических методов решений. База данных создается и пополняется, как правило, в процессе эксплуатации интеллектуальной системы.

Динамика работы описанной выше системы может быть описана следующим образом. При поступлении информации на внешнем языке системы на входе блока ввода информации производится ее интерпретация во внутреннее представление для работы с символьной моделью системы. Блок логического вывода выбирает из базы знаний множество правил, активизированных поступившей входной информацией, и помещает эти правила в банк целей как текущие цели системы. Далее блок логического вывода по заданной стратегии, например, стратегии максимальной достоверности, выбирает правило из базы целей и пытается доопределить переменные модели внешнего мира и исполнительской системы с объектом управления. На основе этого активизируются новые правила базы знаний и

начинается логический вывод в системе продукций (правил). Эта процедура заканчивается, как только решение будет найдено, либо когда будет исчерпан банк целей. Найденное решение из внутреннего представления интерпретируется блоком вывода информации во внешний язык подсистемы управления низшего уровня и объекта управления. Более подробно работа этих подсистем будет рассмотрена в последующих главах.

Работы по созданию промышленных роботизированных систем ведутся уже не одно десятилетие, но именно бурное развитие исследований в области искусственного интеллекта позволило этой отрасли промышленности бурно развиваться в последнее время. Однако, прежде чем инженерам удалось соединить различные по своей сути системы в единое целое, учёным предстояло решить множество проблем, лежащих как в области робототехники, так и в сфере искусственного интеллекта. Интеллектуальные роботы (иногда говорят «интеллектные» или роботы с искусственным интеллектом) явились развитием простейших программируемых промышленных роботов, которые появились в 60-х годах. Тогда же были заложены основы современных и будущих интеллектуальных роботов в исследованиях, связанных с координацией программирования роботов-манипуляторов и технического зрения на основе телевизионной камеры, планирования поведения мобильных роботов, общения с роботом на естественном языке.

Эксперименты с первыми интеллектуальными роботами проводились в конце 60-х - начале 70-х годов в Стэнфордском университете, Стэнфордском исследовательском институте (Калифорния), Массачусетском технологическом институте (Массачусетс), Эдинбургском университете (Великобритания), в Электротехнической лаборатории (Япония) [21].

Типичный интеллектуальный робот состоит из одной или двух рук (манипуляторов) и одной или двух телевизионных камер, размещенных на неподвижной тумбе либо на перемещающейся тележке. На рис. 1.3 показана обобщенная структура информационной системы интеллектуального робота.

Надо иметь в виду, что на подсистему восприятия поступает большой объём разнотипной информации от датчиков различных типов: зрительных, слуховых, тактильных, температурных, лазерных или ультразвуковых дальномеров и т.д. При этом под синтаксисом, в данной схеме, понимается структура в пространстве и во времени этой разнотипной информации, а под семантикой - результат ее восприятия как множества возможных типовых ситуаций или образов, требующих какой-либо дальнейшей обработки. Под миром понимается описание окружения робота как результат работы его подсистемы восприятия. Под действием понимается достаточно сложный двигательный акт, например, перемещение заготовки из входного бункера в шпиндель станка и ее закрепление там, в отличие от движения как результата срабатывания какой-либо одной степени свободы робота, например, вращение робота вокруг вертикальной осп на заданный угол.

Таким образом, можно отметить тот факт, что рождение робототехники выдвинуло задачи машинного зрения и распознавания изображений в число первоочередных. В традиционном распознавании образов появился хорошо

разработанный математический аппарат, и для не очень сложных объектов оказалось возможным строить практически работающие системы классификации по признакам, по аналогии и т. д.

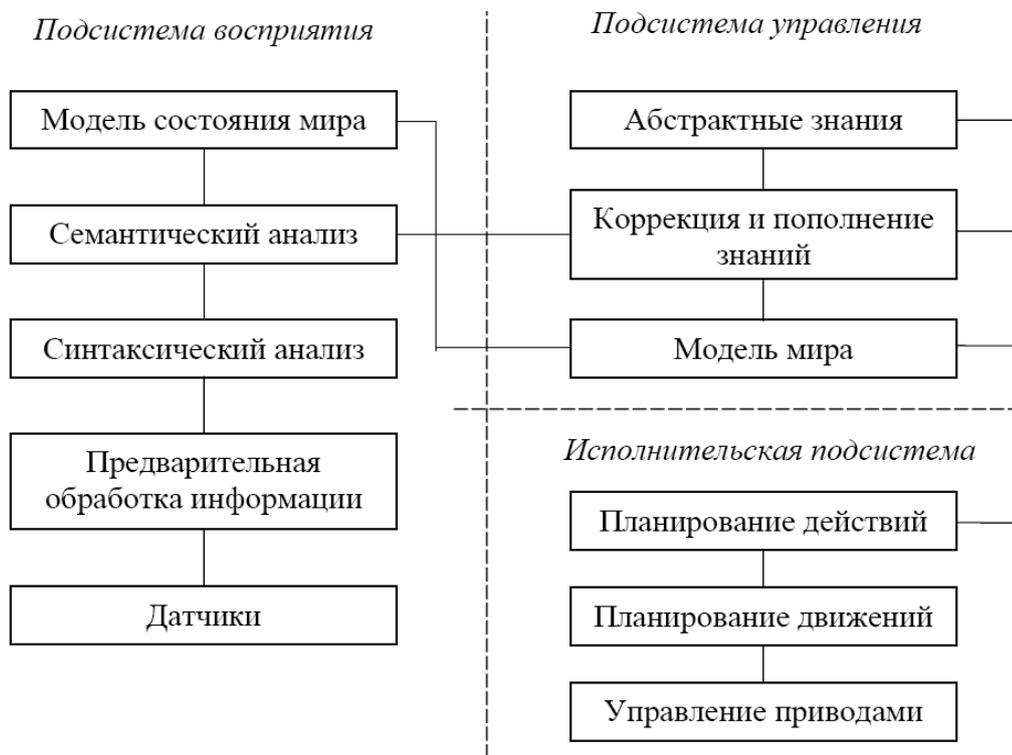


Рис. 1.3 Структура информационной системы интеллектуального робота

В качестве признаков могут рассматриваться любые характеристики распознаваемых объектов. Признаки должны быть инвариантны к ориентации, размеру и вариациям формы объектов. Алфавит признаков придумывается разработчиком системы. Качество распознавания во многом зависит от того, насколько удачно придуман алфавит признаков. Распознавание состоит в априорном получении вектора признаков для выделенного на изображении отдельного распознаваемого объекта, и лишь затем в определении того, какому из эталонов этот вектор соответствует. П. Уинстон в начале 80-х годов обратил внимание на необходимость реализации целенаправленного процесса машинного восприятия. Цель должна управлять работой всех процедур, в том числе и процедур нижнего уровня, т. е. процедур предварительной обработки и выделения признаков. Должна иметься возможность на любой стадии процесса, в зависимости от получаемого результата, возвращаться к его началу для уточнения результатов работы процедур предшествующих уровней. У П. Уинстона, так же как и у других исследователей, до решения практических задач дело не дошло, хотя в 80-е годы вычислительные мощности больших машин позволяли начать решение подобных задач. Таким образом, ранние традиционные системы распознавания, основывающиеся на последовательной организации процесса распознавания и классификации объектов, эффективно решать задачи восприятия сложной зрительной информации не могли.

В начале 70-х годов произошел качественный скачок и пришло понимание, что необходимы глубокие знания в соответствующей области и выделение знаний из данных, получаемых от эксперта. Появляются экспертные системы (ЭС), или системы, основанные на знаниях, без которых не возможно представить себе ни одну современную интеллектуальную технологическую систему.

1.3 Современные тенденции развития интеллектуальных производственных систем

Также стоит отметить тот факт, что одной из основных тенденций развития промышленных интеллектуальных систем, в настоящее время, является явное смещение исследований в область нейрокибернетики, или иначе говоря, подход к разработке систем, демонстрирующих «разумное» поведение, на основе архитектур, напоминающих устройство мозга и называемых нейронными сетями [8]. В 1942 году, когда Н. Винер определил концепции кибернетики, В. Мак-Каллок и В. Питс опубликовали первый фундаментальный труд по нейронным сетям, где говорилось о том, что любое хорошо заданное отношение вход-выход может быть представлено в виде формальной нейронной сети (рис. 1.4). Одна из ключевых особенностей нейронных сетей состоит в том, что они способны обучаться на основе опыта, полученного в обучающей среде. В 1957 году Ф. Розенблат изобрел устройство для распознавания на основе нейронной сети - перцептрон, который успешно различал буквы алфавита, хотя и отличался высокой чувствительностью к их написанию [1]. Пик интереса к нейронным сетям приходится на 60-е и 70-е годы, а в последние десять лет наблюдается резко возросший объем исследований и разработок нейронных сетей. Это стало возможным в связи с появлением нового аппаратного обеспечения, повысившего производительность вычислений в нейронных сетях (нейропроцессоры, транспьютеры и т. п.). Нейронные сети хорошо подходят для распознавания образов и решения задач классификации, оптимизации и прогнозирования, поэтому основными областями их применения являются промышленное производство и робототехника.

Нейронная сеть (НС) считается альтернативным подходом к построению систем искусственного интеллекта, который использует модель, имитирующую структуры нейронов в человеческом мозге. Другим альтернативным подходом считаются генетические алгоритмы (ГА), в основе которых лежат имитация эволюции различных альтернативных конфигураций.

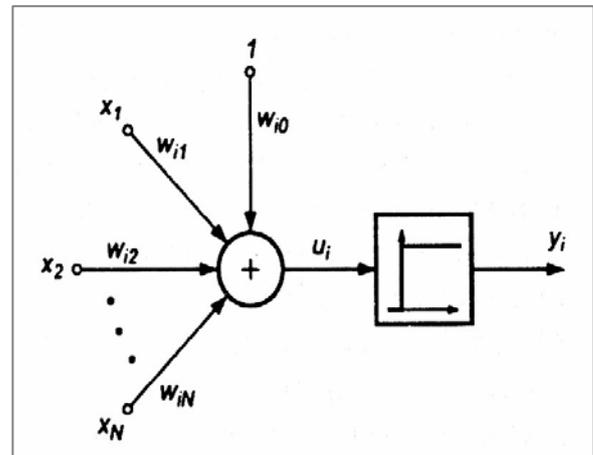
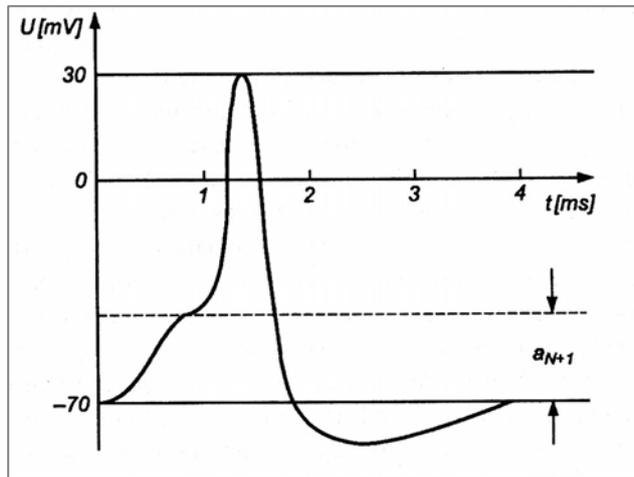
Основными достоинствами искусственных нейронных сетей (ИНС) и генетических алгоритмов (ГА) являются:

- 1) Хранение знания в виде большого числа мелких элементов, распределенных по сети, работа с зашумленными и недостаточными данными;
- 2) Использование принципов скрещивания и мутации- каждое последующее

поколение решений может быть лучше старого;

3) НС и ГА дают естественные модели параллельной обработки данных;

4) Рабочий процесс и процесс принятия решения не замедляется при возникновении большого количества данных.



а)

б)

Рис. 1.4.

Схема функционирования перцептрона

А) типичная схема нервного импульса

Б) модель нейрона Мак-Каллока-Питса, или простейший перцептрон

Как правило, при выполнении своих функций ИНС играет роль универсального аппроксиматора функции нескольких переменных, реализуя нелинейную функцию:

$$y=f(x) \tag{1.1}$$

где, x - это входной вектор; y - реализация векторной функции нескольких переменных. Входные сигналы $x_j (j=1, 2, \dots, N)$ суммируются с учетом соответствующих весов w_i (сигнал поступает в направлении от узла i к узлу j) в сумматоре, после чего результат сравнивается с пороговым значением. Аргументом функции выступает сигнал:

$$u_i = \sum_{j=1}^N w_{ij} x_j(t) + w_{i0} \tag{1.2}$$

где w_i - набор вещественных весовых коэффициентов, определяющие силу связи между нейронами, $\sum_{j=1}^N w_{ij} x_j + w_{i0}$ - уровень активации нейрона,

который определяется взвешенной суммой его входных сигналов.

Пороговая функция определяет активное или неактивное состояние нейрона и предназначена для вычисления выходного значения нейрона путем сравнения уровня активации с некоторым порогом.

Функция $f(ui)$ называется функцией активации. В модели МакКаллока-Питса –это пороговая функция вида:

$$f(u) \begin{cases} 1 & \text{для } u > 1 \\ 0 & \text{для } u \leq 0 \end{cases} \quad (1.3)$$

ИНС представляет собой параллельную, распределенную, адаптивную систему, которая восстанавливает скрытые закономерности и развивает свои способности при обработке информации и в результате обучения.

Таким образом, в ИНС:

- 1) Как в параллельных системах в любой момент времени в активном состоянии могут находиться несколько процессов;
- 2) Как в распределенных системах каждый из процессов может независимо обрабатывать локальные данные и принимать решения, а отдельные процессы обмениваются информацией между собой и внешней информационной средой через каналы связи;
- 3) Как в адаптивных системах процессы обработки информации организуются таким образом, чтобы достигнуть требуемых целей при неопределенных факторах, причем при изменении последних система способна адаптироваться.

ИНС обладает следующими признаками:

- 1) ИНС состоит из простых нейропроцессорных элементов (НЭ), или нейроэлементов, являющихся искусственным аналогом биологических нейронов;
- 2) НЭ связаны между собой направленными информационными каналами (ИК), по которым распространяются информационные сигналы, закодированные в скалярной форме;
- 3) Каждый НЭ может быть связан посредством входных ИК с множеством других НЭ;
- 4) Каждый НЭ имеет единственный выходной ИК, который в последствии может разветвляться;
- 5) Каждый НЭ может обладать собственной внутренней памятью и может осуществлять локальную обработку приходящей к нему информации;
- 6) Обработка информации осуществляется нейроэлементом локально - она зависит только от значений, поступающих к входным ИК, и значений, хранящихся в его внутренней памяти;
- 7) ИНС развиваются и адаптируются в процессе обучения по примерам.

Технологии обучения ИНС подразделяются на две категории:

- 1) Обучение с учителем, при котором имеется множество примеров, в

которых отклик или поведение НС известно;

2) Обучение без учителя или самообучение или самоорганизация, при котором процесс обучения НС происходит автономно - по мере поступления новой информации находят некоторые ее свойства и закономерности и НС обучается отражать их на выходе.

Как уже было сказано выше, другим альтернативным подходом к построению систем искусственного интеллекта являются генетические алгоритмы, представляющие собой алгоритмы, основанные на принципах биологической эволюции. Процессы отбора сильнейших представителей и социальные процессы были формализованы с помощью теории клеточных автоматов, генетических алгоритмов, генетического программирования, искусственной жизни и других форм эмерджентных вычислений.

ГА представляет собой эмерджентную или проявляющуюся модель, т.е. модель, которая имитирует наиболее элегантную и мощную форму адаптации - эволюцию форм жизни животного и растительного мира.

ГА и другие формальные эволюционные аналоги обуславливают наиболее точное решение задачи за счет операции над популяциями кандидатов на роль решения.

Решение задач с помощью ГА включает в себя три стадии.

1. Представление отдельных потенциальных решений в специальном виде, удобных для выполнения эволюционных операций изменения и отбора (битовые строки).

2. Реализация скрещивания и мутации, которая присуща биологическим формам жизни, в результате чего появляется новое поколение (особей) с рекомбинированными свойствами их родителей.

3. На основе некоторого критерия отбора выбираются лучшие формы жизни, т.е. наиболее точно соответствующие решению данной проблемы. Эти особи отбираются для выживания и воспроизведения, т.е. для формирования нового поколения потенциальных решений. В конечном счете некоторое поколение особей и станет решением задачи.

После анализа каждого кандидата выбираются пары для рекомбинации. Для рекомбинации используются генетические операторы, в результате выполнения которых новые решения получаются путем комбинации свойств родителей. Как и в естественном эволюционном процессе, степень участия в репродуктивном процессе определяется для каждого кандидата значением критерия качества: кандидат с более высоким значением критерия качества участвует в процессе воспроизводства с наибольшей вероятностью. Использование наиболее «слабых» кандидатов в процессе воспроизводства не исключается, так как выживание некоторых слабейших особей имеет важное значение для развития популяции: они могут содержать некоторые важные компоненты решения, например, фрагменты битовой строки, которые могут извлекаться при воспроизводстве.

Основными генетическими операторами являются:

1) Скрещивание (*crossover*) - два решения кандидата делятся на несколько частей и обмениваются этими частями, результатом становятся два новых

кандидата;

2) Мутация, которая состоит в случайном выборе кандидата и случайном изменении некоторых его свойств, например: мутация может состоять в случайном выборе бита в шаблоне и изменении его значения с 1 на 0 или на #, значение мутации состоит в возможном восполнении важных компонентов решения, отсутствующих в исходной популяции;

3) Инверсия - изменение порядка следования битов в битовой строке;

4) Обмен – перемена мест двух произвольных битов.

Работа ГА по выполнению генетических операторов продолжается до тех пор, пока не будет достигнуто условие его завершения, например, для одного или нескольких кандидатов значение критерия качества не превысит некоторого порога.

Основными областями применения ГА являются:

- решение комбинаторных задач и задач оптимизации; инструмент реализации процессов оптимизации при моделировании производственных процессов- оптимизация последовательности, упорядочения и выбора (*Simulationstool*);

- в качестве алгоритма нахождения экстремума многоэкстремальной функции.

Пример задачи оптимизации для генерации определенной последовательности - это выработка оптимальной последовательности загрузки одного из производственных участков. Подходящая симуляционная модель предоставит общее время прохождения загрузки для каждого предложенного решения. В процессе эволюции будут появляться все лучшие и лучшие последовательности, при которых задачи выполняются в наиболее короткие сроки.

Основным преимуществом ГА является их проблемная независимость и способность показать при этом хорошие результаты даже при большом количестве симулятивно- технических решений задач оптимизации.

Основными свойствами задач оптимизации, для решения которых используются ГА являются:

1) Большое пространство решений;

2) Неизвестные свойства пространства решений;

3) Дискретные пространства решений, которые не позволяют применения математической числовой оптимизации проблемы;

4) Множество мягких ограничений, т.е. ограничений которые не обязательно должны выполняться, но не учет которых может привести ухудшению результатов.

Количество возможных решений зависит от типа оптимизации, а также от количества элементов множества определения, которое содержит элементы подлежащие оптимизации.

Нейронные структуры и генетические алгоритмы привлекательны как средства реализации интеллекта по многим причинам. Они более пригодны для сопоставления зашумленных и недостаточных данных, т.к. они хранят знания в виде большого числа мелких элементов, распределенных по сети. За счет

использования принципов скрещивания и мутации, каждое последующее поколение решений может быть лучше старого. Нейронные структуры и генетические алгоритмы дают естественные модели параллельной обработки данных, поскольку каждый нейрон, или сегмент решения является собой независимый элемент. Наличие такого свойства структуры человеческого мозга, что на основе большего количества данных быстрее будет принято решение, в противовес компьютеру. Работа при этом не замедляется.

1.4. Роботизированные системы искусственного интеллекта

В настоящее время промышленность широко использует возможности систем искусственного интеллекта. В основном такие системы применяются в гибких производственных комплексах и робототехнических комплексах [2]. Систему искусственного интеллекта, которая может применяться в таких случаях, можно определить как комплекс программно-аппаратных средств для решения интеллектуальных задач, которые позволяют ЭВМ выполнять операции, аналогичные функциям человека, занятого умственным трудом. Поэтому под искусственным интеллектом производственных комплексов будем подразумевать алгоритмическое и программное обеспечение их адаптивных систем управления, позволяющее автоматизировать технологические операции интеллектуального характера. Отличительными признаками СИИ являются наличие баз данных и банков знаний, средств интерпретации задач и планирования их решений, а также связанных с ними алгоритмов формирования понятий, распознавания ситуаций и принятия решений. Решение проблемы представления знаний в памяти ЭВМ открыло принципиальную возможность понимания системами искусственного интеллекта естественного языка и речи. Оно позволило создать интеллектуальные терминалы и интерфейс, обеспечивающие непосредственное речевое или графическое (через дисплей) общение человека с ЭВМ или роботом на естественном языке, ограниченном данной предметной областью.

Системы искусственного интеллекта с базами данных и банками знаний нацелены на решение прикладных задач. Среди них важную роль играют экспертные системы искусственного интеллекта промышленного назначения, которые уже сегодня используются, например, для выработки рекомендаций по выбору компонентов и компоновки гибкого автоматического производства. К числу промышленных систем искусственного интеллекта можно отнести и интеллектуальный человеко-машинный интерфейс.

В последние годы созданы и начинают применяться в промышленности интеллектуальные системы автоматизированного проектирования, системы искусственного интеллекта для распознавания зрительной информации и речи, интеллектуальные системы автоматизации программирования, интеллектуальные автоматизированные системы подготовки производства, встроенные системы искусственного интеллекта для диагностики оборудования, а также ЛИСП - машины для оперативной обработки

символьной информации и ПРОЛОГ - машины для автоматического поиска логических выводов на основе факторов и правил, хранимых в базе знаний [20]. Это позволяет переложить на системы искусственного интеллекта значительную часть умственного труда, которую в условиях обычного производства приходилось возлагать на человека. В результате повышается производительность и степень автоматизации производства. Таким образом, сегодня системы искусственного интеллекта широко представлены на промышленном рынке. Более того, они находят все более широкое применение в адаптивных робототехнических комплексах и гибких технологических модулях.

В отличие от программных роботов, которые манипулируют деталями и инструментами, системы искусственного интеллекта манипулируют знаниями, необходимыми для решения возникающих задач. Тип задачи определяет особенности алгоритма ее решения. Для более четкого понимания специфики алгоритмов решения интеллектуальных технологических задач рассмотрим задачу планирования поведения робота. Решение этой задачи, получаемое в результате интеллектуальной деятельности робота - есть план целенаправленных действий, т. е. конечная последовательность операций, при фактическом выполнении которых гарантируется достижение цели. Каждая операция переводит робот в некоторое новое состояние, которое можно назвать очередной подцелью. В процессе поиска плана поведения могут использоваться как информация, поступающая от информационной системы робота, так и ранее накопленные знания и опыт, хранящиеся в базе знаний системы управления.

Особенности методов решения многих технологических задач гибкой автоматизации можно представить аналогичным образом. Например, при переналадке производства на выпуск нового изделия требуется спланировать, скоординировать и уложить в согласованную схему технологического процесса множество операций: выбор необходимого оборудования, оптимизацию технологических маршрутов, программирование систем управления, диагностику инструмента, контроль качества продукции и т. п. Переход на новую технологию может потребовать согласования основных технологических операций с вопросами совершенно иного характера, связанными, например, с финансированием или охраной окружающей среды. Все эти операции и вопросы взаимосвязаны и должны быть учтены при планировании технологического процесса. Для фактического осуществления этого процесса нужно соответствующим образом запрограммировать системы управления оборудованием гибкого автоматизированного производства, после чего может быть получено требуемое изделие с заданными свойствами.

Таким образом, чтобы решить задачу планирования необходимо составить хорошо скоординированную, согласованную схему операций (логических, математических, технологических и др.), начинающуюся с наперед заданных условий (предпосылок) и заканчивающуюся достижением цели. Решая такого рода интеллектуальные задачи, система искусственного интеллекта постоянно ищет пути обхода препятствий и достижения заданной

цели, пытается выработать какой-то план действий, следуя которому можно достичь эту непосредственно недоступную цель. Умение системы искусственного интеллекта решать интеллектуальные задачи посредством использования имеющихся знаний приобретаетс я путем обучения на опыте и адаптации. Это умение и связанные с ним навыки решения задач в известном смысле гораздо важнее для систем искусственного интеллекта, чем информация, хранящаяся в банке знаний, хотя, конечно, без необходимых знаний невозможно найти и решение.

Отдельно стоит отметить особенности интеллекта робототехнических производственных комплексов. Как правило, в настоящее время под интеллектом робототехнического комплекса понимают способность его системы управления решать технологические задачи интеллектуального характера посредством целенаправленного преобразования информации и знаний, обучения на опыте и адаптации к изменяющейся производственной обстановке [16]. Характерными чертами интеллекта робототехнического комплекса, согласно данному определению, являются их способность к переработке знаний, обучению, накоплению опыта и адаптации к заранее неизвестным и изменяющимся условиям в процессе решения задач. Благодаря этим качествам интеллектуальный робототехнический комплекс может решать самые сложные и разнообразные технологические задачи, а также легко перестраиваться с решения одного класса задач на другой. Таким образом, система управления робототехническим комплексом, наделенная элементами искусственного интеллекта, является универсальным средством решения широкого круга технологических задач. Она позволяет автоматизировать технологические операции интеллектуального характера.

Однако необходимо подчеркнуть, что возможности интеллекта робототехнических комплексов (как, впрочем, и возможности интеллекта человека) принципиально ограничены. Это связано с тем, что объем памяти, скорость запоминания и считывания информации в системах управления на практике ограничены. Кроме того, далеко не всегда удается найти эффективные алгоритмы решения интеллектуальных задач. Тем не менее благодаря способностям к обучению и адаптации робототехнические системы искусственного интеллекта обладают большой гибкостью и могут быстро осваивать новые «профессии», автоматизируя соответствующие интеллектуальные функции человека.

2. Особенности использования интеллектуальных систем и систем искусственного интеллекта в приборостроении

2.1 Элементы искусственного интеллекта, применяемые в приборостроении

Как уже упоминалось выше, интеллектуальные технологии находят всё большее применение в различных областях человеческой деятельности. Использование интеллектуальных производственных систем позволило существенно повысить качество выпускаемой продукции за счёт автоматизации большинства операций, в том числе тех, которые до последнего времени считались мало пригодными для полной автоматизации, например, высокоточная наладка технологического оборудования. Современные адаптивные интеллектуальные системы позволяют автоматизировать практически любую стадию производства приборов и устройств. С другой стороны, стоимость такой автоматизации всё ещё довольно высока и экономическая эффективность внедрения той или иной интеллектуальной производственной системы требует тщательного изучения и обоснования.

Основной особенностью интеллектуальных производственных систем является то, что они оперируют со знаниями [13]. Таким образом, обычный программный технологический комплекс, оперирующий на прямую с деталями или заготовками, является низшим звеном интеллектуального производственного комплекса. Отсюда следует, что интеллектуальную производственную систему можно условно разделить на два модуля – «механический» и «интеллектуальный». Так называемый механический модуль отвечает непосредственно за процесс какого-либо физического воздействия на элементы приборов и устройств, тогда как задача интеллектуального модуля – управление и контроль процесса такого воздействия. Как правило, под механическим модулем понимается либо автоматизированный робототехнический комплекс, либо некая гибкая производственная система. Устройство и особенности таких комплексов будут рассмотрены в следующих главах. Наиболее сложным элементом интеллектуальной производственной системы является её интеллектуальный модуль. Развитие интеллектуальных систем напрямую связано с прогрессом в области информационных технологий, темп развития которых постоянно увеличивается. Сама интеллектуальная производственная система состоит из нескольких элементов, каждый из которых является, по своей сути, отдельным направлением научных знаний.

Таковыми элементами являются:

- интеллектуальное программирование и интеллектуальный интерфейс;
- система адаптивного поиска логического вывода и языковые средства;
- система представления знаний и формирования понятий;
- система формирования адаптивных решающих правил;
- система логического анализа и распознавания сцен;

-система искусственного зрения и распознавания.

Одной из наиболее важных и дорогостоящих компонентов интеллектуальной производственной системы является алгоритмическое и программное обеспечение автоматического управления оборудованием. При его разработке речь идет о создании не отдельных алгоритмов и реализующих их программных модулей, а о сложных программных комплексах, включающих сотни тысяч команд. Для создания таких комплексов нужен труд десятков алгоритмистов и программистов. Отладка и сопровождение программного обеспечения в производственных условиях также требует значительных затрат. В условиях частой смены выпускаемой продукции приходится непрерывно модифицировать алгоритмы и дополнять программные комплексы. По мере расширения областей применения интеллектуальной системы требуются все новые и новые комплексы алгоритмов и программ. Однако их разработка сдерживается не только трудоемкостью самого процесса алгоритмизации и программирования, но и дефицитом квалифицированных кадров. Последние не могут бросить ранее созданные ими программные комплексы и должны их «сопровождать». Все это говорит о том, что алгоритмическое и программное обеспечение становится одним из самых узких мест при разработке и эксплуатации интеллектуальной системы. Его стоимость зачастую превышает стоимость вычислительной техники, на базе которой строится система управления интеллектуальной технологической системой. В этом отношении показательны следующие данные. В США стоимость затрат на разработку программного обеспечения в 2004 г. составила около 40 млрд. долларов. Ожидается, что в 2010 г. она достигнет 125 млрд. долларов [19]. Отсюда следует, что при низких темпах роста производительности труда программистов к разработке алгоритмического и программного обеспечения интеллектуальных систем требуется привлечь большое число специалистов, которых еще предстоит обучить. Сегодня производительность ЭВМ на два порядка превышает производительность программиста при разработке программ. За время, когда производительность ЭВМ возросла в тысячу раз, производительность программиста увеличилась только в десять раз. Поэтому вопрос о разработке и совершенствовании алгоритмического и программного обеспечения для интеллектуальных систем приобретает все большую остроту. Предлагаемые для его разрешения многочисленные рецепты, связанные с внедрением новых технологий программирования и специальных инструментальных систем, практически малоэффективны.

Одним из наиболее перспективных средств резкого ускорения процесса программирования является применение методов искусственного интеллекта. В рамках такого подхода ЭВМ применяется не только на этапе собственно программирования, т. е. синтеза и кодирования программ, но и в процессе постановки задач. Этот принципиально новый подход можно назвать интеллектуальным или концептуальным программированием. Его особенность заключается в том, что с самого начала программирование ведется в терминах и понятиях предметной области решаемых задач, например, в области

приборостроения. Формализованные понятия далее используются для описания задач и автоматического синтеза программных комплексов их решения. Эти понятия, описания и программы представляют собой по существу машинное представление знаний, достаточных для решения некоторого класса задач. На их основе строится интеллектуальный банк знаний. Концептуальный подход позволяет переложить на ЭВМ значительную часть интеллектуальных операций, которые при традиционном программировании приходится выполнять специалистам. В результате удается резко повысить производительность программирования и, тем самым, высвободить значительное число программистов.

Алгоритмизацию и программирование адаптивных систем управления отдельных интеллектуальных производственных комплексов в целом можно рассматривать как своеобразный технологический процесс составления планов решения задач на ЭВМ. Достаточно детальные планы, записанные на понятном ЭВМ языке, составляют суть программного обеспечения с элементами искусственного интеллекта. Главным достоинством такого интеллектуального программного обеспечения является возможность автоматического синтеза программ для решения любой задачи из некоторого класса, сведения о котором хранятся в банке знаний. Для реализации этой возможности служат диалоговый процессор и специальная организующая программа — монитор. Последняя, в соответствии с заданной технологией, вызывает из банка знаний необходимые программные модули, снабжает их соответствующими исходными данными и текущей информацией, поступающей по каналам обратных связей, и «собирает» рабочий набор программ, обеспечивающий адаптивное управление интеллектуальным производственным комплексом в изменяющихся производственных условиях.

Интеллектуальные пакеты управляющих программ, использующие банки знаний, позволяют автоматизировать процесс программирования роботов и другого оборудования интеллектуальных производственных систем непосредственно под заданную технологическую задачу. При этом исходные данные задачи и сам технологический процесс, подлежащий реализации на данном интеллектуальном комплексе, могут быть заданы на естественном языке технолога, а не на алгоритмическом языке ЭВМ. По этим данным автоматически строится алгоритмическая модель технологического процесса, а по ней «собирается» рабочий набор программ адаптивного управления оборудованием интеллектуального технологического комплекса, реализующий заданный технологический процесс.

Совершенствование средств интеллектуального программирования постепенно избавляет технологов от рутинной работы по программированию движений роботов и другого оборудования интеллектуального производственного комплекса. Диалоговый процессор вместе с интеллектуальным монитором позволяет технологу описывать роботизированные технологические процессы на более высоком уровне и на более естественном языке, не прибегая к трудоемкому программированию в кодах ЭВМ. Таким образом, открывается реальная перспектива снять с

технологов функции программистов. Для этого создаётся интеллектуальный технологический интерфейс, обеспечивающий общение технолога с управляющими ЭВМ на профессиональном языке. Для создания такого интерфейса необходимо, в свою очередь, разработать эффективные методы представления знаний, организации диалога и понимания естественного языка. Эти методы относятся к области искусственного интеллекта и успешно реализуются на практике. Более подробно такие методы будут рассмотрены в последующих главах.

Таким образом, практическая значимость новой технологии организации интеллектуального человеко-машинного интерфейса заключается в том, что она обеспечивает прямой доступ к современным высокопроизводительным ЭВМ непрограммирующим специалистам (например, конструкторам или технологам). Для программирования интеллектуальной производственной системы достаточно ввести в банк знаний систему понятий, описывающих рассматриваемый класс технологических процессов, а затем в терминах этих понятий на естественном языке описать конкретное задание, возлагаемое на данный интеллектуальный комплекс. По этому описанию ЭВМ автоматически строит рабочую программу адаптивного управления интеллектуальным производственным комплексом, используя алгоритмы, хранящиеся в банке знаний.

2.2 Особенности представления знаний в интеллектуальных производственных системах

Таким образом, все системы искусственного интеллекта являются системами, базирующимися на знаниях. Основными свойствами таких систем являются следующие.

1. Раздельное хранение знаний, представленных в символьной форме и компонентов обработки этих знаний.

Проведение четкой границы между знаниями самими по себе и формальными системами для их представления и обработки является обязательным для понимания основополагающей концепции обработки знаний. Именно такое разделение является первым шагом на пути демистификации понятия «Искусственный интеллект»

2. Системы могут делать выводы и принимать решения на основании сохраненной информации, которая представляется в системах не явно, однако органически свойственна этим системам.

Типичным является то, что выводам систем, базирующихся на знаниях, присущ не детерминированный (не определенный) характер.

3. Системы обладают способностью к пояснениям, т.е. должны быть в состоянии, по требованию вывести для пользователя понятную и ясную для него цепочку рассуждений (например: шагов резолюции). Способность к четкому пояснению является важным фактором, так как признание интеллектуальной системы пользователем по большей части зависит именно от

качества данной способности к объяснению.

4. Способность к обучению, которая подразумевает, например, способность выводить новые знания на основании информации, полученной от пользователя (информация об удаче или неудаче прошедшего сеанса работы с программой).

Архитектура и основная концепция систем, базирующихся на знаниях, представлена на рис.2.1. Кроме того, на рисунке 2.1 представлен обмен вышеназванными свойствами посредством структуры функциональной взаимосвязи компонентов.

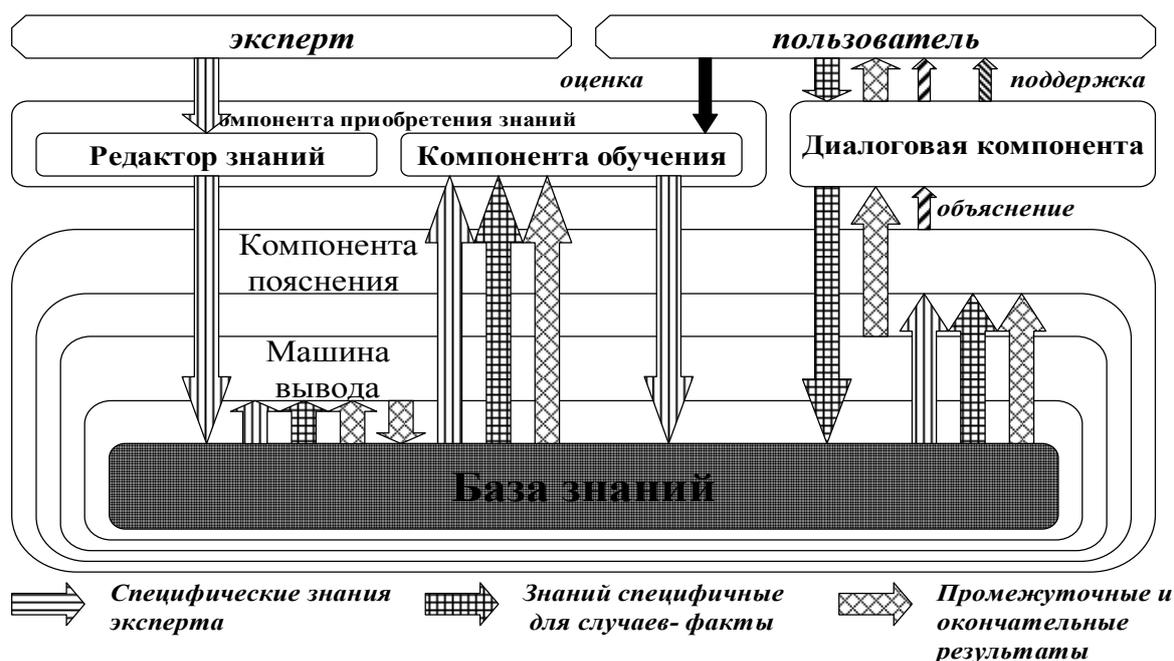


Рис. 2.1 Архитектура систем, базирующихся на знаниях (компонента)

Интеллектуальные система разделяют следующие типы представленных знаний.

1. Специфические для данной области знания эксперта:

- «статистическая база знаний», сформированная при проектировании системы;
- содержание остается неизменным при использовании системы;
- изменение возможно посредством обучения системы.

2. Факты- знания о специальных случаях:

- пополняются за счет ввода знаний пользователем во время работы с системой.

3. Промежуточные и окончательные результаты:

- получаются посредством осуществления выводов на основании имеющихся знаний

Принято различать следующие уровни представления знаний:

- 1) Уровень пользователя - проблемно-ориентированный;
- 2) Уровень, ориентированный на реализацию программной системы „tool“-ориентированный (знания представленные на языке программирования системы,);
- 3) Системный уровень (биты) - имманентный системе (внутрикомпьютерное представление).

Для организации осмысленного диалога между человеком и ЭВМ и автоматизации решения интеллектуальных задач в процессе управления интеллектуальным производственным комплексом необходим специальный язык. Этот язык должен быть удобным средством формулировки заданий, представления знаний, поиска планов и принятия решений. Он должен также предоставить системе управления интеллектуальным производственным комплексом с элементами искусственного интеллекта необходимую основу для логических умозаключений. Благодаря способности рассуждать на своем внутреннем языке формул система управления сможет решать многие интеллектуальные задачи на самом высоком уровне.

Задачи ИИ не решаются с помощью «подгонки» к уже имеющимся понятиям. Они связаны с качественными, а не количественными проблемами, с аргументацией, а не вычислениями, с организацией общих объемов знаний, а не реализацией отдельного четкого алгоритма.

В соответствии с этим , язык представления знаний для систем ИИ должен удовлетворять следующим требованиям.

1. Обработка знаний, выраженных в качественной форме.

Демонстрация связи между элементами описываемой задачи и нашим собственным представлением и пониманием описываемой предметной области.

2. Логическое получение новых знаний из набора фактов и правил.

Язык ИИ должен обеспечить возможность рассуждения об абстрактных описаниях классов, объектов и состояний. Для этого система должна обладать правилами, которые позволяют ей делать логические выводы, исходя из имеющихся фактов.

3. Отображение общих принципов, наряду с конкретными ситуациями.

Введение использования переменных, посредством которых осуществляется обобщение. Переменные позволяют проецировать конкретные ситуации на общие понятия.

4. Передача сложных семантических значений , которая должна осуществляться за счет:

- использования семантических отношений для описания причинных связей между событиями и описания взаимодействия и способов соединения частей чего-либо, а также представления необходимых планов решения с помощью последовательности элементарных действий, которые должны быть выполнены в определенном порядке;

- использования механизма наследования – это инструмент представления таксономической (классифицированной) структурированной информации, который гарантирует, что все члены класса обладают общими свойствами.

5. Рассуждение на метауровне.

«Осведомленность системы о своих знаниях» - это более высокий уровень знаний, называемый - метазнаниями. Метазнания необходимы для проектирования и адекватного описания интеллектуальных систем.

Интеллектуальная система (ИС) должна не только знать предмет, она должна также знать, что она обладает знаниями о предмете.

ИС должна быть способна решать задачи и объяснять эти решения.

ИС должна описывать свои знания как в конкретных, так и в обобщенных терминах, узнавать их ограничения и учиться в процессе взаимодействия с миром.

Проблемно-ориентированные знания могут быть представлены двумя способами.

1. Декларативное (дескриптивное) представление.

Идея представления знаний базируется на принципе того, что знание можно рассматривать независимо от того, как оно будет обрабатываться. Знание и методы его обработки будут четко отделены друг от друга.

При этом данному способу представления свойственны следующие особенности:

- приобретение знаний сконцентрировано на содержании;
- сохранение только один раз вопреки тому, что применение осуществляется в различных контекстах;
- различные «модели» могут быть обработаны одной и той же машиной вывода;
- возможна модификация базы знаний без побочных эффектов;
- никакой эффективной обработки.

2. Процедуральное (императивное) представление.

Идея процедурального представления знаний подчеркивает аспект их обработки. При этом, к представляемым знаниям будут присоединены знания об их обработке (метазнания) явно или не явно и, таким образом, выше указанное четкое отделение, которое присуще декларативному представлению будет затерто.

Особенностями такого представления являются:

- эффективная обработка;
- плохая или практически отсутствующая возможность чтения, редактирования или интерпретации знаний.

2.3 Языки и модели представления знаний

В интеллектуальных системах приборостроения наибольшее распространение получили следующие языки и модели представления знаний [7]:

- 1) язык исчисления предикатов первого порядка (логическая модель);
- 2) семантические или когнитивные сети;
- 3) фреймы;
- 4) продукционные правила.

При этом следует отметить, что применяемые декларативные (дескриптивные) формы представления часто расширяются за счет процедуральных (императивных) компонентов (рис. 2.2).

Для организации осмысленного диалога между человеком и ЭВМ и автоматизации решения интеллектуальных задач в процессе управления интеллектуальным производственным комплексом необходим специальный язык. Этот язык должен быть удобным средством формулировки заданий, представления знаний, поиска планов и принятия решений. Он должен также предоставить системе управления интеллектуальным производственным комплексом с элементами искусственного интеллекта необходимую основу для логических умозаключений. Благодаря способности рассуждать на своем внутреннем языке формул система управления сможет решать многие интеллектуальные задачи на уровне здравого смысла.

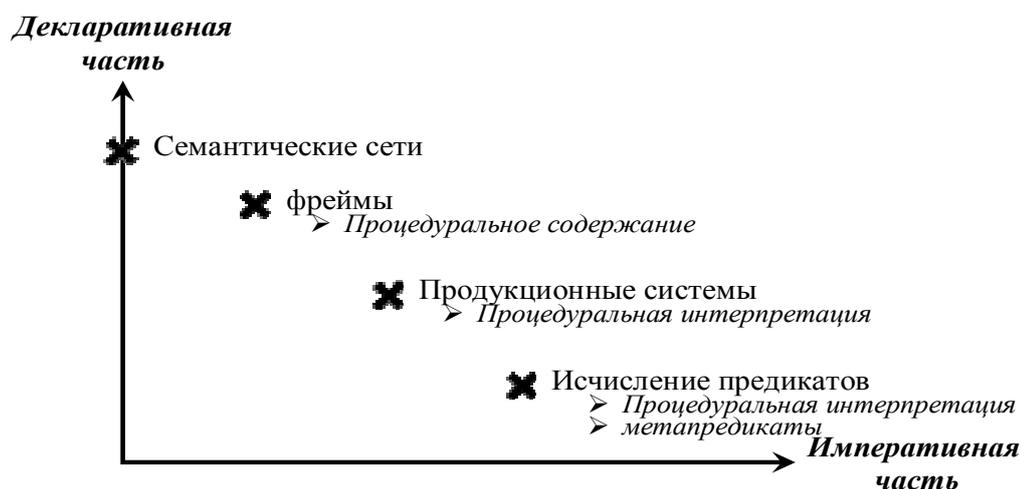


Рис. 2.2 Соотношение декларативных и процедуральных компонентов в моделях представления знаний

Для организации осмысленного диалога между человеком и ЭВМ и автоматизации решения интеллектуальных задач в процессе управления

интеллектуальным производственным комплексом необходим специальный язык. Этот язык должен быть удобным средством формулировки заданий, представления знаний, поиска планов и принятия решений. Он должен также предоставить системе управления интеллектуальным производственным комплексом с элементами искусственного интеллекта необходимую основу для логических умозаключений. Благодаря способности рассуждать на своем внутреннем языке формул система управления комплексом сможет решать многие интеллектуальные задачи на самом высоком уровне.

Одним из основных инструментариев, которые используются для управления интеллектуальными технологическими комплексами, является так называемый язык предикатов или исчисление предикатов [12]. Исчисление предикатов является простейшим логическим языком, удовлетворяющим основным требованиям, предъявляемым к интеллектуальным производственным системам. На этом формальном языке можно записать многие предложения и утверждения естественного языка, что позволяет формализовать знания. В исчислении предикатов содержатся также процедуры (правила) поиска логического вывода. Это позволит придать ЭВМ способность логически рассуждать и строить умозаключения, необходимые для понимания смысла задания и решения возникающих интеллектуальных задач.

Исчисление предикатов содержит следующий экономный алфавит символов[12]:

1. Предметная область и термы. Объекты и понятия, которыми приходится оперировать при решении той или иной интеллектуальной задачи, относятся к некоторому множеству Ω , называемому предметной областью. Фиксированные элементы этой области называются предметными постоянными (константами). Переменные, принимающие значения из Ω , называются предметными переменными. Предметные переменные, константы, а также функции от них называются термами.

2. Переменные высказывания и предикаты. Переменные, принимающие значения «истина» (И) или «ложь» (Л), называются переменными высказываниями. Функции, аргументы которых принимают значения из области Ω , а сами функции принимают всего лишь два значения (И или Л), называются предикатами. Предикат, аргументами которого являются n предметных переменных, называется n -местным. Если $n = 1$, то предикат обычно определяет некоторое свойство предмета, если $n \geq 2$, то предикат может выражать n -арное отношение между предметами.

Таким образом предикат указывает на отношение между несколькими объектами (в том числе нулевым числом объектов) в мире. Количество объектов связанных таким образом определяют арность предиката.

Предикат арности n однозначно отображает множество элементов n из области определения I в однозначно определяемое множество значений истинности.

$$I_n \Rightarrow \{ \text{wahr}, \text{falsch} \}$$

3. Элементарные (атомарные) формулы. Высказывания и выражения вида $A(\omega)$, $F(b,c)$, где A , F - предикаты, а ω , b , c - предметные переменные или константы, называются элементарными, или атомарными формулами. Эти формулы (как высказывания, так и предикаты) всегда принимают лишь два значения: истинно или ложно, поэтому их можно связывать с помощью логических операций, образуя новые формулы.

4. Логические операции. К числу логических операций относятся конъюнкция $\&$ («И»), дизъюнкция \vee («или», «и/или»), отрицание \neg («не», «неверно, что ...»), импликация \rightarrow («если ..., то ...», «влечет»), эквивалентность \leftrightarrow («эквивалентно», «тогда и только тогда»). Эти операции определяются следующим образом: $A \rightarrow B$ истинно тогда и только тогда, когда A и B имеют одинаковые значения; $A \rightarrow B$ ложно тогда и только тогда, когда A истинно, а B ложно; $A \& B$ истинно тогда и только тогда, когда A и B истинны; $A \vee B$ ложно тогда и только тогда, когда A и B ложны; $\neg A$ истинно тогда и только тогда, когда A ложно.

5. Скобки и кванторы. Кроме пяти упомянутых логических связок, в исчислении предикатов употребляются еще скобки $()$ и две новые операции \forall , \exists , выражающие собой утверждения всеобщности и существования. Символ \forall называется квантором всеобщности, а символ \exists - квантором существования.

При этом, формулой исчисления предикатов называется конечная последовательность из символов, которая строится на базе элементарных (атомарных) формул путем перехода от формулы A к формулам типа $\forall\omega A$, $\exists\omega A$ и от формул A и B к формулам типа $A\&B$, $A\rightarrow B$ и т.д. Если в формуле A сделать замену переменных (как свободных, так и связанных), то полученное выражение снова будет формулой. Элементарная формула или ее отрицание, входящие в правильно построенную формулу, называются литерами, а дизъюнкция литер называется простым дизъюнктом. Если дизъюнкт не содержит никаких литер, то он называется пустым дизъюнктом и обозначается *nil*. Язык логики предикатов содержит в себе всю алгебру высказываний, т. е. элементарные высказывания, принимающие значения «истинно» или «ложно», логические операции и, следовательно, все мыслимые формулы. Кроме того, этот язык допускает переменные высказывания, отнесенные к произвольной предметной области. Благодаря этому формулам языка можно дать ту или иную интерпретацию, т. е. с содержательной (семантической) точки зрения формулы можно сделать осмысленными.

Одним из наиболее удобных средств поиска логического вывода является так называемый метод резолюций [11]. В основе этого метода лежит идея доказательства от противного. Она заключается в том, что вместо заданной формулы B , которая предполагается тождественно истинной, рассматривается ее отрицание $\neg B$ и доказывается противоречивость этой формулы. Процесс поиска логического вывода завершается, как только устанавливается справедливость некоторых альтернативных формул L и $\neg L$. Полученное противоречие доказывает неудовлетворимость формулы $\neg B$ и, следовательно, тождественную истинность исходной формулы B .

Принцип резолюции в исчисление высказываний состоит в выборе двух дизъюнктов D_i и D_j , в один из которых входит литера L , а в другой - ее отрицание $\neg L$. Резольвентой называется новая формула $R = P \vee Q$, получаемая из $D_i = P \vee L$ и $D_j = Q \vee (\neg L)$ путем вычеркивания литер L и $\neg L$. Это соответствует применению правила «модус поненс» к рассматриваемым дизъюнктам. В исчислении предикатов принцип резолюций усложняется. В этом случае дизъюнкты зависят от переменных.

Пусть, например, $D_i = P(\omega) \vee L(\omega)$, $D_j = Q(\omega) \vee (\neg L(\varphi(\omega)))$. Теперь уже нельзя вычеркнуть литеры $L(\omega)$ и $\neg L(\varphi(\omega))$, так как они зависят от разных переменных. Поэтому приходится подставлять вместо этих переменных подходящие термы. Так, подставляя в D_i - вместо ω терм $\tau = \varphi(\omega)$, получим $D_i = P(\varphi(\omega)) \vee L(\varphi(\omega))$. Отсюда находим резольвенту $R = P(\varphi(\omega)) \vee Q(\omega)$. Получение очередной резольвенты в форме пустого дизъюнкта *nil* свидетельствует о том, что доказываемая формула B действительно логически следует из заданной системы формул.

Число резольвент, формируемых в процессе поиска логического вывода, конечно. Оно существенно зависит от выбора стратегии поиска, т. е. правила выбора дизъюнктов для синтеза очередной резольвенты. Большой практический интерес представляет оптимальная стратегия, позволяющая получить решение за минимальное число шагов. Синтез такой стратегии, связанный с нахождением кратчайшего пути на графе, наталкивается на значительные трудности. Поэтому разработано много эвристических стратегий, позволяющих сокращать число резольвент, необходимых для решения задачи. Например, стратегия предпочтения одночленов или разного рода семантические и адаптивные стратегии [12]. В некоторых интеллектуальных задачах факт выводимости заданной формулы B (тракуемой как задание или вопрос) из системы формул (тракуемых как аксиоматическое описание знаний и накопленного опыта) оказывается недостаточным. Примером может служить задача планирования поведения манипулятора робота. В подобного рода задачах нужно знать тот ответный терм τ , при котором данная формула $B(\omega)$ логически выводима из системы аксиом. Иными словами, нужно выяснить, следует ли логически формула $\exists \omega B(\omega)$ из заданной системы аксиом и, если следует, то при каком значении переменной $\omega = \tau$ это достигается.

Таким образом, именно умение отыскивать такие значения (ответный терм) для переменной, связанной квантором существования, позволяет ставить роботу вопросы общего характера и осуществлять диалог с ним. Например, если спросить у робота: «Какие действия и в какой последовательности нужно совершить, чтобы собрать из имеющихся деталей определенную конструкцию?», то ответом на этот вопрос должна быть не просто констатация факта, что сборка данной конструкции возможна, а робот должен предложить еще и развернутый план (технологический маршрут) сборки.

Описанный язык исчисления предикатов и методы поиска логического вывода представляют собой эффективное средство для автоматического решения широкого класса интеллектуальных задач. Этот логический язык, а также связанный с ним метод резолюций позволяют ЭВМ логически

рассуждать и самообучаться. Первым этапом автоматического решения задач, требующих логического анализа, является формулировка этих задач в терминах исчисления предикатов. Для этого нужно, прежде всего, задать предметную область, т. е. совокупность относящихся к решаемой задаче объектов, и выделить их существенные свойства. Далее нужно, присвоив определенный содержательный (семантический) смысл предикатным и функциональным символам, формализовать данные и условия задачи в виде правильно построенных формул, удовлетворяющих рассматриваемой интерпретации. Эти формулы, описывающие условия задачи, называются аксиомами. Аксиомы можно рассматривать как концептуальное определение совокупности рассматриваемых объектов, их свойств и отношений между ними. Формулировка задачи на языке исчисления предикатов - это один из наиболее ответственных этапов интеллектуального программирования. На этом этапе от инженера требуются глубокие знания существа решаемой задачи. Практически весьма важно, чтобы формулировка задачи (связанная с заданием системы аксиом и формулы цели) была, по возможности, простой. Эффективность поиска логического вывода можно увеличить путем уменьшения числа предикатов и аксиом, определяющих данные и условия задачи. С этой целью разумно использовать ранее доказанные формулы или ввести более сложные предикаты, образующие новые аксиомы, которые можно рассматривать как результат обучения в процессе решения задачи. Такие аксиомы, описывающие на языке исчисления предикатов приобретаемый опыт, можно назвать аксиомами обучения. Введение аксиом обучения позволяет увеличивать и улучшать знания о рассматриваемом классе задач в процессе их непосредственного решения. Таким образом, аксиомы обучения являются по существу средством представления новых знаний и уточнения старых.

При интеллектуальном программировании важную роль играют стратегии поиска логического вывода. Именно стратегия определяет, в каком порядке будут образовываться резольвенты и, следовательно, насколько быстро будет найдено решение задачи. Образно говоря, стратегия - это та активная часть, способная к обучению и адаптации, которая определяет машинный способ рассуждений в процессе решения задач. Она как бы подсказывает ЭВМ идею поиска логического вывода, исходя из заданной системы аксиом, в которой заключены необходимые для решения задачи знания. Если идея (стратегия) хороша, то решение будет найдено быстро. Весьма важно также, чтобы в системе аксиом были достаточно полно отражены не только необходимые сведения об условиях задачи, но и прошлый опыт решения задач. Поскольку хорошие идеи (стратегии) часто имеют своим источником прошлый опыт и ранее приобретенные знания, то это наводит на мысль о целесообразности синтеза и использования адаптивных стратегий. Стратегия поиска логического вывода называется адаптивной, если она целенаправленно меняется (подстраивается) в процессе решения задач в зависимости от приобретаемого опыта [12]. Примерами простейших адаптивных стратегий могут служить семантические стратегии, в которых сначала выбираются термины, соответствующие наиболее интересным объектам, затем - предикаты,

описывающие их свойства, и, наконец, формулы, содержащие эти свойства.

При синтезе адаптивных стратегий важную роль играет критерий предпочтения аксиом обучения (хранящихся в памяти ЭВМ наряду с исходной системой аксиом), который позволяет уменьшить исходную неопределенность относительно условий решения задачи. Использование аксиом обучения особенно эффективно в тех случаях, когда в них либо раскрывается неопределенность (т. е. содержится новая необходимая для решения задачи информация), либо запоминается в компактной форме часто встречающийся в рассматриваемом классе задач фрагмент решения. Таким образом, если в процессе решения очередной задачи потребуются вывести уже доказанную ранее формулу, то критерий предпочтения аксиом обучения сократит общее число шагов поиска логического вывода по крайней мере на длину вывода соответствующей аксиомы обучения.

Знание языка исчисления предикатов и связанных с ним средств поиска логического вывода является необходимым для решения задач автоматического формирования понятий, представления знаний, синтеза решающих правил, распознавания объектов и анализа сцен. В свою очередь, без решения этих задач невозможно осуществить внедрение элементов искусственного интеллекта в производственно-технологические системы.

Проблема формирования понятий и представления знаний играет важную роль не только при организации интеллектуального интерфейса, но и при разработке адаптивных систем распознавания для интеллектуальных производственных комплексов. Острая необходимость в эффективных методах решения этой проблемы возникает, например, при распознавании речевых команд, при анализе видеосцен и при моделировании окружающей среды в памяти управляющей системы комплекса.

Для формирования понятий и представления знаний, связанных с задачей распознавания, используется описанный выше язык исчисления предикатов. При этом роль предметной области Ω здесь играет множество объектов из M классов, а каждому классу объектов Ω_k , $k = 1, \dots, M$, соответствует свой решающий предикат, представляющий собой характеристическую функцию класса Ω_k . В режиме обучения интеллектуальной производственной системе предъявляются типичные образцы объектов из разных классов. Например, в случае сборочного комплекса предъявляются детали, из которых нужно собрать изделие. При этом «учитель», в роли которого обычно выступает человек, сообщает системе, к какому классу каждый данный объект принадлежит. Подмножество предъявленных «эталонных» объектов Ω_0 называется обучающей выборкой. По обучающей выборке легко построить логическое описание всех эталонных объектов. Процесс построения аксиом классов (по мере предъявления интеллектуальному комплексу эталонных объектов) по сути представляет собой процесс формирования понятий интеллектуального производственного комплекса об объектах окружающей среды. Однако на практике ввиду ограниченности обучающей выборки построить такую идеальную систему аксиом обычно не удается. Поэтому в принципе возможны ошибки при распознавании объектов, не входящих в обучающую выборку.

Для оценки качества системы аксиом классов существует понятие экстраполирующей силы. Обычно качество системы аксиом классов оценивается либо на обучающей выборке, либо на контрольной выборке, если таковая имеется. Чем больше экстраполирующая сила, тем лучше сконструированная система аксиом, т. е. тем большую информацию о классах она несет.

Благодаря обучению понятиям интеллектуальный производственный комплекс приобретает способность к решению задач распознавания. Среди них наиболее важными в прикладном отношении являются такие задачи, как классификация речевых команд или идентификация дикторов, распознавания обычных яркостных или дальностных (т. е. полученных с помощью дальномеров) изображений отдельных объектов и рабочих сцен, находящихся в поле зрения комплекса. Характерной особенностью ряда задач такого рода является то, что классы распознаваемых объектов инвариантны относительно той или иной группы преобразований. Многие традиционные методы формирования понятий основываются на построении их описаний в том или ином, заранее выбранном классе функций. Такие описания определяют в пространстве признаков поверхности, разделяющие классы. Решающие предикаты при этом строятся по обучающей выборке, а применяются к новым объектам. Поэтому весьма важно, чтобы решающие предикаты обладали высокой экстраполирующей силой не только на обучающей выборке, но и на всем множестве распознаваемых объектов. Требование высокой экстраполирующей силы решающих предикатов в общем случае трудно формализуемо. Однако для некоторых конкретных классов функций увеличение экстраполирующей силы связано с минимизацией сложности описаний. Так, при синтезе полиномиальных описаний решение задачи сводится к построению полиномов минимальной степени с минимальным числом членов в разложении, обеспечивающего безошибочную классификацию элементов обучающей выборки. Стоит также отметить, что принцип минимальной сложности был положен в основу синтеза адаптивных логических решающих правил и реализующих их оптимальных распознающих графов.

Таким образом, задача обучения понятиям сводится к построению по обучающей выборке идентифицирующих правил минимальной сложности в классе. При этом сложность может трактоваться по-разному. Следует отметить, что именно требование минимальной сложности, позволяющее строить наиболее простые и информативные идентифицирующие правила, отличает метод минимальной сложности от других методов формирования понятий, рассмотренных в работах [11,12].

Распознающие правила иногда удобно строить в виде ориентированного графа — разветвленного дерева решений. Узлам такого графа соответствуют некоторые предикаты-признаки; ветвям, исходящим из узла - возможные значения этих признаков. Все ветви заканчиваются листьями, которые ставятся соответственно номеру классов. Каждая ветвь, заканчивающаяся листом, содержит такую совокупность внутренних узлов, среди которых нет узлов, соответствующих одному и тому же значению предикатов. Распознающее

правило, представленное в такой форме, называется последовательным распознающим правилом, а реализующий его граф типа «дерево классов» распознающим графом. Процесс распознавания на таком «дереве классов» представляет собой последовательное «раскрытие» узлов графа, т. е. определение значений соответствующих предикатов на распознаваемом объекте. Любая ветвь на распознающем графе, соответствующая последовательности «раскрытых» узлов, приводит (с определенной вероятностью) к тому или иному классу. Задача синтеза последовательных логических распознающих правил и реализующих их графов минимальной сложности сводится к построению бинарно-древовидной дизъюнктивной формы, обладающей теми же свойствами, что и описанные выше оптимальные идентифицирующие правила. Таким образом, возникает последовательная разновидность ранее сформулированной задачи формирования понятий: построить оптимальное распознающее правило, но не в классе дизъюнктивной формы, а в более узком классе бинарно-древовидных дизъюнктивных форм. Совершенно аналогично формулируются задачи синтеза понятий в тех случаях, когда признаки являются не двузначными, а многозначными предикатами.

Достоинством оптимальных идентифицирующих и распознающих правил является гарантируемая ими высокая точность распознавания при весьма экономном логическом описании классов и представлении соответствующих понятий. Это приводит в ряде случаев к значительной экономии времени и средств (в частности, памяти системы управления интеллектуальным производственным комплексом), необходимых для распознавания различных классов объектов и ситуаций. Для представления знаний в классах объектов внешней среды в виде формул исчисления предикатов можно использовать либо информацию, заключенную в обучающей выборке, либо словесное описание классов.

Таким образом, система управления интеллектуальным комплексом, реализующая описанные выше логические средства формирования понятий и представления знаний, автоматически строит (по обучающей выборке или исходя из априорного описания классов) проблемно ориентированную систему аксиом классов, обладающую свойствами полноты, непротиворечивости и инвариантности. Получающаяся в результате аксиоматическая система понятий выступает как эффективное средство логического представления знаний о внешней среде в памяти управляющей ЭВМ комплекса.

В свою очередь, для распознавания классов ситуаций и принятия адекватных решений системе управления интеллектуальным производственным комплексом необходимо сформировать соответствующие решающие правила. Рассмотрим общую схему и конкретные алгоритмы синтеза и оптимизации адаптивных логических решающих правил. Принцип минимальной сложности, лежащий в основе их синтеза, обеспечивает простоту реализации и высокую экстраполирующую силу. Благодаря этим качествам адаптивные решающие правила находят все более широкое применение в приборостроении, в частности, в интеллектуальных сборочных комплексах для адресования деталей на конвейерах.

Стоит также обратить внимание на связь синтезированных решающих правил с фреймами. При этом под фреймом следует понимать, прежде всего, ту минимально необходимую структурированную информацию, которая однозначно определяет данный класс объектов. Наличие фрейма позволяет относить объект к тому классу, который им определяется. Данное определение весьма точно выражает сущность синтезированных адаптивных логических решающих правил минимальной сложности. Поэтому сами эти правила и определяемые ими описания классов можно условно назвать логическими фреймами.

В общем случае фреймы представляют собой совокупность знаний о достаточно сложных объектах и ситуациях. Поэтому они содержат не только локальные сведения о конкретных объектах, которые можно представить с помощью логических описаний в терминах предикатов-признаков, но и знания о возможных действиях и условиях их применимости. Кроме того, фреймы содержат некоторые «дыры», называемые слотами, которые заполняются по мере конкретизации знаний в процессе решения задач.

2.4. Семантические сети

Удобной формой записи фреймов являются семантические сети, дополненные описаниями возможных действий и условий их применимости. Первоначально семантические сети использовались лингвистами для представления смысла текста естественного языка. В дальнейшем появилось много разновидностей таких сетей. Общим для них является то, что вершины сетей соответствуют некоторым объектам, а дуги - отношениям между соответствующими объектами. Иногда вершины представляют сложные отношения между объектами. В этом случае такие вершины соединяются дугами с теми объектами, которые связаны данным отношением. Часто для наглядности на дугах отмечается, какую роль играет каждый объект или отношение [7].

Важным достоинством семантических сетей является то, что представляемые ими знания хорошо поддаются обработке на ЭВМ. Это обеспечивается явным заданием связей между объектами и позволяет расшифровать смысл текста, заданного семантической сетью. Семантическая сеть может описывать класс объектов или ситуаций. В этом случае она имеет вид распознающего или идентифицирующего графа. Однако знания такого рода для интеллектуальной технологической системы носят фрагментарный характер. Они, как правило, недостаточны для организации целенаправленного функционирования производственного комплекса. Тем не менее, эти фрагментарные знания могут с успехом использоваться для синтеза правил поведения интеллектуальной производственной системы типа «класс ситуаций - действие». Такие правила - фреймы позволяют организовать целенаправленное адаптивное поведение интеллектуальной производственной системы в недетерминированной и изменяющейся производственной

обстановке. Например, в технологической системе распознавания и адресования деталей на подвесном конвейере в качестве «класса ситуаций» может использоваться аксиома класса деталей одного типа, а в качестве «действия» - сигнал на приводы стрелок ответвления, обеспечивающий адресование (и доставку) деталей другого класса на соответствующие позиции сборочного конвейера.

Адаптивный интеллектуальный технологический комплекс с элементами искусственного интеллекта должен обладать способностью анализировать окружающую производственную обстановку, распознавать целевые объекты (детали, инструменты и т. п.), идентифицировать препятствия. Для решения этих задач традиционные методы распознавания образов не приспособлены, а иногда и не пригодны.

Главная трудность связана с тем, что производственным комплексам обычно приходится иметь дело не с одним объектом, принадлежащим тому или иному классу, а с совокупностью объектов из разных классов. Другая трудность обусловлена сильной вариативностью объектов в поле зрения интеллектуального технологического комплекса. Так, например, могут сильно изменяться расстояния до окружающих объектов, ракурс, в котором они воспринимаются, условия освещения и т. д. Более того, одни объекты могут заслонять другие, затрудняя тем самым их распознавание. Поэтому от системы управления интеллектуальным технологическим комплексом требуется определенная активность и интеллектуальность, необходимые для преодоления охарактеризованных выше трудностей.

Немногочисленные известные алгоритмы анализа сцен ввиду своей эвристичности далеко не всегда приводят к успеху даже в очень упрощенных и стилизованных условиях распознавания. Значительный интерес для приборостроения представляют метод и системы инвариантного распознавания изображений, использующие в качестве видеодатчиков средства когерентной и некогерентной оптики. Однако и этому методу присущи определенные ограничения и недостатки. К ним относятся требование группового характера преобразований объектов на изображении сцены (что на практике выполняется далеко не всегда) и сложность выделения отдельных объектов путем вычисления их инвариантов. Имеются также хорошо зарекомендовавшие себя эвристические подходы к выделению отдельных объектов на сложной сцене без каких-либо попыток к их распознаванию. Так, в настоящее время существует ряд программ для ЭВМ, позволяющих выделять отдельные объекты на контурном изображении сцены путем предварительной разметки линий и выявления среди них граничных линий на основе анализа типа узлов. Как правило, такие программы используют сложные алгоритмы лингвистического анализа сложных (главным образом, контурных) изображений.

Принципиально иной подход к распознаванию сцен по их изображениям базируется на предварительном (в процессе обучения интеллектуальной системы) формировании понятий в виде аксиом классов с последующим выделением тех или иных объектов сцены посредством логического анализа и распознавания. Этот подход воплощен в рассматриваемом ниже адаптивном

методе логического распознавания сцен, который нашёл широкое применение в современных интеллектуальных технологиях, применяемых при производстве приборов. Этот метод свободен от отмеченных недостатков: он позволяет сравнительно легко и просто с помощью ЭВМ преодолеть возникающие трудности.

Основная идея метода заключается, во-первых, в логическом представлении (в терминах предикатов-признаков) понятий, формируемых в процессе обучения интеллектуального производственного комплекса в виде проблемно-ориентированных аксиом классов, и, во-вторых, в организации процесса распознавания (включая режимы идентификации, классификации и анализа объектов сцены) с помощью алгоритмических средств поиска логического вывода. Метод хорошо приспособлен для реализации на ЭВМ, причем программирование логических алгоритмов формирования понятий и распознавания сцен удобно осуществлять как на языках высокого уровня (типа «С++»), так и на языке логического программирования «ПРОЛОГ». Управляющая система интеллектуального производственного комплекса, реализующая этот метод, обладает важным свойством: она легко дообучается новым понятиям и адаптируется к изменяющимся производственным условиям, что позволяет надежно распознавать на сцене как известные, но сильно преобразованные объекты, так и совершенно новые классы объектов.

В режиме обучения интеллектуальной технологической системе предъявляются в различных ракурсах эталонные объекты (например, детали или инструменты) из разных классов. При этом автоматически строятся описания классов в виде формул исчисления предикатов $A_1(\omega), \dots, A_m(\omega)$, связывающих исходные предикаты-признаки и принимающих истинные значения по меньшей мере на элементах обучающей выборки. Структура, свойства и способы построения таких формул, называемых аксиомами классов, подробно описаны в следующей главе. Здесь остановимся только на одном важном свойстве этих аксиом — их инвариантности по отношению к заданной совокупности преобразований.

Дело в том, что на практике объекты из каждого класса часто отличаются только преобразованиями g из некоторой фиксированной совокупности преобразований G . В других случаях для порождения всех элементов некоторого класса Ω_k может оказаться недостаточно одного эталонного элемента из этого класса и может потребоваться некоторое множество «эталонов», соответствующее, например, существенно различным ракурсам восприятия объектов из данного класса. В подобных случаях естественно потребовать, чтобы интеллектуальный комплекс одинаково классифицировал объекты, отличающиеся допустимыми преобразованиями g из G . Поскольку конкретное преобразование g , действовавшее на распознаваемый объект, обычно неизвестно, то РТК, по существу, является адаптивным по отношению к совокупности возможных преобразований.

При конструировании распознающей системы такого интеллектуального производственного комплекса весьма важно выбрать исходные предикаты-признаки так, чтобы они были инвариантны относительно допустимых

преобразований. Смысл такой инвариантности заключается в том, что предикаты-признаки должны принимать одни и те же значения на объектах, отличающихся друг от друга допустимыми преобразованиями. Однако одной инвариантности признаков мало. Нужно еще, чтобы совокупность инвариантных признаков обладала определенной полнотой, а именно: они должны обеспечить возможность восстановить любое конкретное преобразование g из G , отличающее распознаваемый объект от эталонного.

Важно отметить, что если аксиомы классов строятся на основе инвариантных предикатов-признаков, то они сами также будут инвариантны относительно совокупности допустимых преобразований. Если же система предикатов полна по отношению к совокупности преобразований G , то синтезированные из них аксиомы классов обладают следующими свойствами: во-первых, они дают полное описание классов, инвариантных относительно G ; во-вторых, они позволяют найти неизвестное преобразование g из G , отличающее данный объект от эталонного. Благодаря этому обеспечивается принципиальная возможность безошибочного распознавания классов объектов, инвариантных по отношению к заданной совокупности преобразований при обучении интеллектуального комплекса по выборке минимального объема — по одному эталонному представителю из каждого класса.

В режиме распознавания перед интеллектуальной системой могут возникнуть задачи трех типов: идентификация, классификация и анализ. Цель идентификации заключается в том, чтобы система отыскивала на изображении сцены объект данного класса. Целью классификации является выяснение вопроса: объекты каких классов представлены на изображении сцены. Наконец, цель анализа заключается в выяснении того, объекты каких классов и в каком количестве имеются на изображении сцены, а также в определении их расположения, ориентации и т. п. Задача классификации объектов на рабочей сцене сводится к многократному решению задач идентификации для каждого класса объектов. При этом последовательно выясняется, имеются ли на данной сцене объекты первого, второго и других классов. Задача анализа рабочей сцены также сводится к многократному решению задачи идентификации для всех классов объектов. Отличие ее от задачи классификации заключается в том, что в процессе анализа должны быть идентифицированы и выявлены все объекты, составляющие сцену. Поэтому по мере идентификации того или иного объекта предметные константы, определяющие этот объект, исключаются. Одновременно из описания сцены вычеркиваются все предикаты, которые входят в описание идентифицированного объекта. Эта операция сводится к «стиранию» изображения идентифицированного объекта из оперативной памяти, где хранится изображение рабочей сцены.

Для оставшейся части анализируемой сцены вновь решается задача идентификации, пока не будет получен ответ, что объектов данного класса на сцене нет. Тогда описанная процедура идентификации повторяется для нового класса объектов. В результате такого логического анализа выделяются все объекты, составляющие рабочую сцену, и указывается: к какому классу каждой из них принадлежит, где расположен и как ориентирован.

Таким образом, в рамках описанного метода разделение рабочей сцены на отдельные объекты разных классов происходит в процессе их распознавания. В этом заключается одно из принципиальных отличий и преимуществ данного метода от других известных, которые сводятся либо только к разбиению сцены на объекты (без каких-либо попыток к их распознаванию), либо к распознаванию заранее выделенных объектов. Благодаря этому, именно данный метод нашёл широкое применение в интеллектуальных производственных системах, которые используются при производстве приборов и устройств.

Общая схема описанной адаптивной системы логического распознавания рабочих сцен представлена на рис. 2.3.

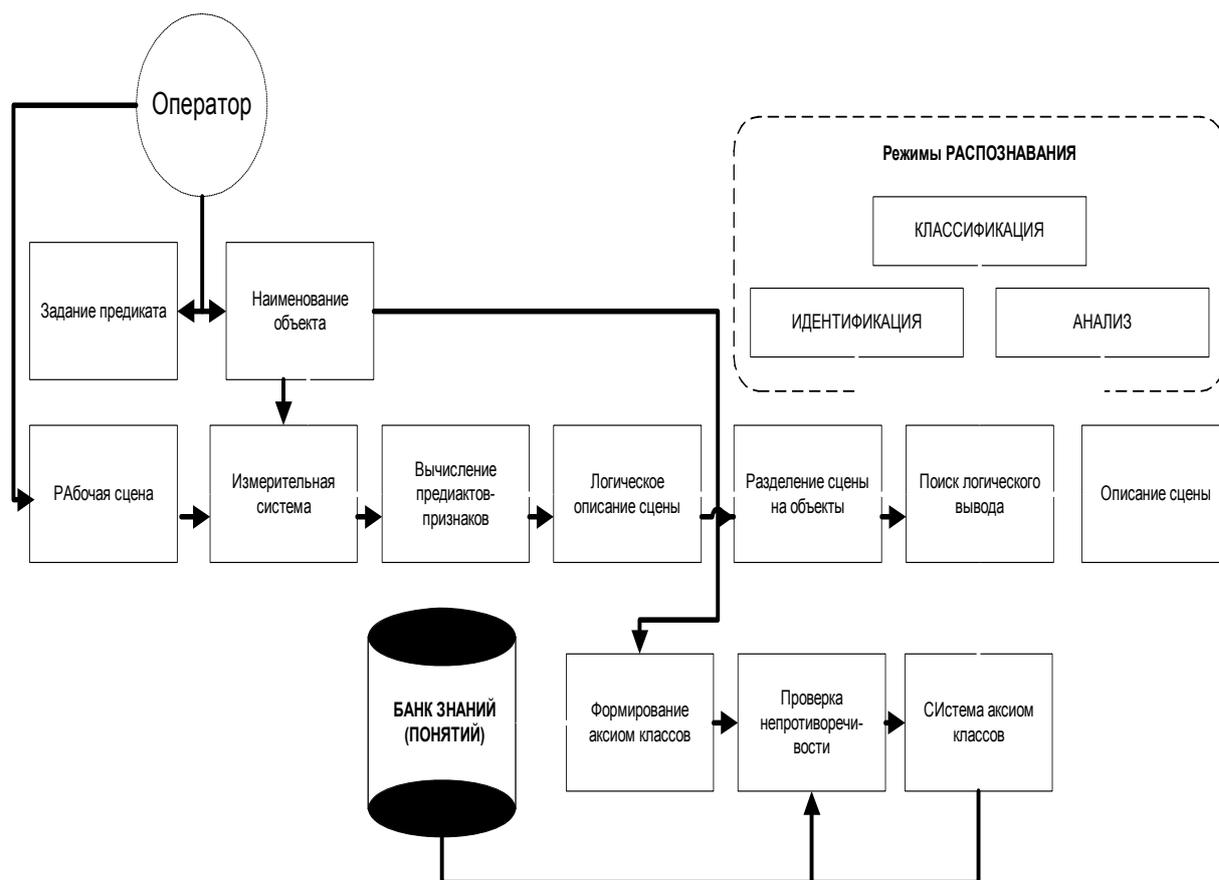


Рис. 2.3 Адаптивная система логического распознавания сцен

Здесь штриховыми стрелками обозначены каналы передачи информации в режиме обучения РТК, а сплошными - каналы передачи информации в режиме распознавания, т. е., в процессе идентификации, классификации или анализа сцены.

Продemonстрируем особенности логического метода распознавания на примере решения задачи автоматического анализа технологических сцен по их контурным изображениям [16]. Изображение ω будем задавать матрицей координат вершин на плоскости изображения и матрицей связности, элементы которой определяются следующим образом: $m_{ij} = 1$, если из i -й вершины выходит ребро, оканчивающееся в j -й вершине, и $m_{ij} = 0$ - в противном случае.

Таким образом, если на изображении сцены имеется N вершин, то она задается $N^2 \neq 2N$ числами. Для простоты изложения метода ограничимся случаем, когда объекты, составляющие рабочую сцену, являются многогранниками. При указанных ограничениях на изображении сцены могут встретиться вершины только следующих четырех типов:

- 1) вершины типа V (вершины, у которых на изображении не видно одно из образующих ребер);
- 2) вершины типа Y (вершины, у которых любые два ребра лежат по разные стороны от прямой, образованной третьим ребром);
- 3) вершины типа W (вершины, у которых все три образующих ребра лежат в одной полуплоскости);
- 4) вершины типа T (вершины, образованные пересечением проекций ребер многогранников).

Определим исходные предикаты-признаки таким образом, чтобы каждой вершине на изображении сцены соответствовал предикат, указывающий ее тип. Такие предикаты задаются соотношениями, представленными на рис. 2.4.

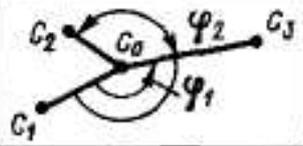
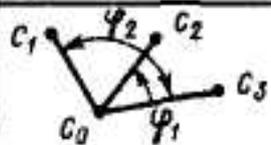
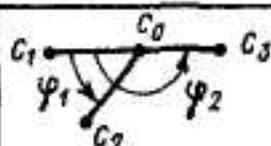
Тип вершины	Изображение вершины	Описание предиката
V		$V(C_0, C_1, C_2) = 1$ $\varphi < \pi$
Y		$Y(C_0, C_1, C_2, C_3) = 1$ $\varphi_1 < \varphi_2 < \pi$
W		$W(C_0, C_1, C_2, C_3) = 1$ $\varphi_1 < \varphi_2 < \pi$
T		$T(C_0, C_1, C_2, C_3) = 1$ $\varphi_1 < \varphi_2 < \pi$

Рис. 2.4 Исходные предикаты контурных изображений

По матрице координат и матрице связности легко определить тип каждой вершины и порядок аргументов в соответствующем предикате.

Рассмотрим для определенности сцены, состоящие из объектов четырех классов, изображения которых в различных характерных ракурсах представлены на рис. 2.5. В режиме обучения интеллектуальной производственной системе предъявляются отдельные объекты с указанием, к какому классу они относятся. По каждому изображению ω_i объекта из k -го класса автоматически строится его логическое описание $z_i(\omega)$ (аксиома k -го

подкласса). Так как выбранная система предикатов полна (в смысле однозначности проекций вершин многогранников), то нет необходимости выписывать все предикаты для каждой вершины изображения, — достаточно вычислить предикат, истинный для данной вершины. Этим обеспечивается

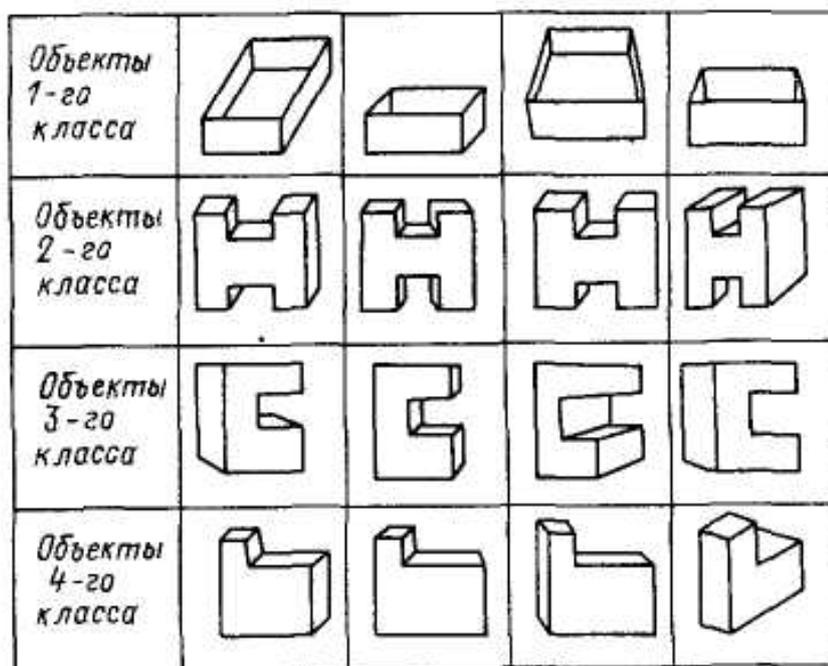


Рис. 2.5 Изображение объектов из обучающей выборки

совпадение ранга конъюнкции логического описания объекта с числом вершин на его изображении.

Аксиомы классов строятся по обучающей выборке. Поставим перед РТК задачу распознавания сложной сцены ω , изображенной на рис. 2.6.

Согласно дереву логического вывода, представленному на рис. 2.7, получаются следующие результаты: на изображении сцены $z(\omega)$ имеется объект 1-го класса, причем система подстановок в процессе распознавания позволяет однозначно восстановить изображение идентифицированного объекта. Решая задачу классификации объектов рабочей сцены на рис. 2.6, интеллектуальный производственный комплекс выделяет четыре класса объектов. В результате анализа сцены интеллектуальный комплекс легко выделит и распознает все незаслоненные объекты. Для распознавания же заслоненных объектов нужны специальные методы.

Таким образом, описанные средства логического распознавания позволяют интеллектуальному производственному комплексу осуществлять автоматический анализ сложных рабочих сцен по их контурным изображениям. Они применимы также и в тех случаях, когда изображения объектов и сцен заданы в виде матрицы яркости. При этом достаточно в качестве исходных предикатов-признаков взять трехместные предикаты, первыми аргументами которых являются соответствующие номера столбца и строки матрицы яркости, а третьим - градация яркости данного элемента изображения. Использование и программная реализация на управляющей ЭВМ алгоритмов логического

распознавания позволяют существенно расширить функциональные (интеллектуальные) возможности адаптивных интеллектуальных комплексов. Острая практическая потребность в таких производственных комплексах с элементами искусственного интеллекта возникает сегодня во многих отраслях промышленности, в том числе и в приборостроении.

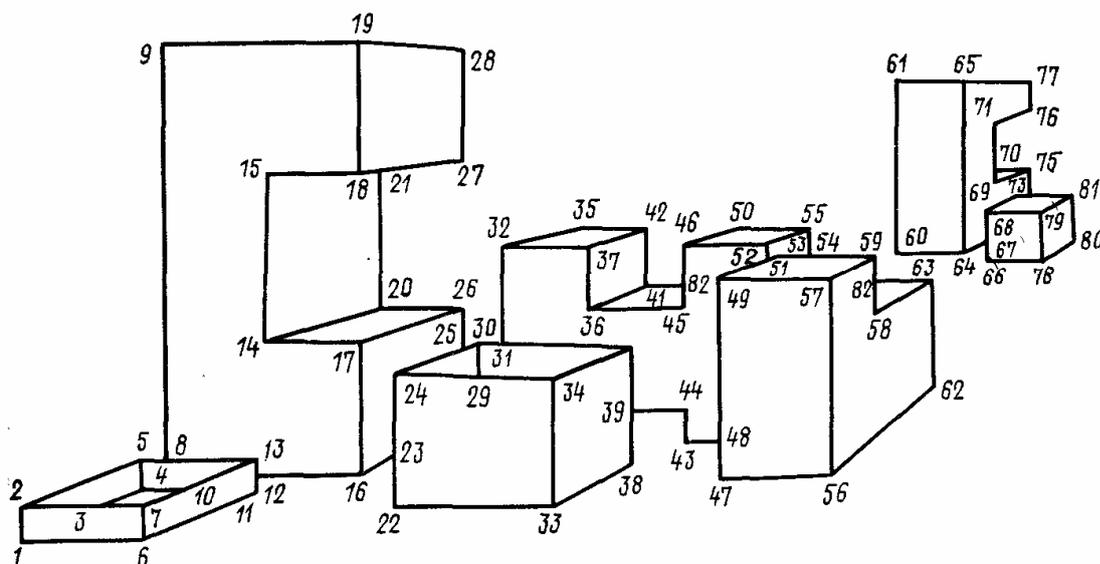


Рис. 2.6 Изображение сложной сцены и её кодирование

Важным элементом искусственного интеллекта для интеллектуальной производственной системы являются системы искусственного зрения и распознавания. Эти системы состоят из датчиков видеосигналов и средств распознавания. В качестве видеодатчиков обычно используются фотодиоды, телекамеры или лазеры. Средства распознавания объектов по видеосигналам реализуются в виде программного обеспечения для специализированной микроЭВМ или микропроцессоров.

Системы искусственного зрения и распознавания, используемые на производстве, принято называть системами технического зрения. При этом техническое зрение трактуется как собственно процесс восприятия окружающих объектов с помощью видеодатчиков, так и их распознавание на основе полученной видеоинформации.

Одной из главных причин «интеллектуализации» производства с помощью систем технического зрения является необходимость автоматизации широкого класса технологических операций, требующих визуального контроля. Примерами таких операций могут служить сборка и сварка изделий. Другая не менее важная причина заключается в известной неупорядоченности и нестационарности рабочей обстановки, характерной для многих типов производств. В такой обстановке обычные технологические комплексы с программным управлением теряют работоспособность.

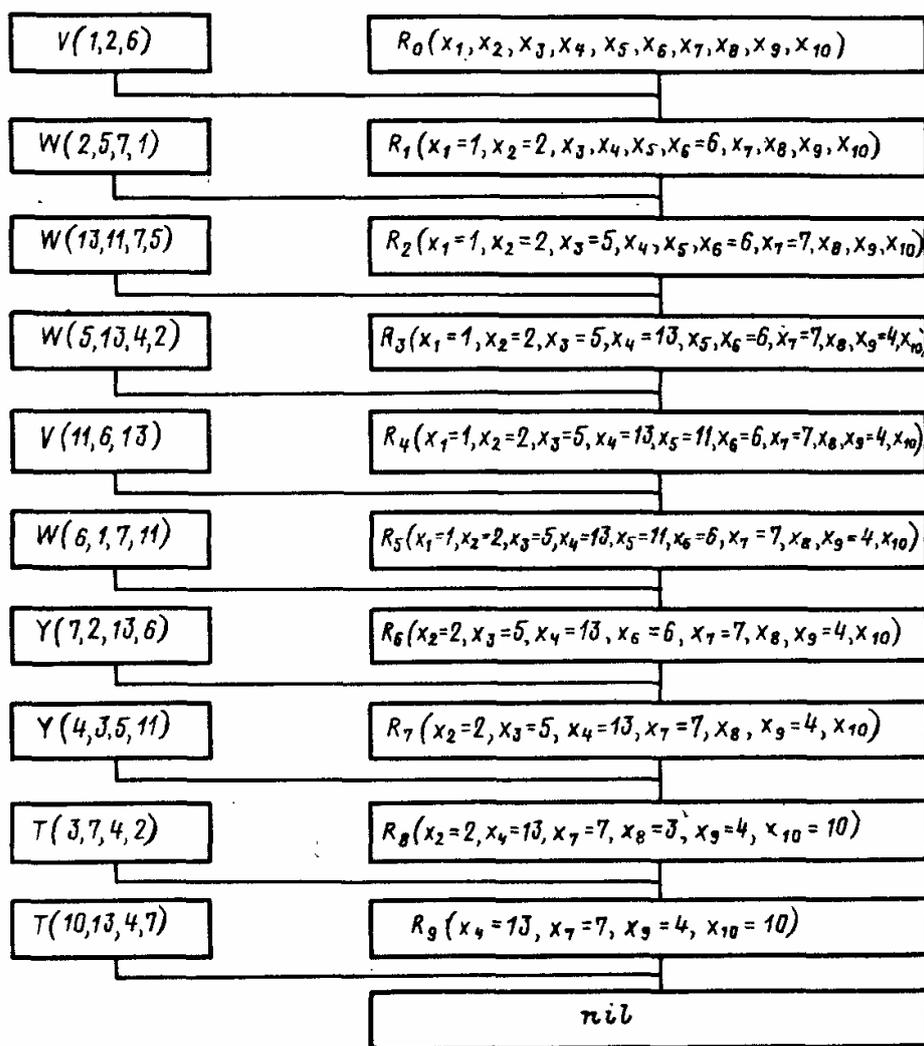


Рис. 2.7 Процесс логического распознавания

Организация и поддержание идеальных условий требуют изготовления специальной технологической оснастки для предварительного упорядочивания обстановки в рабочей зоне (точное позиционирование и ориентирование деталей и т. п.). Это усложняет процесс роботизации производства, приводит к значительным дополнительным затратам и существенно снижает экономический эффект от внедрения интеллектуальных производственных технологий. Тем не менее, в условиях массового и крупносерийного производства соответствующие затраты оправданы. Однако в производственных комплексах, ориентированных на выпуск продукции малыми сериями, предварительное упорядочивание рабочей среды зачастую не только экономически неоправданно, но и может оказаться невозможным. В этих условиях весьма перспективно использование интеллектуальных производственных комплексов с системой технического зрения, поскольку они могут работать с неориентированными и произвольно расположенными деталями, адаптироваться к непредсказуемым изменениям производственных условий.

В случае необходимости манипуляционные и транспортные роботы с

адаптивным управлением и системой технического зрения могут применяться для предварительного упорядочивания рабочей среды (например, для складирования неориентированных деталей в специальные пакеты и т. п.).

Таким образом, интеллектуальные производственные комплексы, снабженные системой технического зрения, могут служить как для автоматизации технологических процессов в недетерминированной и изменяющейся производственной обстановке, так и для предварительного упорядочивания объектов в рабочей зоне.

Разработка системы технического зрения включает в себя ряд этапов, в том числе: выбор подходящих видеодатчиков с учетом условий освещенности в рабочей зоне; оценку информативности снимаемых признаков; синтез решающих правил и их реализацию на базе современных средств вычислительной техники. Системы технического зрения работают в двух режимах: обучения и распознавания. В режиме обучения системе предъявляют объекты разных классов (например, детали и инструмент) в характерных рабочих ракурсах. По этим данным автоматически строятся описания классов и решающие правила, которые хранятся в базе знаний. В режиме распознавания осуществляется (в зависимости от целевых условий) идентификация нужного объекта, классификация видимых объектов или анализ, описание и интерпретация рабочей обстановки. Полученная таким образом информация используется далее для адаптивного управления роботами и другим технологическим оборудованием интеллектуального производственного комплекса.

Системы технического зрения применяются в приборостроении для выполнения следующих технологических операций:

- 1) Загрузки станков и обрабатывающих центров произвольно расположенными заготовками и деталями из бункеров или с конвейера;
- 2) Визуального контроля и отбраковки деталей или изделий;
- 3) Визуального контроля положения инструмента при механической обработке заготовки;
- 4) Сортировки и укладки в пакеты или специальные накопители неупорядоченных деталей (в том числе деталей, поступающих в таре);
- 5) Комплектации узлов и сборки изделий из неупорядоченных деталей;
- 6) Дуговой сварки изделий сложной конфигурации;
- 7) Окраски деталей сложной формы;
- 8) Адресования деталей с подвесных конвейеров на сборочные и т.д.

Разнообразие технологических процессов, требующих визуального контроля и распознавания, привело к созданию большого числа адаптивных интеллектуальных технологических комплексов, отличающихся компоновкой, типом применяемых видеодатчиков, методами и средствами распознавания. В качестве методов автоматического распознавания используются логические, логико-вероятностные, инвариантные и дискриминантные (перцепторные) алгоритмы и решающие правила. Эти методы реализуются либо программно на миниЭВМ или микропроцессорах, либо аппаратно на интегральных

микросхемах.

Компоновка производственного комплекса зависит в значительной мере от числа и способа включения в него систем технического зрения.

В большинстве интеллектуальных производственных комплексах восприятие и анализ обстановки в рабочей зоне производится с помощью одной или нескольких систем технического зрения, непосредственно не связанных с роботами. Такая компоновка обеспечивает естественное распознавание процессов обработки видеоинформации и адаптивного управления роботами и другим технологическим оборудованием. Кроме того, она позволяет воспринимать рабочую зону в целом и непосредственно контролировать происходящие изменения.

Недостатком данной компоновки является то, что в момент захвата деталей манипуляторы роботов заслоняют именно ту часть рабочей зоны, информация о которой больше всего нужна для качественного выполнения технологических операций (например, для сборки). Поэтому, иногда системы технического зрения встраивают в роботизированные элементы производственного комплекса, устанавливая видеодатчик на манипуляторе (компоновка «глаз на руке») или прямо в механизм захвата (компоновка «глаз в руке»). Преимущество такой компоновки заключается в том, что манипуляторы роботов теперь уже не перекрывают поле зрения системы технического зрения. Кроме того, облегчается самонаведение захвата на деталь, причем процесс наведения становится адаптивным (за счет использования визуальной обратной связи). Однако при этом информация о рабочей зоне носит локальный характер. На практике, при выборе рациональной компоновки производственного комплекса целесообразно комбинировать оба способа установки системы технического зрения.

Рассмотрим теперь общие принципы использования обратной связи через систему технического зрения для организации адаптивного управления интеллектуальным производственным комплексом.

Наибольшее распространение получил принцип эпизодического обращения к системе технического зрения, когда необходимая для адаптивного управления комплексом видеоинформация поступает дискретными порциями (кадрами) лишь в определенные моменты времени. Эта информация обычно используется для формирования целевых условий и ограничений, а также для автоматического расчета или коррекции программных движений роботов и технологического оборудования. В процессе же фактической отработки программных движений производственный комплекс функционирует «вслепую».

Значительный интерес для улучшения качества адаптивного управления представляет принцип непрерывного использования обратной связи от системы технического зрения. Этот принцип лежит в основе нового класса адаптивных систем управления — систем «визуального сервоуправления» и «визуального самонаведения». Они все шире применяются при сборке сложных изделий, при слежении сварочной головкой за швом, при наведении измерительного щупа на контрольные точки детали, при взятии движущихся по конвейеру объектов и т.

п. В этих случаях обычно используется компоновка системы технического зрения типа «глаз в руке». Целевые условия формулируются в терминах желаемого расположения захвата или рабочего органа в системе координат «глаза». «Визуальное сервоуправление» сводится к устранению рассогласования между целевыми и фактическими характеристиками на основании показаний системы технического зрения.

Робототехнические системы технического зрения существенно отличаются от обычных, используемых, например, в читающих автоматах, в системах анализа аэрофотоснимков, морфологических срезов, отпечатков пальцев и т. п. Эти отличия обусловлены, прежде всего, способностью производственных комплексов активно воздействовать как на объекты в рабочей зоне, так и на саму систему технического зрения (изменение ракурса восприятия), подвижностью воспринимаемых объектов, а также необходимостью оперативной обработки информации для адаптивного управления производственным комплексом без замедления работы оборудования.

Особенности систем технического зрения, используемых в интеллектуальных производственных комплексах, порождают специфические требования, которые необходимо учитывать при их разработке или выборе.

Среди наиболее важных требований отметим следующие:

1) высокая скорость обработки видеoinформации (в ряде робототехнических систем технического зрения время распознавания одной детали составляет десятки или сотни миллисекунд);

2) гибкость и обучаемость, т. е. способность переобучаться для распознавания новых классов объектов;

3) высокая точность распознавания и идентификации геометрических параметров объектов (координаты, ориентация и т. п.);

4) инвариантность к различным возмущениям (изменение освещения, контрастности, ракурса и т. п.) и высокая помехоустойчивость в производственных условиях.

Кроме того, робототехнические системы технического зрения должны легко сопрягаться со стандартным интерфейсом управляющих ЭВМ, иметь малые габаритные размеры и низкую стоимость.

На рис. 2.8 представлен самообучающийся видео датчик ICS 100 немецкой фирмы SICK, специализирующейся на производстве различного рода датчиков и систем технического зрения для промышленности.

Камера системы технического зрения имеет гибкую настройку с функциями изменения «обучаемой» области, поиска области и другими средствами управления. Индикация «баланса серого» и других параметров позволяет добиться высокой точности настроек.



Рис. 2.8 Самообучающийся видео датчик системы технического зрения ICS 100

Задача сортировки деталей, связанная с частой сменой объекта слежения, может быть решена при помощи функции «одновременного обнаружения». Данная функция позволяет одновременно распознавать до 4-х различных объектов, включая объекты с высокой отражающей способностью.

3. Методы поиска решений технологических задач и системы представления знаний

3.1 Фреймы

В общем случае фреймы представляют собой совокупность знаний о достаточно сложных объектах и ситуациях. Модель представления знаний на основе фреймов использует концепцию организации памяти, понимания и обучения человека (рис. 3.1).

Фрейм (рамка) – это единица представления знаний, детали которой могут изменяться в соответствии с текущей ситуацией. Фрейм в любой момент может быть дополнен различной информацией, касающейся способов применения данного фрейма, последствий этого применения и т.д.

Поэтому фреймы содержат не только локальные сведения о конкретных объектах, которые можно представить с помощью логических описаний в терминах предикатов-признаков, но и знания о возможных действиях и условиях их применимости.

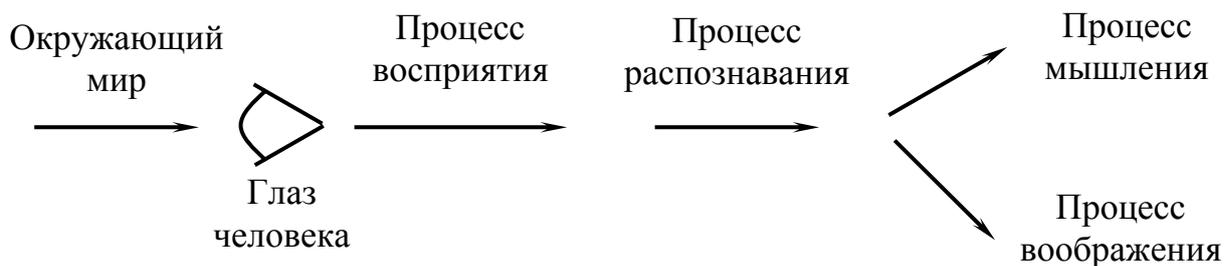


Рис. 3.1 Структурная схема процесса мировосприятия человеком

В основе фреймовой модели представления знаний систем искусственного интеллекта лежат два основных допущения.

1. Ощущения, представления, моделирование и мышление человека приводится в движение ожиданием.

При названии определения в мозгу тот час вырабатываются ассоциации о представлении и свойствах названного объекта;

2. Существует иерархия, в соответствии с которой свойства могут наследоваться.

Свойства априори приписываются объекту. Только специфические и аномальные свойства требуют отдельного упоминания.

Пример: собака (рис. 3.2).

Ожидание:

зверь, 4 лапы, морда, хвост, лает, может укусить, хорошее чутье.

Специфическое (специфическим является ниже приведенное фотографическое изображение собаки, представленное на рис. 3.2):



Рис. 3.2 Собака.

Типичная структура фрейма представлена на рис.3.3.

Имя фрейма
Надфрейм
Подфрейм
Имя слота1: значение слота1 признак11: значение11 ... признак n1: значение n1
... ..
Имя слота m: значение слота m признак 1m: значение 1m ... признак nm: значение nm

Рис. 3.3 Типичная структура фрейма

Каждый из фреймов имеет надфрейм и подфрейм, за счет чего реализуется иерархия наследования свойств. Надфрейм описывает наиболее общие свойства, относящиеся к группе объектов. Подфрейм конкретизирует частные свойства объекта или его признака. Признаки объектов записываются в значении слотов, каждому из которых присваивается еще и имя. За счет существования иерархии наследования свойств каждый фрейм описывает только определяющие свойства групп объектов или конкретного объекта.

Каждая из интеллектуальных систем, использующая в качестве модели представления знаний фреймы имеет как заполненные фреймы или иначе

экзофреймы, которые вносятся в исходную базу данных, так и пустые фреймы - портофреймы, значение и имена слотов которых заполняются по мере использования системы и решения конкретных задач. Фрейм можно представить в виде сети, верхние узлы которой четко определены, так как всегда справедливы для предполагаемой задачи.

На рис. 3.4 представлен фрейм и его подфрейм.

Фреймы и их содержание может быть представлено с помощью языка исчисления предикатов:

- фрейм(имя_фрейма , список_надфреймов, список_подфреймов)
- слот(имя_фрейма , имя_слота, значение_слота)
- признак(имя_фрейма , имя_слота , признак , значение)

<p>Легковой автомобиль Надфрейм: <i>автомобили</i> Подфреймы: <i>кабриолет, лимузин, вен</i> Мотор: горючее топливо Топливо: default <i>Бензин</i> количество цилиндров: ≥ 2 объем: $> 1000 \text{ cm}^3$ Привод: default <i>коробка скоростей</i> количество передач: default 5 количество ведущих колес: default 2 Количество колес: default 4</p>	<p>Лимузин Надфрейм: <i>легковой автомобиль</i> Подфреймы: <i>компактен, средний класс, ...</i> Общая масса: $\geq 2 \text{ t}$ Кол-во мест: default 5 Количество колес: default 4</p>
---	---

Рис. 3.4 Фрейм и его подфрейм (default – значение параметра по умолчанию).

Система вывода для фреймов организована с помощью родовидных связей или иерархической структуры наследования.

Система фреймов (для описания конкретной предметной области) представляет собой иерархическую структуру, в которой фреймы группируются с помощью родовидных связей.

На верхнем уровне иерархии находится фрейм, содержащий наиболее общую информацию, истинную для всех остальных фреймов.

Фреймы наследуют значения характеристик своих родителей, находящихся на более высоком уровне иерархии.

Некоторым ячейкам или значениям класса можно присвоить значение по умолчанию, которое наследуются через иерархию: класс - подкласс и класс - член. Значения по умолчанию используются, если нет другой информации.

Такая структура наследования позволяет учитывать различного рода исключения.

Ячейки фрейма содержат, как правило, следующую информацию.

1. Данные для идентификации фрейма (имя фрейма);
2. Взаимосвязь этого фрейма с другими фреймами (надфреймы и подфреймы);
3. Дескрипторы требований для фрейма (пример: стул- высота сидения 40см, высота спинки стула - 60 см.) для определения соответствия новых объектов стереотипу фрейма;
4. Процедуральная информация об использовании описанной структуры, представляющая возможность присоединять к ячейке процедуральный код;
5. Информация по умолчанию. Значение ячейки, которое должно быть истинным, если не найдены противоположные;
6. Информация для нового экземпляра. Незаполненные слоты до указания их значения для отдельного экземпляра или отдельного аспекта решения проблемы.

Слот может содержать не только конкретное значение, но и имя процедуры, позволяющей вычислить его по заданному алгоритму, а также одну или несколько эвристик, с помощью которых это значение определяется. В слот может входить не одно, а несколько значений.

Иногда этот слот включает в себя компонент, называемый фасетом который задает диапазон или перечень его возможных значений. Фасет указывает также граничное заполнение значения слота. Фасеты - это дескрипторы на уровне слотов.

Среди процедур могут быть выделены процедуры - демоны и процедуры - слуги .

Процедуры - демоны запускаются автоматически при выполнении некоторого условия (побочный эффект некоторого другого действия в БЗ).

Процедуры - слуги активизируются только по специальному запросу. Наиболее распространенные области применения данной модели представления знаний для систем искусственного интеллекта представлены на рис. 3.5.



Рис.3.5 Наиболее популярные области применения фреймовой, как модели представления знаний.

К достоинствам фреймовой модели следует отнести следующее:

- простота обращения и обработки;
- заполнение фреймов очень просто;
- простота вывода
(выводы на основании знаний о вышестоящих концепциях соответствуют человеческому мышлению);
- позволяют манипулировать как декларативными, так и процедурными знаниями.

К недостатка относят:

- не существует формальной семантики;
- не существует общепринятой методы исчислений т.е., обработка должна быть приспособлена к случаю применения;
- малый потенциал вывода(выводы базируются только на знаниях о иерархических отношениях).

3.2 Семантические и когнитивные сети

Удобной формой записи фреймов являются семантические сети (СС), дополненные описаниями возможных действий и условий их применимости. Первоначально семантические сети использовались лингвистами для представления смысла текста естественного языка. В дальнейшем появилось много разновидностей таких сетей [12].

Общим для них является то, что вершины сетей соответствуют

некоторым объектам, понятиям, фактам и событиям, а дуги - отношениям между соответствующими объектами. Как узлы, так и связи обычно имеют метки. Сам по себе термин Семантическая сеть обозначает свойство представлений, основанных на графах (пример СС рис. 3.6). Иногда вершины представляют сложные отношения между объектами. В этом случае такие вершины соединяются дугами с теми объектами, которые связаны данным отношением. Часто для наглядности на дугах отмечается, какую роль играет каждый объект или отношение.

Выводы в СС делаются путем прослеживания связей с соответствующими понятиями. Основным способом вывода в семантической сети является сопоставление частей сетевой структуры.

Сущность его состоит в следующем: содержание и цель запроса к семантической сети списывается автономной семантической сетью (сетью запроса), построенной по тем же правилам и отображающей те же объекты и отношения, которые представлены семантической сетью. Поиск ответа на запрос реализуется сопоставлением сети запроса с фрагментами семантической сети базы знаний.

Наиболее распространенными типами отношений объектов, описываемых с помощью модели СС являются:

-БЫТЬ ЭЛЕМЕНТОМ КЛАССА (ЯВЛЯТЬСЯ) - означает, что объект входит в состав данного класса;

-ИМЕТЬ - позволяет задавать свойства объектов, например, жираф имеет длинную шею;

-ЯВЛЯТЬСЯ СЛЕДСТВИЕМ - отражает причинно-следственные связи;

ИМЕТЬ ЗНАЧЕНИЕ - задает значение свойств объектов, например, пациент может иметь двух братьев.

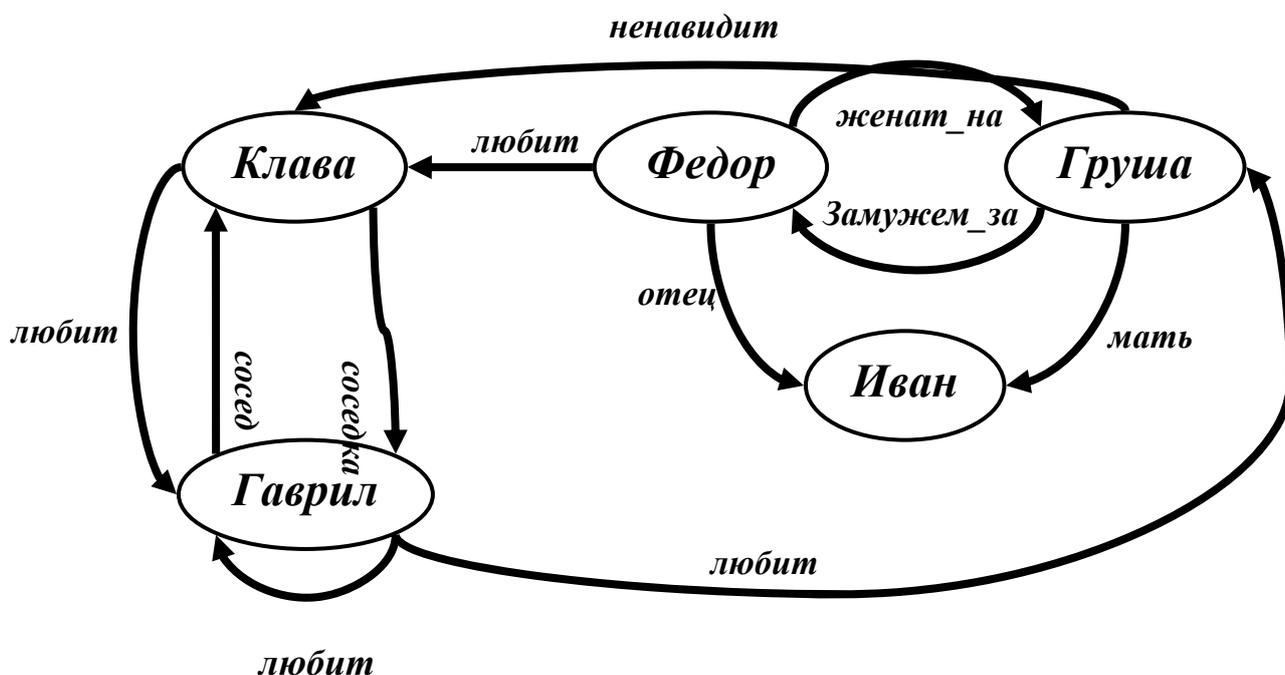


Рис. 3.6 Пример семантической сети

Этим четырем, наиболее распространенным видам отношений соответствуют следующие свойства или метазнания:

- отношения **is_a (a kind_of)** являются транзитивными :
- $\forall \forall X \forall Y \forall Z (is_a(X,Z) \leftarrow is_a(X,Y) \wedge is_a(Y,Z));$
- для отношения \in (элемент чего-то) и **is_a** является действительным:
- $\forall \forall X \forall Y \forall Z (is_a(Z,X) \leftarrow is_a(Y,X) \in \wedge (Z,Y));$
- для отношений типа **part_of** (является частью), **has** (имеет) и **is_a** действительно:
- $\forall \forall X \forall Y (part_of(X,Y) \leftrightarrow has(Y,X));$
- $\forall \forall X \forall Y \forall Z (has(X,Z) \leftarrow is_a(X,Y) \wedge has(Y,Z)).$

Другой способ осуществления вывода- это использование приведенных выше метазнаний или свойств.

СС могут быть представлены с помощью логической модели в соответствии с шаблоном, представленным на рис.3.7.

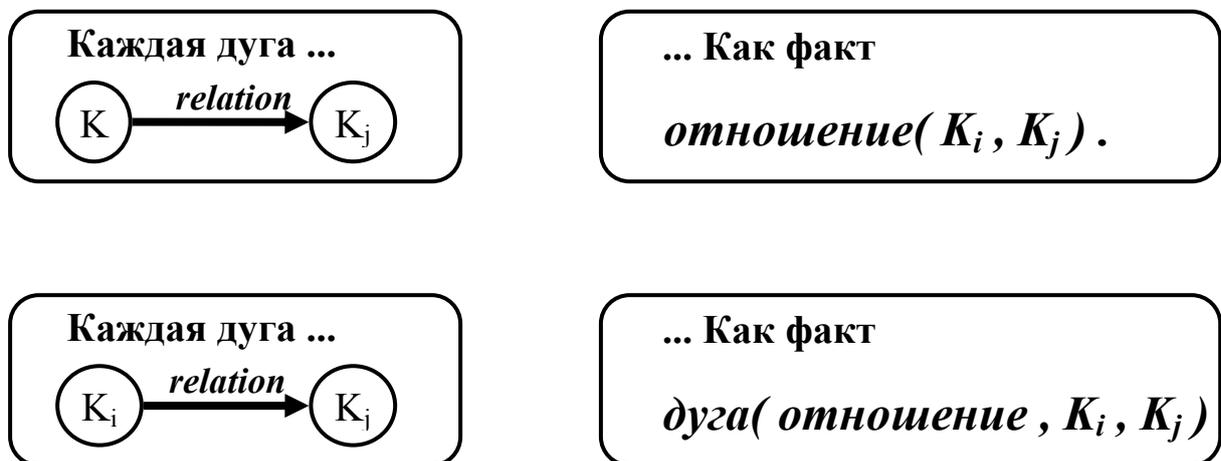


Рис. 3.7 Представление СС с помощью языка исчисления предикатов.

Оба варианта реализации семантических сетей являются «частями языка» PROLOG (2- и 3-арные факты без переменных).

PROLOG это „часть языка“ ИП1 (клаузальные формы Хорна).

Важным достоинством семантических сетей является то, что представляемые ими знания хорошо поддаются обработке на ЭВМ. Это обеспечивается явным заданием связей между объектами и позволяет расшифровать смысл текста, заданного семантической сетью. Семантическая сеть может описывать класс объектов или ситуаций. Наибольшее распространение в интеллектуальных технических системах получили не СС, как токовые, а их более развитый вариант концептуальные графы или когнитивные сети. Они отличаются от СС наличием двух основных дополнений:

Дополнение 1

Цель: Представление отношений любой арности для повышения выразительности, так как по своей выразительности СС уступают языку исчисления предикатов.

Идея: Отношения и арность будут записываться в качестве узлов

Математическое представление: предикат арности 1; предикаты арности более 1 (рис.3.8).

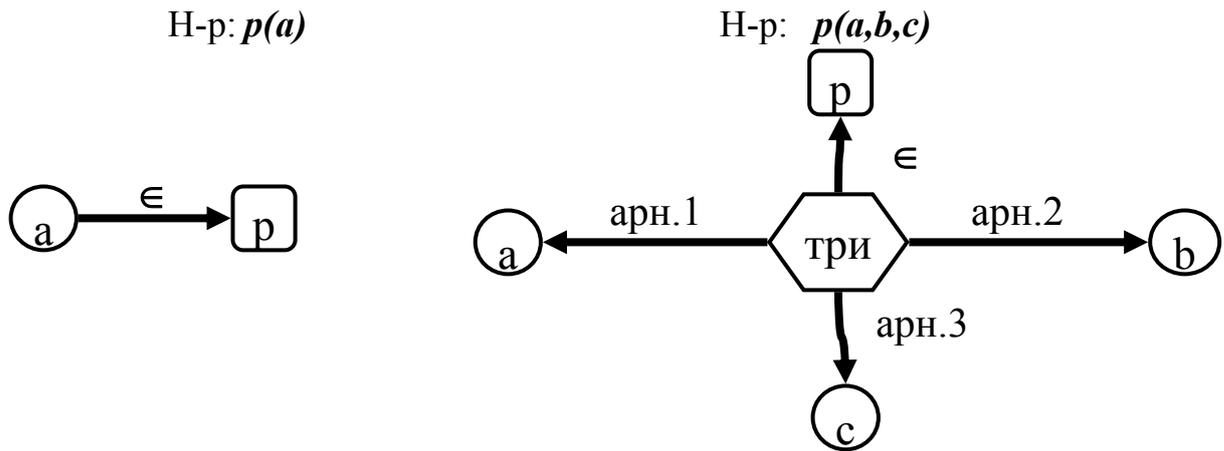


Рис. 3.8 Схема математического представления.

Дополнение 2: типы, экземпляры, имена.

Цель: определение отношений класс - элемент и класс - подкласс.

В СС из-за отсутствия такого определения возникает семантическая неоднозначность. Так как какие-то свойства объектов могут принадлежать экземплярам, а другие классам.

Идея: расставление меток.

Каждый прямоугольник понятия снабжается меткой конкретного типа, определяющей тип или класс экземпляра, представленного этим узлом. Таким образом, узел, снабженный меткой «собака», представляет некоторый объект этого типа. Типы имеют иерархическую структуру. Прямоугольник с одной и той же меткой типа отражает понятие одного и того же типа. Однако эти прямоугольники могут либо представлять, либо не представлять одно и тоже индивидуальное понятие.



Рис. 3.9 Пример концептуального графа

Концептуальные графы позволяют также описывать конкретные, но не именованные экземпляры.

Для обозначения каждого экземпляра в мире рассуждений используется уникальный дескриптор, называемый маркером «#». Маркеры отличаются от имен тем, что они являются уникальными: экземпляр может иметь одно имя, несколько имен или вообще являться безымянным, но он имеет право только на один маркер. Это различие позволяет преодолевать семантические неоднозначности, возникающие при именовании объектов.



Рис. 3.10 Пример использования маркера.

Маркеры позволяют отделять экземпляры от их имен. Если собака :#23 имеет имя Ася, то для его описания можно использовать концептуальное отношение «имя». Имя, заключенное в кавычки указывает, что это строка. Там где возникает опасность неоднозначности, граф может быть упрощен за счет ссылки прямо на имя (рис. 3.11).

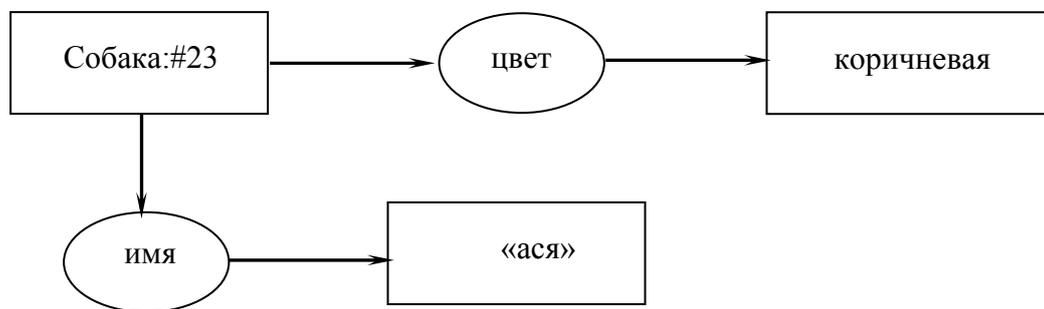


Рис. 3.11 Пример использования маркеров и отделения экземпляров от имен.

Вместо обращения к объекту по маркеру или по имени можно использовать обобщенный маркер «*» для определения не специфицированных экземпляров. Эта информация часто опускается в метках понятий: понятия «собака» и «собака: *» считаются эквивалентными.

Дополнительно к обобщенному маркеру КГ позволяют использовать именованные переменные. Они представляются звездочкой, за которой следует имя переменной. Это полезно, если два отдельных узла должны обозначать один и тот же, но не указанный экземпляр.

Итак, каждый узел понятия означает экземпляр определенного типа. Это объект, относящийся к данному понятию, или объект ссылки. Ссылка задается либо индивидуально, либо для всех экземпляров в целом. Если для

объекта ссылки используется маркер экземпляра, понятие является конкретным, а если общий маркер- понятие считается родовым.

Дополнение 3 (рис. 3.12).

Цель: представление иерархии типов.

Идея: введение абстрактных уровней.

Математическое представление: узлы одной сети могут представлять собой подсети.

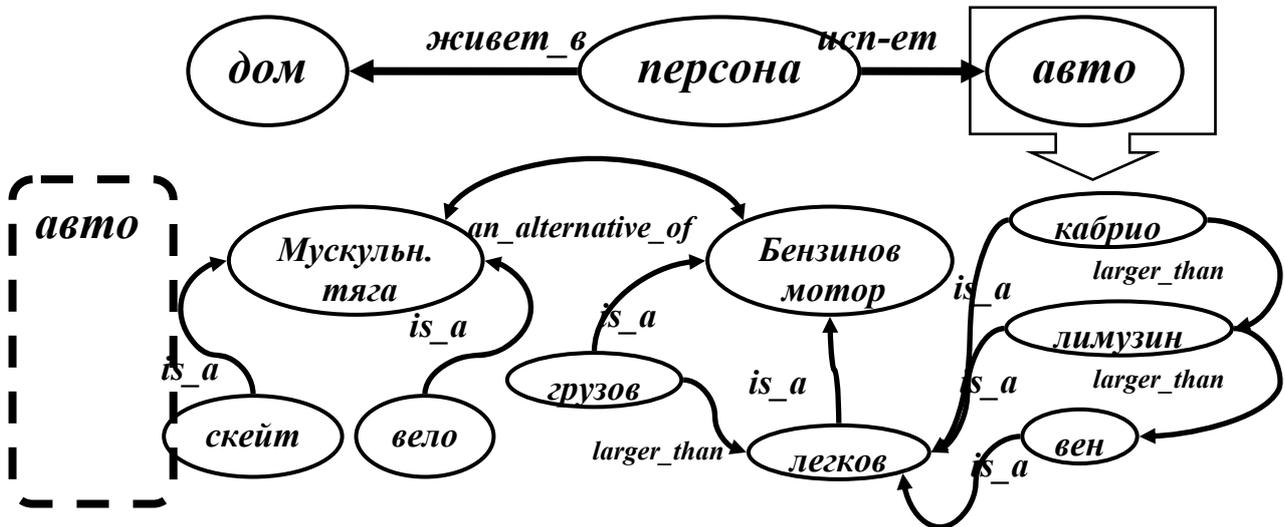


Рис. 3.12 Пример представления иерархии типов.

При этом подуровень «авто» представляет собой высказывания или пропозициональные понятия и обозначается прямоугольником, содержащим концептуальный граф. Узлы внутри такого уровня будут называться пропозициональными.

Дополнение 4: иерархия типов (рис.3.13).

Иерархия типов- это частное упорядочение на множестве типов, которое можно обозначить множеством \leq .

Если t и s – типы и $t \leq s$, то t – это подтип s , а s -это супертип t . Тип может иметь один или несколько супертипов, а также один или несколько подтипов.

Если t , s и u – типы, причем $t \leq s$ и $t \leq u$, то говорят, что t - это общий подтип для s и u .

Каждая пара типов должна иметь минимальный общий супертип и максимальный общий подтип. Тип v является минимальным общим супертипом для типов s и u , если $s \leq v$ и $u \leq v$ и для любого общего супертипа w типов s и u и $v \leq w$.

Универсальный тип- супертип всех типов, обозначается \top . Абсурдный тип подтип всех типов, обозначается \perp .

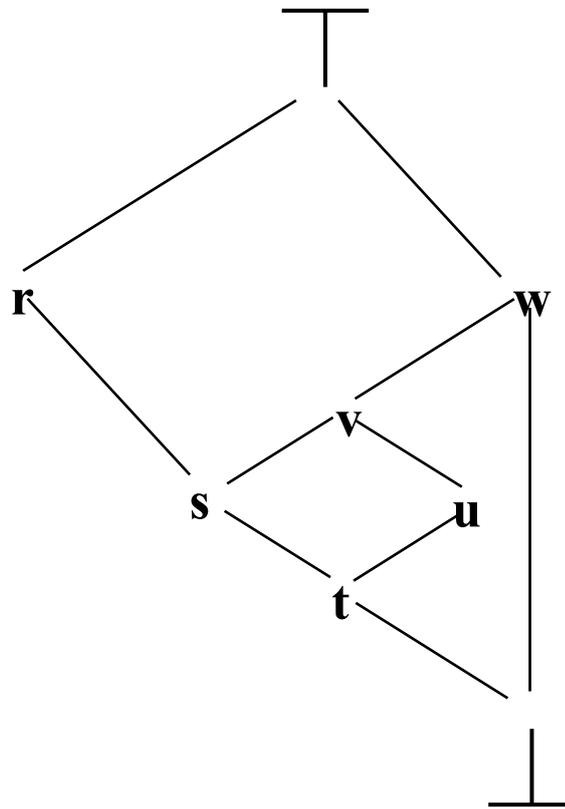


Рис. 3.13 Представление с помощью графа иерархии типов

Таким образом концептуальный граф имеет вид распознающего или идентифицирующего графа. Однако, знания такого рода для интеллектуальной технологической системы носят фрагментарный характер. Они, как правило, недостаточны для организации целенаправленного функционирования производственного комплекса. Тем не менее, эти фрагментарные знания могут с успехом использоваться для синтеза правил поведения интеллектуальной производственной системы типа «класс ситуаций - действие». Такие правила-фреймы позволяют организовать целенаправленное адаптивное поведение интеллектуальной производственной системы в недетерминированной и изменяющейся производственной обстановке. Например, в технологической системе распознавания и адресования деталей на подвесном конвейере в качестве «класса ситуаций» может использоваться аксиома класса деталей одного типа, а в качестве «действия» - сигнал на приводы стрелок ответвления, обеспечивающий адресование (и доставку) деталей другого класса на соответствующие позиции сборочного конвейера.

3.3 Продукционные правила или продукционные системы

Наиболее удачной моделью представления знаний, отражающей синтез правил « ситуация - действие», являются продукционные правила.

Продукционная система (ПС) - это модель вычислений, имеющая особое значение для создания алгоритмов поиска и моделирования решения задач человеком.

ПС представляет собой базу знаний (правила продукции) и машину вывода - специальную программу «сопоставления по образцу» [11].

Основными компонентами продукционной системы являются ниже приведенные компоненты.

1. Набор продукционных правил или продукций.

Продукция - это пара «условие-действие», которая определяет одну порцию знаний, необходимую для решения задачи.

Условная часть правила - это образец (шаблон), который определяет, когда это правило может быть применено для решения какого-либо этапа задачи.

Часть действия - определяет соответствующий шаг в решении задачи.

2. Рабочая память

Она содержит описание текущего состояния «мира» (проблемы) в процессе рассуждений. Это описание является образцом, который сопоставляется с условной частью продукции с целью выбора соответствующих действий при решении задачи. Если условие некоторого правила соответствует текущему содержанию рабочей памяти, то может выполняться действие, связанное с этим условием. Действия продукционных правил предназначены для изменения содержания рабочей памяти.

3. Цикл «распознавание - действие».

Управляющая структура продукционной системы очень проста. Рабочая память инициализируется начальным описанием задачи. Текущее состояние решения задачи представляется набором образцов в рабочей памяти. Эти образцы сопоставляются с условиями продукционных правил, что порождает подмножество правил вывода, называемое конфликтным множеством. Условия этих правил согласованы с образцами рабочей памяти. Продукции, содержащиеся в конфликтном множестве, называют допустимыми.

Выбирается и активизируется одна из продукций конфликтного множества (разрешение конфликта). Активизация правила означает выполнение его действия, которое осуществляется в соответствии с определенной стратегией: эвристики, приоритезация в порядке нахождения первого соответствия условию. Выполненное действие влечет за собой изменение содержания рабочей памяти.

Схема работы продукционной системы представлена на рис. 3.14.

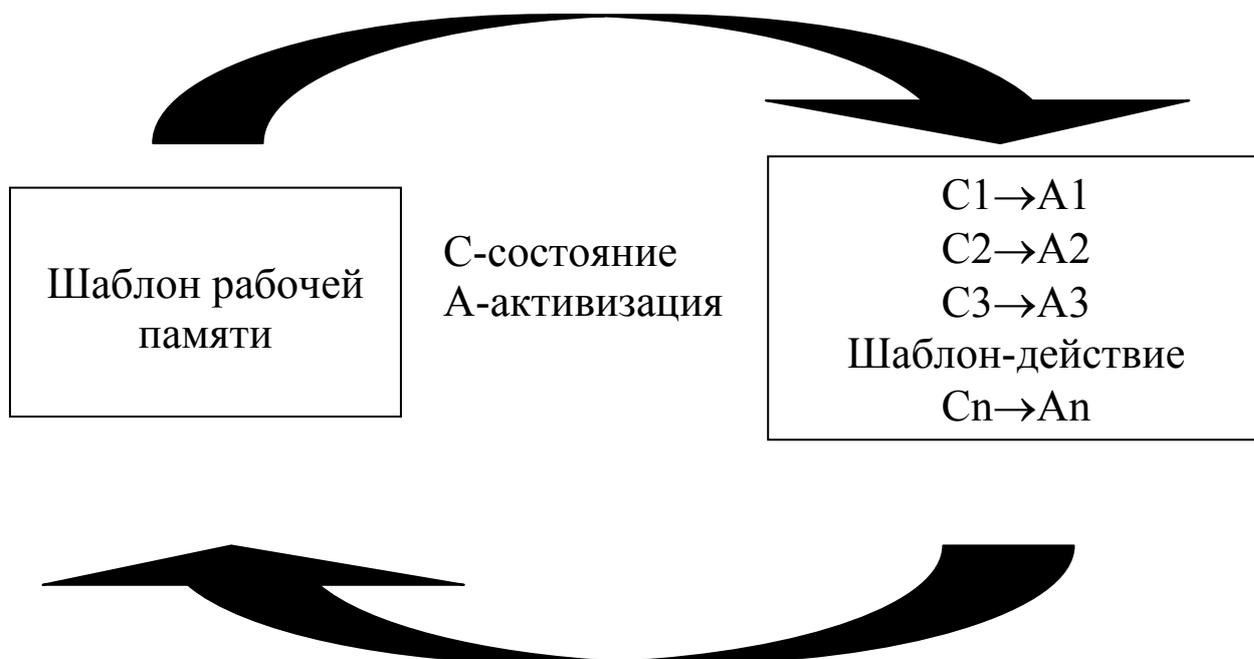


Рис. 3.14 Схема работы продукционной системы

Цикл работы ПС заканчивается, если содержание рабочей памяти не соответствует никаким условиям.

Практическая реализация ПС содержит механизмы возврата в предыдущее состояние рабочей памяти для выхода из тупиковых состояний в процессе поиска.

Для управления поиском на множестве продукционных правил используются несколько стратегий.

1. Управление посредством выбора стратегии поиска на основе данных или от цели. В соответствии с этим ПС могут быть классифицированы как ПС с прямыми (от данных к гипотезе) и обратными (доказательство или опровержение гипотезы) выводами. Часто используется комбинация обратного и прямого поиска.

2. Управление поиском с помощью структуры правил. ПС проверяет правила в определенном порядке -реализация порядка активации правил.

3. Управление поиском через разрешение конфликтов (эвристическое). Эвристики поиска могут быть непосредственно представлены в правилах, описывающих знания. Простейший пример - это условие, при котором из рабочей памяти выбирается первое соответствующее правило. Потенциально для разрешения конфликтов может быть применена любая стратегия.

Так, например, один из языков программирования ЭС использует следующие стратегии.

1. Рефракция. После активизации правило не может быть запущено снова, пока не изменятся элементы рабочей памяти, соответствующие его условию. Рефракция препятствует заикливлению.

2. Новизна. Предпочтение отдается правилам, условия которых соответствуют образцам, добавленным в рабочую память последними. Поиск

сосредотачивается на одной линии рассуждения.

3. Специфичность. Используется более конкретное, а не более общее правило. Более специфично правило в том случае, если оно содержит больше условий и, следовательно, соответствует меньшему количеству образцов.

Продукционные системы имеют следующие достоинства:

1. Разделение знания и управления.

Простота изменения БЗ, без изменения кода программы управления. Знания - в правилах, управление - в цикле «условие-действие».

2. Естественное соответствие поиску в пространстве состояний. Компоненты ПС естественно отображают логическую структуру поиска в пространстве состояний.

Последовательные состояния рабочей памяти - вершины графа пространства состояний.

Правила вывода - набор возможных переходов между состояниями. Упрощается выполнение отладки и документации алгоритмов поиска.

3. Модульность продукционных правил.

Отсутствие синтаксического взаимодействия продукционных правил. Правила могут только влиять на активизацию других правил, изменяя образец в рабочей памяти. Правила не могут «вызывать» другие правила непосредственно, как подпрограммы и устанавливать значения переменных в других продукционных правилах. Эта синтаксическая независимость позволяет реализовывать инкрементальную систему разработки ЭС путем последовательного добавления, удаления или изменений знаний (правил) системы.

4. Возможности эвристического управления поиском.

5. Трассировка и трактовка.

Облегчается контроль за работой системы. Цепочка правил, используемых в процессе решения, отражает «цепочку рассуждений» системы и дает ясную интерпретацию прошедших действий и методов решения задачи.

6. Независимость от выбора языка.

Модель управления ПС не зависит от представления правил и рабочей памяти, если это представление поддерживает сравнение с образцами.

7. Правдоподобная модель решения задач человеком, т.е. отражает способ мышления человека при решении задач.

К недостаткам причисляют следующее:

1. Процесс вывода менее эффективен, поскольку большая часть времени при выводе затрачивается на непосредственную проверку применимости всех правил.

2. Сложно представить родовидную иерархию понятий. Представление знаний с помощью ПС иногда называют «плоским», так как в ПС отсутствуют средства для установления иерархии правил.

3. Объем БЗ увеличивается линейно, пропорционально включению в нее новых фрагментов знаний.

Резюмируя все выше представленные модели представления знаний в интеллектуальных технических системах можно прийти к выводу, что основная дилемма выбора заключается между выразительностью и

эффективностью вывода. Резюме представлено на рис.3.15.

2. Представление знаний
2.9. Общее



Рис. 3.15 Сравнение всех представленных моделей представления знаний интеллектуальных систем

3.4 Методы поиска решений технологических задач

Правильно составленные выражения исчисления предикатов позволяют описывать объекты и отношения из области определения, а правила вывода (например, правило отделения *modus ponens*) - логически получать новые знания из логических описаний. Эти правила вывода определяют пространство, в котором ведется поиск решения задачи.

Чтобы разрабатывать и внедрять алгоритмы поиска, разработчик должен уметь анализировать и прогнозировать их поведение. При этом перед ним стоят такие вопросы.

Гарантировано ли нахождение решения в процессе поиска?

Является ли поиск конечным, или в нем возможны зацикливания?

Если решение найдено, является ли оно оптимальным?

Как процесс поиска зависит от времени выполнения и используемой памяти?

Как интерпретатор может наиболее эффективно упростить поиск?

Ниже излагаемый материал представляет собой краткий обзор теории поиска в пространстве состояний, основной задачей которого является формирование общих представлений о теории поиска в пространстве состояний.

3.4.1 Структуры данных для поиска в пространстве состояний

Граф состоит из множества вершин N_1, N_2, \dots, N_n , которое не обязано быть конечным, и множества дуг, соединяющих некоторые пары вершин.

Граф является размеченным, если для каждой вершины графа заданы метки или дескрипторы. На графе пространства состояний эти дескрипторы идентифицируют состояния в процессе решения задачи. Если дескрипторы двух вершин не различаются, то вершины считаются одинаковыми.

Граф является ориентированным, если каждой дуге на нем приписано направление. Дуги графа также могут быть размеченными. Метки дуги используются для задания именованного отношения.

Путь на графе - это последовательность дуг, соединяющих соседние вершины. Путь представляется списком дуг, представленных в порядке их следования.

Корневой граф - содержит единственную вершину, от которой существует путь к любой вершине графа. Эта вершина называется корнем. Корень не имеет родителей.

Дерево - это граф, на котором для любых двух вершин существует не более одного пути между ними. Дерево не содержит петель и циклов.

Отношения на графах описываются с помощью понятий:

- родители,
- потомки (предок предшествует потомку),

-вершины-брatья (образованы одними родителями).

В корневом графе вершина называется предком всех вершин, расположенных после нее (правее нее), и в то же время потомком всех вершин, расположенных ниже нее (левее нее).

Если путь включает в себя некоторую вершину более одного раза, то говорят, что путь содержит петлю или цикл. Дуги в корневом дереве ориентированы от корня. Каждая вершина в корневом дереве имеет единственного родителя.

3.4.2 Представление задачи в пространстве состояний

В пространстве состояний задачи вершины графа - это состояния решения задачи, а дуги - это этапы ее решения. Одно или несколько начальных состояний, соответствующих исходной информации о состоянии задачи, образуют корень дерева.

Граф также включает одно или несколько целевых условий, которые соответствуют решениям исходной задачи. Поиск в пространстве состояний характеризует процесс нахождения пути решения.

Пространство состояний определяется четверкой $[N, A, S, GD]$:

N - множество вершин графа или состояний в процессе решения задачи;

A - множество дуг между вершинами, соответствующих шагам в процессе решения задачи.

S - это непустое множество начальных состояний задачи.

GD - непустое подмножество N , состоящее из целевых состояний. Эти состояния описываются одним из следующих способов.

1. Изменяемыми свойствами состояний, встречающихся в процессе поиска.
2. Свойствами путей, которые приобретаются в процессе поиска, например, стоимостью перемещения по дугам пути.

Допустимый путь - это путь из вершины множества S в вершину из множества GD .

Путь к решению является целью поиска. Он начинается от исходного состояния и продолжается, пока не будет достигнута цель поиска. Порождение новых состояний вдоль пути обеспечивается такими операторами, как «допустимые ходы» в игре или правила вывода в логической задаче или ЭС.

Задача алгоритма поиска состоит в нахождении допустимого пути в пространстве состояний. Алгоритмы поиска должны направлять пути от начальной вершины к целевой, поскольку именно они содержат цепочку операций, ведущую к решению задачи.

Одна из основных проблем - это возможность достижения результатов разными путями. Поэтому становится очень важным найти *оптимальный*

путь решения задачи.

Множественные пути к состояниям могут привести к петлям и циклам. Поэтому алгоритмы поиска на произвольных графах должны обнаруживать и устранять петли на допустимых путях.

3.4.3 Поиск на основе данных и от цели

Поиск на основе данных (прямая цепочка) начинается с условия задачи и выполняется путем применения допустимых правил для получения новых фактов, ведущих к цели.

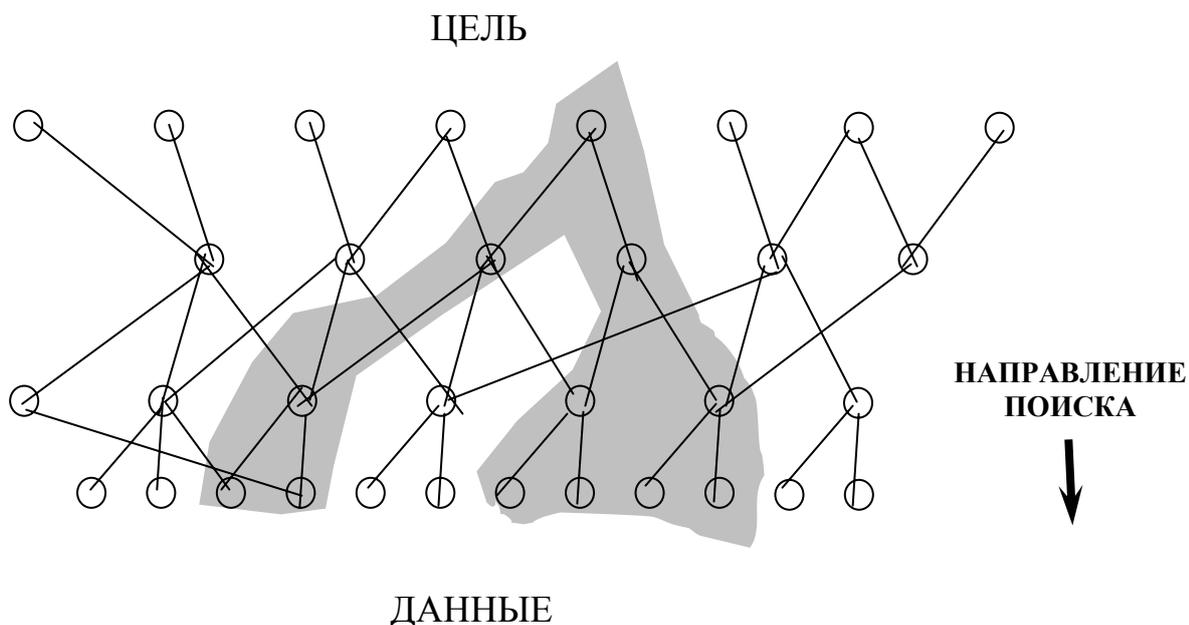


Рис. 3.16 Пространство состояний, поиск от цели

Поиск на основе данных применяется при следующих условиях.

1. Все или большинство исходных данных заданы в постановке задачи. Задача интерпретации состоит в выборе этих данных и их представлении в виде, подходящем для использования в интерпретирующих системах более высокого уровня.
2. Существует большое число потенциальных целей, но всего лишь несколько способов применения фактов и представления информации о конкретном примере задачи.
3. Сформулировать цель или гипотезу очень трудно.

Поиск от цели (обратная цепочка) начинается с обращения к цели и продолжается путем определения правил, которые могут привести к цели, и построения цепочки подцелей, ведущей к исходным данным задачи.

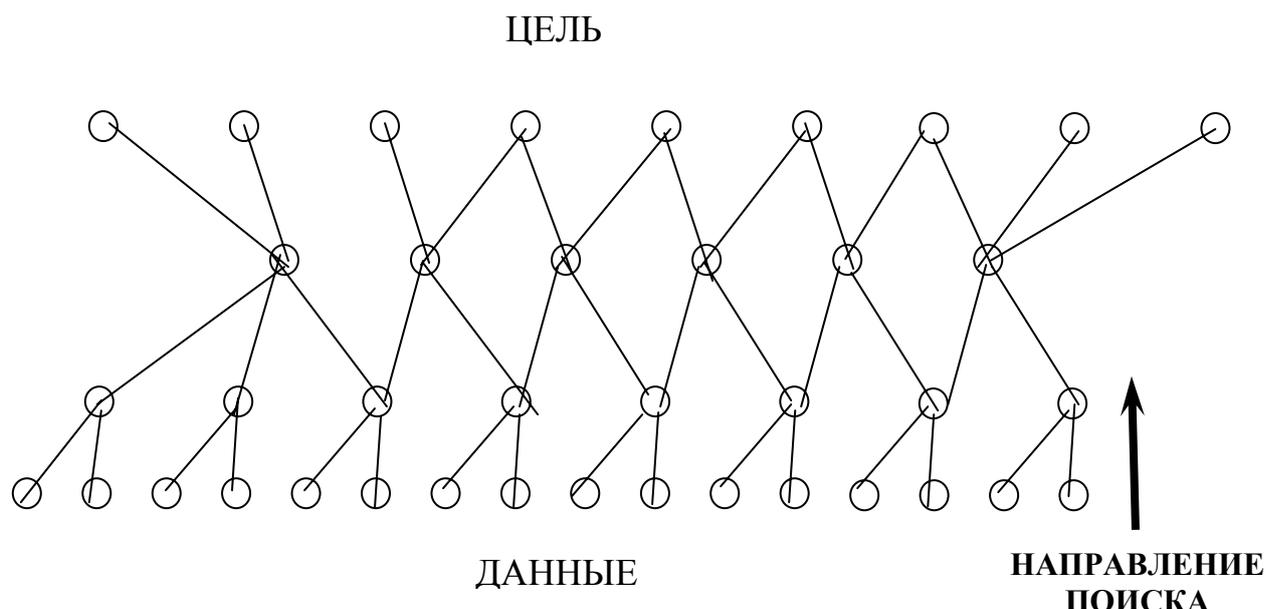


Рис. 3.17

Пространство состояний, поиск на основе данных

Алгоритм осуществления поиска от цели можно записать следующим образом:

1. Рассмотрение цели.
2. Анализ правил, ведущих к цели, и определение условий их применения.
3. Идентификация правил как подцелей.
4. Осуществление поиска.

Выбор той или иной стратегии поиска зависит от условий конкретной задачи.

Процесс поиска от цели рекомендован в следующих случаях.

1. Цель поиска (или гипотеза) явно присутствует в постановке задачи или может быть легко сформулирована.
2. Имеется большое число правил, которые на основе полученных фактов позволяют продуцировать возрастающее количество заключений и целей. Своевременный отбор вновь возникших целей позволяет отсеять множество возможных ветвей и сделать процесс поиска более эффективным.
3. Исходные данные не приводятся в задаче, но подразумевается, что они должны быть известны решателю. В этом случае стратегия поиска от цели играет роль руководителя для правильности постановки задачи.

При поиске от цели подходящие правила применяются для исключительно не перспективных ветвей поиска.

Пример - система DENDRAL для исследования молекулярных структур органических соединений, или системы PROSPECTOR и Dipmeter - анализаторы данных, определяющие, какие минералы с наибольшей вероятностью могут быть найдены в некотором месте.

3.5 Поиск в глубину и в ширину

Существует два варианта обхода узлов графа, представляющего собой пространство состояний, отражающих этап решения задачи :

поиск в глубину (*depth - fist*);

поиск в ширину (*breadth - fist*).

3.5.1. Поиск в ширину

Поиск в ширину исследует пространство состояний по уровням, один за другим. Только если состояний на данном уровне больше нет, алгоритм переходит к следующему уровню.

Поиск в ширину использует два списка *open* и *closed* для отслеживания продвижения в пространстве состояний. На рис. 3.18 представлен граф, отражающий пространство состояний и списки *open* и *closed*.

Список *open* подобно списку **NSL** в алгоритме поиска с возвратами, содержит сгенерированные состояния, потомки которых еще не были исследованы. Порядок удаления из списка *open* определяет порядок поиска.

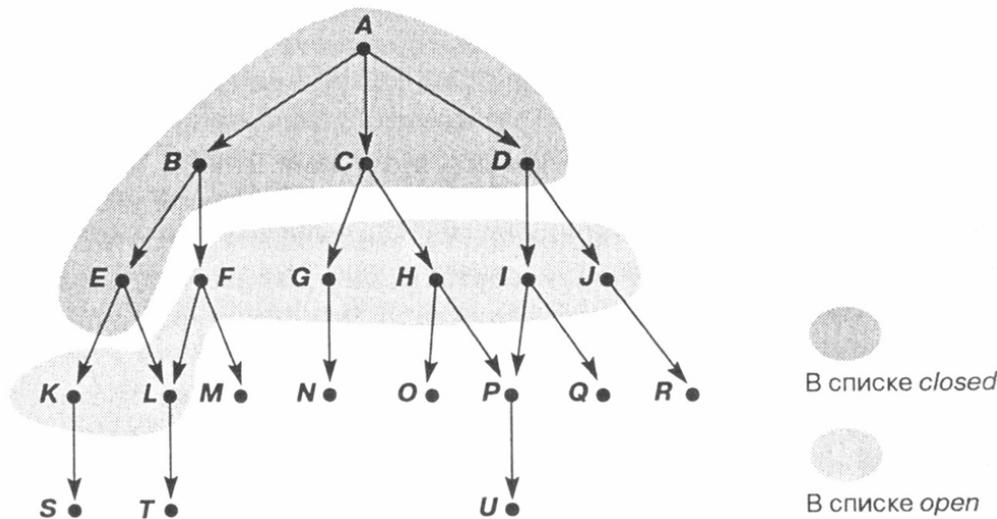
В список *closed* заносятся уже исследованные состояния. Список *closed* объединяет в себе списки **DE** и **SL**, используемые в алгоритме поиска с возвратами.

Дочерние состояния генерируются правилами вывода. На каждой итерации генерируются все дочерние состояния *X* и записываются в список *open*.

Список *open* действует как «очередь» и обрабатывает данные в порядке поступления или «первым поступил- первым обслужен». Данная структура данных носит название *FIFO first-in-first-out*. Состояния добавляются в список справа, а удаляются слева. Таким образом, в поиске участвуют состояния, которые находятся в списке *open* дольше всего. Дочерние состояния, которые уже были записаны в списки *open* или *closed* отбрасываются.

«Пограничные» состояния поиска на любой стадии записываются в *open* , а уже рассмотренные - в *closed*.

Поскольку при поиске в ширину узлы графа рассматриваются по уровням, сначала исследуются те состояния, пути к которым короче. Таким образом, поиск в ширину гарантирует нахождение кратчайшего пути от начального состояния к цели. Более того, поскольку вначале рассматриваются те состояния, пути до которых были короче, то при повторном проходе эти состояния отбрасываются.



1. `open=[A]; closed=[]`
2. `open=[B,C,D]; closed=[A]`
3. `open=[C,D,E,F]; closed=[B,A]`
4. `open=[D,E,F,G,H]; closed=[C,B,A]`
5. `open=[E,F,G,H,I,J]; closed=[D,C,B,A]`
6. `open=[F,G,H,I,J,K,L]; closed=[E,D,C,B,A]`
7. `open=[G,H,I,J,K,L,M]` (так как L уже в open); `closed=[F,E,D,C,B,A]`
8. `open=[H,I,J,K,L,M,N]; closed=[G,F,E,D,C,B,A]`
9. И так далее, пока или U найдено, или `open=[]`

Рис. 3.18 Граф, представляющий собой пространство состояний и списки *open* и *closed* для всех итераций поиска. Выделена шестая итерация

Все посещенные состояния хранятся в *closed*. Если путь является решением, то он возвращается алгоритмом. Это может быть сделано путем накопления информации и предках каждого состояния. Состояние хранится вместе с записью родительского состояния, т.е. в виде пары (состояние, родитель)

Пример:

open = [(D,A), (E,B), (F,B), (G,C), (H,C)] *closed* = [(C,A), (B,A), (A,nil)]

A имеет родителя nil, т.е. является начальным состоянием. Это служит сигналом прекращения восстановления пути. Поскольку поиск в ширину находит каждое состояние по кратчайшему пути и сохраняет каждую версию кратчайшего состояния, этот путь от начала до цели является самым коротким.

Достоинства

1. Находит самый кратчайший путь к вершине.
2. Если в задаче существует простое решение, то оно будет найдено.

Недостатки

1. При большом коэффициенте ветвления или в случае необходимости глубокого поиска возникает опасность комбинаторного взрыва.
2. Степень использования пространства при поиске в ширину определяется в терминах списка *open* и является экспоненциальной функцией длины пути в любой момент времени, что является не-желательным, если путь к решению достаточно длинный.

3.5.2 Поиск в глубину

При поиске в глубину после исследования состояния начала необходимо оценить все его потомки и их потомки, а затем исследовать одну из его вершин братьев. Поиск в глубину по возможности углубляется в область поиска. Если дальнейшие потомки состояния не найдены, то рассматриваются вершины брата. Данный алгоритм поиска является упрощенным вариантом поиска с возвратами.

В этом алгоритме состояния, являющиеся потомками исходного состояния, добавляются и удаляются с левого конца списка *open*, т.е. список *open* реализован как стек магазинного типа или структура данных *LIFO* - „*last-in-first-out*“ («последним пришел- первым обслужен»). При организации списка *open* в виде стека предпочтение отдается самым «молодым», недавно сгенерированным состояниям, т.е. осуществляется принцип поиска в глубину.

Как и при поиске в ширину, в списке *open* перечислены все обнаруженные, но еще не оцененные состояния (текущая граница поиска), а в *closed* записаны уже рассмотренные состояния.

В отличие от поиска в ширину поиск в глубину не гарантирует нахождения оптимального пути к состоянию, если оно встретилось впервые. Позже в процессе поиска могут быть найдены различные пути к данному состоянию. Если длина пути имеет значение в процессе решения задачи, то в случае нахождения алгоритмом некоторого состояния повторно, необходимо сохранить именно тот путь, который является короче. Это можно сделать, сохраняя тройку значений (состояние, родитель, длина пути).

При генерации дочерних состояний значение длины пути просто увеличивается на единицу и сохраняется вместе с потомками. Если дочернее состояние достигнуто по многим путям, эту информацию нужно использовать для сохранения лучшей версии. Эта идея развивается при рассмотрении *алгоритма А*. Тем не менее, сохранение лучшей версии состояния при простом поиске в глубину совершенно не гарантирует, что цель будет достигнута по кратчайшему пути.

После выбора стратегии поиска, от данных или от цели, оценки графа и выбора метода поиска (в глубину или в ширину), дальнейший ход действий зависит от конкретной задачи.

При выборе метода поиска решающую роль могут играть следующие

факторы:

1. Необходимость обнаружения кратчайшего пути к цели.
2. Коэффициент ветвления пространства поиска.
3. Доступное время вычислений и возможности пространства.
4. Средняя длина пути к целевой вершине.
5. Необходимость получения всех решений или только первого.

Достоинства

1. Быстро проникает в глубину пространства.
2. Эффективен для областей поиска с высокой степенью связанности, так как ему не нужно помнить все узлы в списке *open*.
3. Степень использования пространства состояний - это линейная функция от пути.

Недостатки

1. Может затеряться в глубинах графа пространства состояний.
2. Не обеспечивает наиболее короткого пути к цели.

3.5.3 Поиск в глубину с итерационным заглаблением

При таком поиске используется предельное значение глубины поиска, которая позволяет ограничить поиск только заданным числом уровней.

Итерационным поиском называется поиск в глубину, первая итерация которого ограничена 1 уровнем. Если цель не найдена, то выполняется еще один шаг с глубиной 2 и т.д. то тех пор пока не будет найдена цель. На каждой итерации алгоритм выполняет поиск в глубину с учетом текущего предельного числа уровней. При этом, при переходе от одной итерации к другой информация о пространстве состояний не сохраняется.

Кажется что итерационное продвижение менее эффективно по времени, чем поиск в глубину или в ширину. Однако временная сложность такого алгоритма имеет тот же порядок величины, что и каждый из этих алгоритмов $O(B^n)$, где n - степень использования пространства на каждом уровне, B - среднее число дочерних состояний вершины.

При выборе метода поиска решающую роль могут играть следующие факторы:

- необходимость обнаружения кратчайшего пути к цели;
- коэффициент ветвления пространства поиска;
- доступное время вычислений и возможности пространства;
- средняя длина пути к целевой вершине;
- необходимость получения всех решений или только первого.

3.6 Процедура поиска с возвратами. BACKTRACK.

Решение недетерминированных задач просто представить с помощью дерева поиска в глубину с возвратами (для языка ПРОЛОГ) на примере реализации метода резолюции (рис. 3.19). Для более четкого понимания постановки задачи в таблице 1 приведен синтаксис клаузальных выражений, используемых языком логического программирования ПРОЛОГ.

Таблица 3.1 Синтаксис клаузальных хорновских выражений языка ПРОЛОГ

	ИП1	PROLOG
Факт	$p(t, \dots tn) \leftarrow true$	$p(t, \dots tn)$
Правило	$p(t, \dots tn) \leftarrow q(t, \dots tn) \wedge r(t, \dots tn).$	$p(t, \dots tn) :- q(t, \dots tn), r(t, \dots tn).$
Вопрос (Цель)	$false- \leftarrow p(t, \dots tn) \wedge q(t, \dots tn).$	$?- \leftarrow p(t, \dots tn), q(t, \dots tn).$

В случае, когда БЗ не содержит альтернативных вариантов подстановки (альтернативных решений) или иначе, постановка задачи детерминирована, применение логического программирования не требуется, т.к. решение проблемы может быть алгоритмизировано, могут быть использованы классические методы программирования.

Применение логического программирования (построение интеллектуальных программных продуктов) становится необходимым при возможности принятия альтернативного решения задачи (для резолюции - возможность альтернативных подстановок).

Логические программы не содержат алгоритмов решения задачи, решение осуществляется с помощью поиска в глубину с возвратами («backtrack») и апробации всех 2х возможных альтернативных решений.

При каждом переходе в новое состояние производится оценка на допустимость состояний и, если при использовании правила перехода для текущего состояния получается недопустимое состояние, то производится возврат к тому предыдущему состоянию, из которого было достигнуто это текущее состояние. Эта процедура получила название бэктрекинг (bac tracing или BACKTRACK).

Пример постановки задачи, решение которой осуществляется с помощью метода резолюции и графически представлено на рис.3. 19 в качестве графа поиска в глубину с возвратами.

- База знаний:** (1) A1.
 (2) A2.
 (3) B1 :- A1, A3.
 (4) B1 :- A1, A2.

Цель (Вопрос, гипотеза): ?- A₂, B₁.

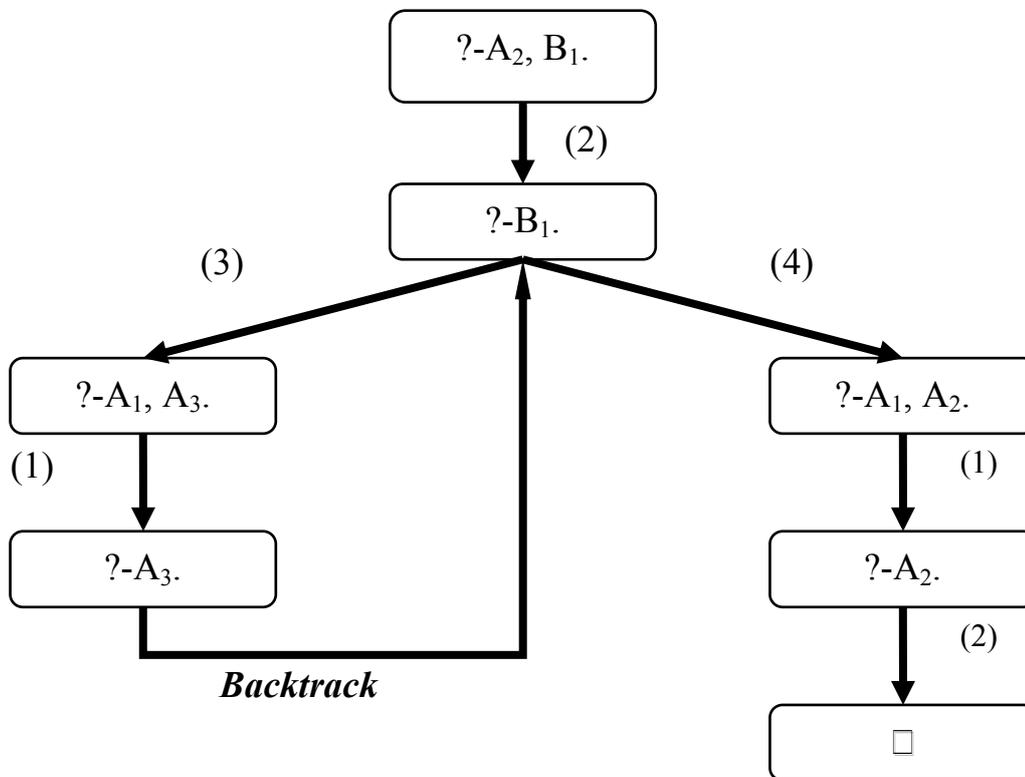


Рис. 3.19

Резолюция как граф поиска в глубину с возвратами

Поиск с возвратами - это метод систематической проверки различных путей в пространстве состояний. Такой поиск допускает естественную реализацию в рекурсивной среде, ориентированной на использование стеков.

Стек - это линейная структура, доступ к которой осуществляется только с одного конца. Следовательно, элементы добавляются и удаляются из структуры с одного конца. Стек обладает структурой данных LIFO (Last-In-First-Out) «последним вошел, первым вышел». Примером работы такой структуры служить поиск в глубину.

Для функционирования стека необходимо определить следующие операции:

1. Проверка наличия элементов в стеке (проверка пустоты стека);
2. Добавление элемента в стек;
3. выталкивание или удаление последнего элемента (или верхушки) из стека;
4. Просмотр последнего элемента (верхушки) без удаления;
5. Проверка наличия данного элемента в стеке;
6. Добавления списка элементов в стек.

Операции 5 и 6 можно определить на основании первых четырех операций.

Особенность алгоритма в возвратами «запускается» в случае, если поиск привел в тупиковую вершину. В этом случае алгоритм возвращается в ближайшую из пройденных вершин и исследует все вершины брата, а затем спускается по одной из ветвей, ведущих от вершины брата. Этот процесс описывается следующим рекурсивным правилом.

Если исходное состояние S не удовлетворяет требованиям цели, то из списка его потомков выбирается первый S_{child1} , и к этой вершине рекурсивно применяется процедура поиска с возвратами. Если в результате поиска с возвратами в подграфе с корнем S_{child1} цель не обнаружена, то процедура повторяется для S_{child2} . Эта процедура продолжается до тех пор, пока один из потомков рассматриваемой вершины-брата не окажется целевым узлом, либо не выяснится что все вершины брата уже рассмотрены. Если же ни одна из вершин братьев вершины S не привела к цели, то возвращаемся к предку вершины S , и процедура повторяется для вершины брата S и т.д.

В алгоритме поиска с возвратами, а также алгоритмах поиска в глубину, поиска в ширину и поиска по первому наилучшему совпадению (об этих алгоритмах речь пойдет ниже) используют идеи поиска с возвратами, в том числе следующие.

1. Формируется список неисследованных состояний (NSL), для того чтобы иметь возможность возвратиться к любому из них;
2. Поддерживается список неудачных состояний (DE), чтобы оградить алгоритм от проверки бесполезных путей;
3. Поддерживается список узлов текущего пути (SL), который возвращается по достижении цели. Текущее состояние всегда равно последнему состоянию, занесенному в этот список;
4. Каждое новое состояние проверяется на вхождение в эти списки, чтобы предотвратить заикливание;

3.7 Алгоритмы эвристического поиска

Эвристика (Э) - набор правил для выбора тех ветвей в пространстве состояний, которые с наибольшей вероятностью приведут к приемлемому решению проблемы [12].

В области ИИ эвристики используются в двух ситуациях.

1. Проблема может не иметь точного решения из-за неопределенности в постановке задачи и/или в исходных данных (пример - медицинская диагностика).
2. Проблема может иметь точное решение, но стоимость его поиска может быть очень высокой. (эвристика позволяет избежать этой проблемы и вести поиск по наиболее «перспективному» пути, исключая из рассмотрения не перспективные состояния и их потомки)

Эвристика - это только предположение следующего шага, который будет сделан на пути решения проблемы.

Эвристические алгоритмы состоят из двух частей:

-эвристическая мера;

-алгоритма, который использует эвристическую меру для поиска в пространстве состояний.

3.7.1 Алгоритм наискорейшего спуска для дерева решений

Пример построения более узкого дерева рассматривается в задаче о коммивояжере, наиболее часто приводимой для демонстрации данного алгоритма. Торговец должен побывать в каждом из 5 городов, обозначенных на карте (рис. 3.20).

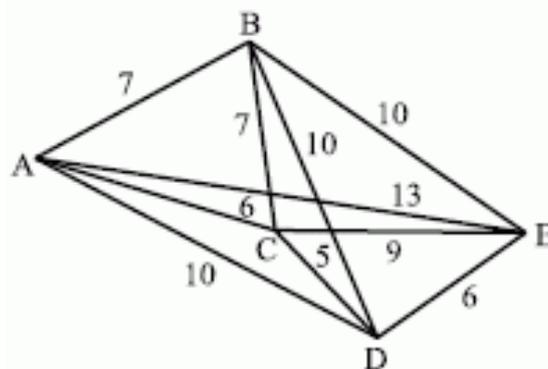


Рис.3 .20

Пространственный граф состояний задачи о коммивояжере

Задача состоит в том, чтобы, начиная с города А, найти минимальный путь, проходящий через все остальные города только один раз и приводящий обратно в А. На графе, представленном на рис. 3.20 дуги отражают расстояние между городами, а вершины-города, каждая из дуг имеет длину, коротая обозначена на графе. Идея метода исключительно проста - из каждого города идем в ближайший, где мы еще не были. Решение задачи показано на рис. 3.21.

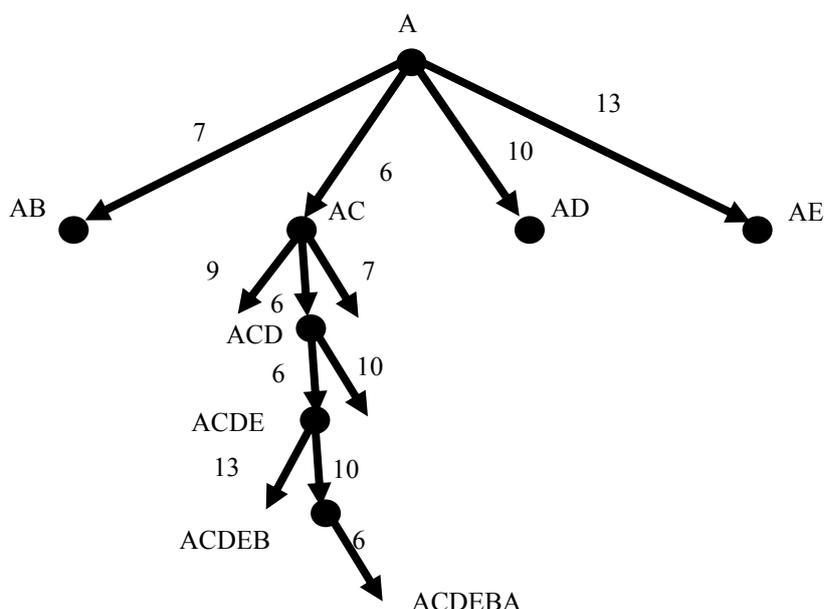


Рис. 3.21

Граф пространства состояний, представляющий решение задачи о коммивояжере с помощью алгоритма наискорейшего спуска.

Такой алгоритм поиска решения получил название алгоритма наискорейшего спуска (в некоторых случаях - наискорейшего подъема). Алгоритм наискорейшего спуска иногда именуется еще и «жадным алгоритмом». Основным механизмом такого алгоритма эвристического поиска является применение процедуры поиска экстремума. Стратегии, основанные на поиске экстремума, оценивают не только текущее состояние, но и его потомков. Для дальнейшего поиска выбирается наилучший потомок, при этом о родителях и братьях забывают. Поиск прекращается, когда достигается состояние, которое лучше, чем любой из его наследников. Основным недостатком такой стратегии поиска является возможность остановки поиска на локальном максимуме. Решением этой проблемы является применение более гибкого варианта «жадного» алгоритма.

Решение заключается в использовании «жадным» алгоритмом приоритетной очереди, что обеспечивает восстановление алгоритма из точки локального максимума.

«Жадный» алгоритм поиска как и поиски в глубину и в ширину использует списки сохранения состояний: *open* - содержит текущие состояния поиска, *closed* - все проверенные состояния.

Отличительной чертой данного алгоритма является то, что на каждом шаге алгоритма в список *open* записываются состояния с учетом некоторой «эвристической оценки» его близости к цели, т.е. образуется приоритетная очередь.

Функция *best_first_search* вычисляет эвристическую оценку состояний в списке *open* и сортирует список в соответствии с этими эвристическими значениями. Таким образом, на каждой итерации будут рассматриваться более перспективные состояния списка *open*.

«Жадный» алгоритм поиска - это общий алгоритм эвристического поиска на любом графе пространства состояний (как алгоритмы поиска в ширину или глубину). Этот алгоритм в одинаковой степени применим к поиску данных и поиску решения задачи, поддерживает различные функции задачи.

На каждой итерации функция *best_first_search* удаляет первый элемент из списка *open*. Достигнув цели, алгоритм возвращает путь, который ведет к решению. Каждое состояние сохраняет информацию о родителе с целью его восстановления, чтобы позволить алгоритму найти кратчайший путь к решению.

Если первый элемент в списке *open* не является решением, то алгоритм использует продукционные правила соответствия и операции, чтобы сгенерировать все возможные потомки данного элемента. Если потомок уже находится в списке *open* или *closed*, то алгоритм выбирает кратчайший из возможных путей достижения этого состояния.

Поскольку при выборе состояния используется эвристика, которая может оказаться ошибочной, то алгоритм не отказывается от остальных состояний и сохраняет в списке *open*. Если ошибка встречается, то алгоритм в итоге возвращается к некоторому ранее сгенерированному «лучшему» состоянию в *open* и продолжает поиск в другой части пространства, т.е. выполняется операция BACKTRACK.

3.7.2. Алгоритмы оценочных (штрафных) функций

Функция эвристической оценки состояний f состоит из двух компонентов:

$$f = g(n) + h(n), \quad (3.1)$$

где $g(n)$ - фактическая длина пути от произвольного состояния n к начальному;

$h(n)$ - эвристическая оценка расстояния (стоимости пути) от состояния n к цели.

Если эта функция используется в алгоритме *best_first_search* как стоимость пути (стоит в скобках) оценки для $h(n)$, то такой алгоритм называется алгоритмом А.

Пример функции эвристической оценки состояний f в игре «Пятнашка» (рис. 3.22)

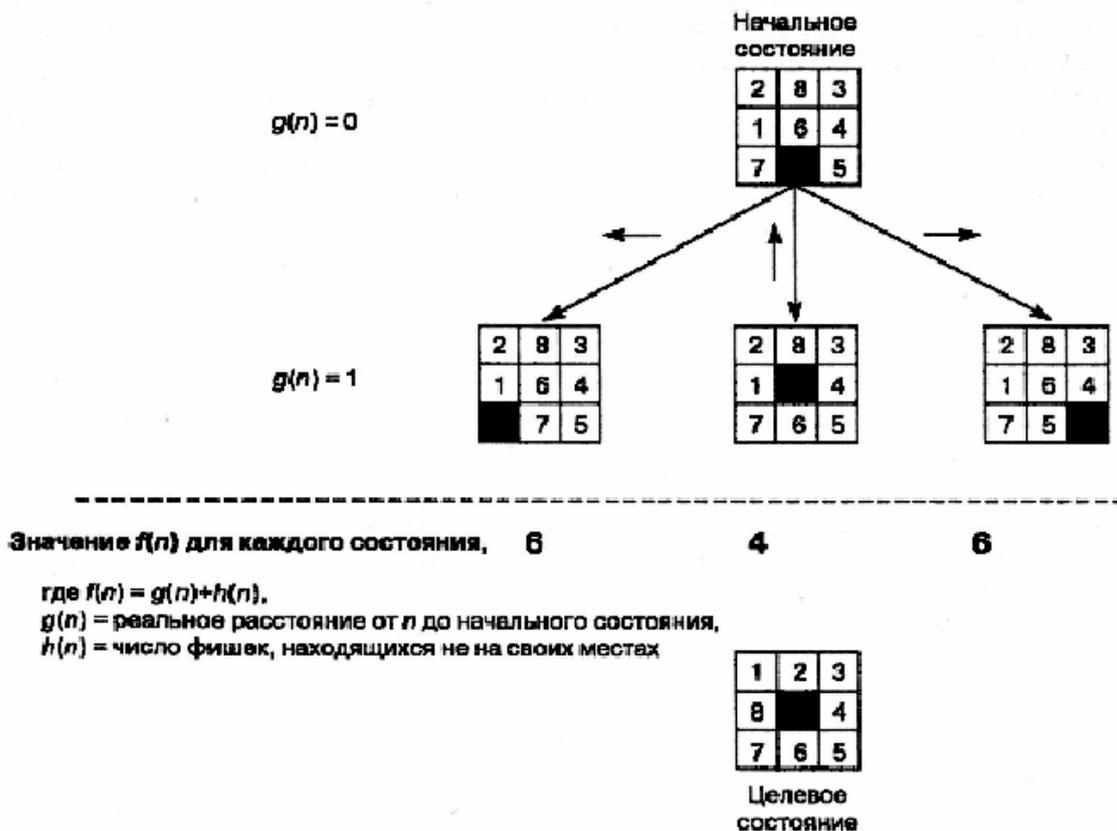


Рис. 3.22

Пример функции эвристической оценки состояния

Состояния в списке *open* сортируются в соответствии со значением f .

В действительности компонент $g(n)$ придает поиску свойства поиска в ширину. Он предотвращает возможность заблуждения из-за ошибочной оценки: если эвристика непрерывно возвращает «хорошие» оценки для

состояний на пути, не ведущем к цели, то значит g будет расти и доминировать над h , возвращая процедуру поиска по кратчайшему пути. Это избавляет алгоритм от зацикливания.

Компонент $h(n)$ ведет алгоритм поиска к эвристически наиболее перспективным состояниям.

В отличие от игр, на примерах которых были разработаны эвристические алгоритмы, в таких системах как ЭС или для планирования к каждому состоянию не может быть применена одна и та же эвристика.

Эвристики, решающие специфические задачи «зашивают» в синтаксис и семантику соответствующих операторов решения данной задачи. На каждом шаге решения определяется, какую эвристику следует использовать. Проверка осуществляется в процессе проверки соответствия по шаблону.

Поведение эвристики оценивается рядом критериев, и в первую очередь ниже приведенными тремя:

- допустимость;
- монотонность;
- информативность.

Остальные параметры, которые могут существовать, формулируются в зависимости от конкретного приложения.

Мера допустимости

Алгоритм поиска является **допустимым**, или приемлемым, если гарантируется нахождение кратчайшего пути к решению.

Поиск в ширину является допустимой стратегией поиска.

Монотонность - это свойство алгоритма последовательно находить кратчайший путь к каждому состоянию, которое встречается в процессе поиска.

Информированные эвристики

Если из двух эвристик h_1 и h_2 для всех состояний n в пространстве поиска выполняется неравенство $h_1(n) \leq h_2(n)$, то эвристика h_2 является **более информативной**.

В процессе поиска h_2 оценивает на много меньше состояний, она является информативной эвристикой.

Чем больше информирован алгоритм эвристики, тем меньше состояний требуется проверить, чтобы получить оптимальное решение.

Однако, в определенных случаях, небольшое количество состояний может потребовать большого количества вычислений.

Общими эвристическими стратегиями являются (пояснение осуществлено на примере метода резолюции) следующие:

1. Эвристическая стратегия предпочтения единичного выражения

Суть: максимальное использование единичных выражений в процессе резолюции.

Основное требование: одна из резольвент должна всегда представлять собой единичное выражение. При невозможности осуществления данного требования на n -ом шаге резолюции его выполнение обеспечивается на последующих шагах (особенно в случае существования возможности осуществления нескольких альтернативных подстановок)

2. Эвристическая стратегия «множества поддержки»

Суть: для некоторого набора исходных дизъюнктивных выражений S указывается подмножество T , называемое «множеством поддержки».

Основное требование: одна из резольвент в каждом опровержении должна иметь предка из «множества поддержки».

При резолюции, отрицание цели является подмножеством исходных выражений, т.е. оно уже может быть подмножеством поддержки. Использование множества поддержки ускоряет процесс резолюции для пар выражений, в которых одно представляет собой отрицание цели или полученное на его основе дизъюнктивное выражение.

3.7.3 Алгоритм минимакса

В 1945 году Оскар Моргенштерн и Джон фон Нейман предложили метод минимакса, нашедший широкое применение в теории игр [12]. Предположим, что противник использует оценочную функцию (ОФ), совпадающую с нашей ОФ. Выбор хода с нашей стороны определяется максимальным значением ОФ для текущей позиции. Противник стремится сделать ход, который минимизирует ОФ. Поэтому этот метод и получил название минимакса. На рис. 3.23 приведен пример анализа дерева ходов с помощью метода минимакса (выбранный путь решения отмечен жирной линией). Жирной линией отмечен путь, который на каждом своем шаге, поочередно, выполняет условие максимума и минимума.

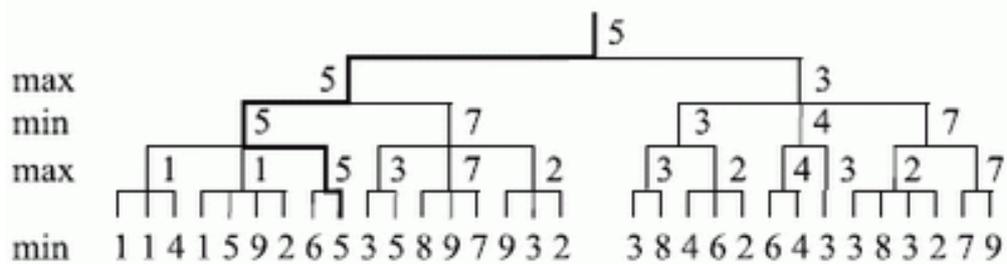


Рис. 3.23 Дерево ходов

Даже такой простой подход позволяет избежать части тупиковых состояний в процессе поиска и сократить время по сравнению с полным перебором. Кстати, этот подход достаточно распространен в экспертных производственных системах.

3.7.4 Альфа-бета-процедура

Теоретически, это эквивалентная минимаксу процедура, с помощью которой всегда получается такой же результат, но заметно быстрее, так как целые части дерева исключаются без проведения анализа. В основе этой процедуры лежит идея Дж. Маккарти об использовании двух переменных,

обозначенных α и β (1961 год) [12]. Основная идея метода состоит в сравнении наилучших оценок, полученных для полностью изученных ветвей, с наилучшими предполагаемыми оценками для оставшихся. Можно показать, что при определенных условиях некоторые вычисления являются лишними. Рассмотрим идею отсечения на примере, который графически представлен на рис. 3.24. Предположим, позиция A полностью проанализирована и найдено значение ее оценки α . Допустим, что один ход из позиции Y приводит к позиции Z, оценка которой по методу минимакса равна z . Предположим, что $z \leq \alpha$. После анализа узла Z, когда справедливо соотношение $y \leq z \leq \alpha \leq s$, ветви дерева, выходящие из узла Y, могут быть отброшены (альфа-отсечение).

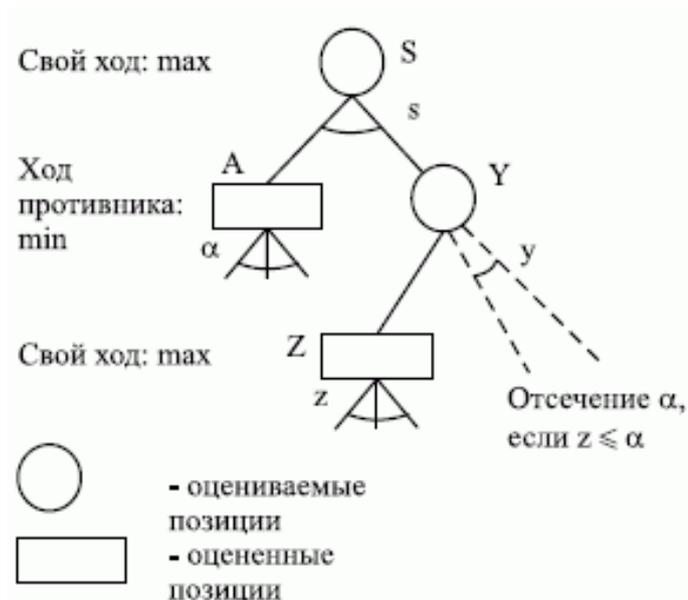


Рис. 3.24

Альфа бета отсечение

Если мы захотим опуститься до узла Z, лежащего на уровне произвольной глубины, принадлежащей той же стороне, что и уровень S, то необходимо учитывать минимальное значение оценки β , получаемой на ходах противника. Отсечение типа β можно выполнить всякий раз, когда оценка позиции, возникающая после хода противника, превышает значение β . Алгоритм поиска строится так, что оценки своих ходов и ходов противника сравниваются при анализе дерева с величинами α и β соответственно. В начале вычислений этим величинам присваиваются значения $+\infty$ и $-\infty$, а затем, по мере продвижения к корню дерева, находится оценка начальной позиции и наилучший ход для одного из противников. Правила вычисления α и β в процессе поиска рекомендуются следующие:

1. у МАХ вершины значение равно наибольшему в данный момент значению среди окончательных возвращенных значений для ее дочерних вершин;
2. у MIN вершины значение β равно наименьшему в данный момент значению среди окончательных возвращенных значений для ее дочерних вершин.

Правила прекращения поиска:

- 1) можно не проводить поиска на поддереве, растущем из всякой MIN вершины, у которой значение β не превышает значения всех ее родительских МАХ вершин;
- 2) можно не проводить поиска на поддереве, растущем из всякой МАХ вершины, у которой значение не меньше значения β всех ее родительских MIN вершин.

На рис.3.25 показаны α β отсечения для конкретного примера. Таким образом, β алгоритм дает тот же результат, что и метод минимакса, но выполняется быстрее.

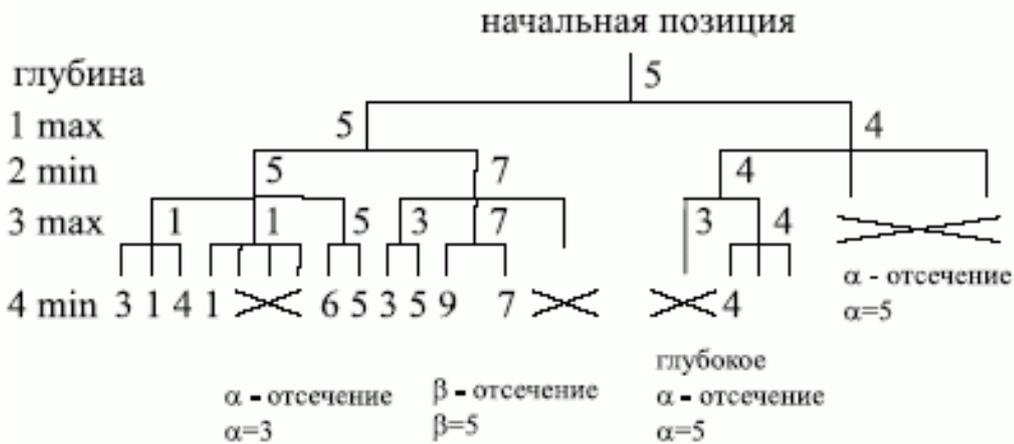


Рис. 3.25.

α β отсечение для конкретного примера

Использование алгоритмов эвристического поиска для поиска на графе И, ИЛИ выигрышной стратегии в более сложных задачах и играх (шашки, шахматы) не реален. По некоторым оценкам игровое дерево игры в шашки содержит 10^{40} вершин, в шахматах 10^{120} вершин. Если при игре в шашки для одной вершины требуется $1/3$ наносекунды, то всего игрового времени потребуется 10^{21} веков. В таких случаях вводятся искусственные условия остановки, основанные на таких факторах, как наибольшая допустимая глубина вершин в дереве поиска или ограничения на время и объем памяти. Многие из

рассмотренных выше идей были использованы А. Ньюэллом, Дж. Шоу и Г. Саймоном в их программе GPS. Процесс работы GPS в общем воспроизводит методы решения задач, применяемые человеком: выдвигаются подцели, приближающие к решению; применяется эвристический метод (один, другой и т. д.), пока не будет получено решение. Попытки прекращаются, если получить решение не удастся, например. Если полученный результат не удовлетворяет условиям эвристической оценки. Программа STRIPS (STanford Research Institut Problem Solver) вырабатывает соответствующий порядок действий робота в зависимости от поставленной цели. Программа способна обучаться на опыте решения предыдущих задач. Большая часть игровых программ также обучается в процессе работы. Например, знаменитая шашечная программа Самюэля, выигравшая в 1974 году у чемпиона мира, "заучивала наизусть" выигранные партии и обобщала их для извлечения пользы из прошлого опыта. Программа HASHER Зуссмана, управляющая поведением робота, обучалась также и на ошибках.

3.8 Поиск решений на основе исчисления предикатов

Семантика исчисления предикатов обеспечивает основу для формализации логического вывода. Возможность логически выводить новые правильные выражения из набора истинных утверждений очень важна. Логически выведенные утверждения корректны и они совместимы со всеми предыдущими интерпретациями первоначального набора выражений. Обсудим вышесказанное неформально и затем введем необходимую формализацию.

В исчислении высказываний основным объектом является переменное высказывание (предикат), истинность или ложность которого зависит от значений входящих в него переменных. Так, истинность предиката "x был физиком" зависит от значения переменной x. Если x - П. Капица, то предикат истинен, если x - М. Лермонтов, то он ложен. На языке исчисления предикатов утверждение $\forall x(P(x) \rightarrow Q(x))$ читается так: "для любого x если P(x), то имеет место и Q(x)". Иногда его записывают и так: $\forall x (P(x) \rightarrow Q(x))$. Выделенное подмножество тождественно истинным формулам (или правильно построенным формулам - ППФ), истинность которых не зависит от истинности входящих в них высказываний, называется аксиомами.

В исчислении предикатов имеется множество правил вывода.

1. *Modus ponens*, которое читается так "если истинна формула A и истинно, что из A следует B, то истинна и формула B"

2. *Modus tollendo tollens* : Если из A следует B и B ложно, то и A ложно.

3. *Modus ponendo tollens* : Если A и B не могут одновременно быть истинными и A истинно, то B ложно.

4. *Modus tollendo ponens* : Если либо A, либо B является истинным и A не истинно, то B истинно.

Формулы, находящиеся над чертой, называются посылками вывода, а под чертой - заключением. Это правило вывода формализует основной закон дедуктивных систем: из истинных посылок всегда следуют истинные заключения. Аксиомы и правила вывода исчисления предикатов первого порядка задают основу формальной дедуктивной системы, в которой происходит формализация схемы рассуждений в логическом программировании. Можно упомянуть и другие правила вывода.

Решаемая задача представляется в виде утверждений (аксиом) $F_1, F_2 \dots F_n$ исчисления предикатов первого порядка. Цель задачи B также записывается в виде утверждения, справедливость которого следует установить или опровергнуть на основании аксиом и правил вывода формальной системы. Тогда решение задачи (достижение цели задачи) сводится к выяснению логического следования (выводимости) целевой формулы B из заданного множества формул (аксиом) $F_1, F_2 \dots F_n$. Такое выяснение равносильно доказательству общезначимости (тождественно-истинности) формулы

Из практических соображений удобнее использовать доказательство от противного, то есть доказывать невыполнимость формулы. На доказательстве от противного основано и ведущее правило вывода, используемое в логическом программировании, - принцип резолюции. Робинсон открыл более сильное правило вывода, чем *modus ponens*, которое он назвал принципом резолюции (или правилом резолюции). При использовании принципа резолюции формулы исчисления предикатов с помощью несложных преобразований приводятся к так называемой дизъюнктивной форме, то есть представляются в виде набора дизъюнктов (смотри табл. 3.1 и рис. 3.19). При этом под дизъюнктом понимается дизъюнкция литералов, каждый из которых является либо предикатом, либо отрицанием предиката.

Главная идея принципа резолюции, как правила вывода, заключается в проверке того, содержит ли множество дизъюнктов R пустой (ложный) дизъюнкт. Обычно резолюция применяется с прямым или обратным методом рассуждения. Прямой метод из посылок $A, A \rightarrow B$ выводит заключение B (правило *modus ponens*). Основным недостатком прямого метода состоит в его не направленности: повторное применение метода приводит к резкому росту промежуточных заключений, не связанных с целевым заключением. Обратный вывод является направленным: из желаемого заключения B и тех же посылок он выводит новое подцелевое заключение A . Каждый шаг вывода в этом случае связан всегда с первоначально поставленной целью. Существенный недостаток метода резолюции заключается в формировании на каждом шаге вывода множества резольвент - новых дизъюнктов, большинство из которых оказывается лишними. В связи с этим разработаны различные модификации принципа резолюции, использующие более эффективные стратегии поиска и различного рода ограничения на вид исходных дизъюнктов. В этом смысле наиболее удачной и популярной является система ПРОЛОГ, которая использует специальные виды дизъюнктов, называемых дизъюнктами Хорна (смотри таблицу 1).

Процесс доказательства методом резолюции (от обратного) состоит из следующих этапов (проверьте правильность выполнения данных этапов с помощью графа на рис. 3.19):

1. Предложения или аксиомы приводятся к дизъюнктивной нормальной форме;
2. К набору аксиом добавляется отрицание доказываемого утверждения в дизъюнктивной форме;
3. Выполняется совместное разрешение этих дизъюнктов, в результате чего получаются новые, основанные на них дизъюнктивные выражения (резольвенты);
4. Генерируется пустое выражение, означающее противоречие;
5. Подстановки, использованные для получения пустого выражения, свидетельствуют о том, что отрицание отрицания истинно.

В методе резолюции порядок комбинации дизъюнктивных выражений не устанавливался. Значит, для больших задач будет наблюдаться экспоненциальный рост числа возможных комбинаций. Поэтому в процедурах резолюции большое значение имеют также эвристики поиска и различные стратегии. Одна из самых простых и понятных стратегий - стратегия предпочтения единичного выражения, которая гарантирует, что резольвента будет меньше, чем наибольшее родительское выражение. Ведь в итоге мы должны получить выражение, не содержащее литералов вообще.

Среди других стратегий стратегия "множества поддержки" (смотри выше), стратегия линейной входной формы показывает отличные результаты при поиске в больших пространствах дизъюнктивных выражений. Суть стратегии такова. Для некоторого набора исходных дизъюнктивных выражений S можно указать подмножество T , называемое множеством поддержки. Для реализации этой стратегии необходимо, чтобы одна из резольвент в каждом опровержении имела предка из множества поддержки. Можно доказать, что если S - невыполнимый набор дизъюнктов, а S и T - выполнимый, то стратегия множества поддержки является полной в смысле опровержения. С другими стратегиями для поиска методом резолюции в больших пространствах дизъюнктивных выражений студент может познакомиться в специальной литературе.

Исследования, связанные с доказательством теорем и разработкой алгоритмов опровержения резолюции, привели к развитию языка логического программирования PROLOG (Programming in Logic). PROLOG основан на теории предикатов первого порядка. Логическая программа - это набор спецификаций в рамках формальной логики. Несмотря на то, что в настоящее время удельный вес языков LISP и PROLOG снизился и при решении задач ИИ все больше используются C, C++ и Java, однако многие задачи и разработка новых методов решения задач ИИ продолжают опираться на языки LISP и PROLOG.

Поиск решений в продукционных системах осуществляется в соответствии с механизмом их работы и представлен в подразделе «Продукционные правила и продукционные системы».

4. Использование экспертных систем для решения технологических задач производства деталей приборов

4.1 Экспертные системы: определение и классификация

Одним из наиболее важных достижений современной науки в области искусственного интеллекта стала разработка мощных компьютерных систем, получивших название «экспертных» или основанных на «знаниях» систем. В современном обществе при решении задач управления сложными многопараметрическими и сильносвязанными системами, объектами, производственными и технологическими процессами приходится сталкиваться с решением неформализуемых либо трудноформализуемых задач. Такие задачи часто возникают в приборостроении как при проектировании деталей приборов, так и в процессе их изготовления.

В большинстве случаев под экспертной системой (ЭС) подразумевается программа, которая использует знания специалистов (экспертов) о некоторой конкретной узко специализированной предметной области и в пределах этой области она способна принимать решения на уровне эксперта-профессионала [17].



Рис. 4.1. Структура экспертной системы

База знаний предназначена для хранения экспертных знаний о предметной области, используемых при решении задач экспертной системой.

База данных предназначена для временного хранения фактов или гипотез, являющихся промежуточными решениями или результатом общения системы с внешней средой, в качестве которой обычно выступает человек, ведущий диалог с экспертной системой. Машина логического вывода – механизм рассуждений, оперирующий знаниями и данными с целью получения новых данных из знаний и других данных, имеющихся в рабочей памяти. Для этого обычно используется программно реализованный механизм дедуктивного логического вывода (какая-либо его разновидность) или механизм поиска решения в сети фреймов или семантической сети.

Машина логического вывода может реализовывать рассуждения в виде:

- 1) дедуктивного вывода (прямого, обратного, смешанного);
- 2) нечеткого вывода;
- 3) вероятностного вывода;
- 4) унификации (подобно тому, как это реализовано в Прологе);
- 5) поиска решения с разбиением на последовательность подзадач;
- 6) поиска решения с использованием стратегии разбиения пространства поиска с учетом уровней абстрагирования решения или понятий, с ними связанных;
- 7) монотонного или немонотонного рассуждения;
- 8) рассуждений с использованием механизма аргументации;
- 9) ассоциативного поиска с использованием нейронных сетей;
- 10) вывода с использованием механизма лингвистической переменной.

Подсистема общения служит для ведения диалога с пользователем, в ходе которого ЭС запрашивает у пользователя необходимые факты для процесса рассуждения, а также дающая возможность пользователю в какой-то степени контролировать и корректировать ход рассуждений экспертной системы. Подсистема объяснений необходима для того, чтобы дать возможность пользователю контролировать ход рассуждений и, может быть, учиться у экспертной системы. Если нет этой подсистемы, экспертная система выглядит для пользователя как «вещь в себе», решениям которой можно либо верить, либо нет. Нормальный пользователь выбирает последнее и такая ЭС не имеет перспектив для использования.

Подсистема приобретения знаний служит для корректировки и пополнения базы знаний. В простейшем случае - это интеллектуальный редактор базы знаний, в более сложных экспертных системах – средства для извлечения знаний из баз данных, неструктурированного текста, графической информации и т.д.

4.2 Технологические задачи, решаемые при помощи экспертных систем

Основными категориями решаемых ЭС задач являются: диагностика, управление (в том числе технологическими процессами), интерпретация, прогнозирование, проектирование, отладка и ремонт, планирование, наблюдение (мониторинг), обучение.

Экспертные системы целесообразно использовать тогда, когда: 1) разработка ЭС возможна; 2) оправдана; 3) методы инженерии знаний соответствуют решаемой задаче.

Рассмотрим более подробно эти условия.

Разработка ЭС возможна когда:

- существуют эксперты в данной области;
- эксперты должны сходиться в оценке предлагаемого решения;
- эксперты должны уметь выразить на естественном языке и объяснить используемые методы;
- задача требует только рассуждений, а не действий;
- задача не должна быть слишком трудной, ее решение должно занимать у эксперта до нескольких часов или дней, а не недель или месяцев;
- задача должна относиться к достаточно структурированной области;
- решение не должно использовать в значительной мере здравый смысл (т.е. широкий спектр общих сведений о мире и о способе его функционирования).

Разработка ЭС оправдана, если:

- решение задачи принесет значительный эффект;
- использовать человека-эксперта невозможно из-за ограниченного количества экспертов или из-за необходимости выполнения экспертизы одновременно во многих местах;
- при передаче информации эксперту происходит значительная потеря времени или информации;
- необходимо решать задачу в окружении, враждебном человеку.

Методы инженерии знаний соответствуют задаче, если задача обладает следующими характеристиками:

- может быть естественным образом решена посредством манипуляции с символами, а не с числами;
- имеет эвристическую природу, т.е. не годится задача, которая может быть решена гарантированно с помощью некоторых формальных процедур;
- должна быть достаточно сложной, чтобы оправдать затраты, но не чрезмерно сложной;
- должна быть достаточно узкой, но практически значимой.

4.3 Этапы проектирования экспертных систем

В проектировании экспертных систем можно выделить следующие этапы [14].

1. Идентификация.

1.1. Определение участников и их ролей в процессе создания и эксплуатации экспертной системы. В процессе создания экспертной системы могут участвовать следующие специалисты: инженеры по знаниям, эксперты, программисты, руководитель проекта, заказчики (конечные пользователи). При реализации сравнительно простых экспертных систем программистов может не быть. Роль инженера по знаниям – выуживание профессиональных знаний из экспертов и проектирование базы знаний экспертной системы и ее архитектуры. Программист необходим при разработке специализированного для данной экспертной системы программного обеспечения, когда подходящего стандартного (например, оболочки для создания экспертных систем) не существует или его возможностей не достаточно и требуются дополнительные модули. В процессе эксплуатации могут принимать участие конечные пользователи, эксперты, администратор.

1.2. Идентификация проблемы.

На этом этапе разработчики должны ответить на ряд вопросов, определяющих особенности решаемых экспертами, а, следовательно, будущей экспертной системой, задач. Эти особенности определяют и особенности архитектуры экспертной системы, формируемой на последующих этапах. К этим вопросам относятся следующие:

- какой класс задач должна решать ЭС;
- как эти задачи могут быть охарактеризованы или определены;
- какие можно выделить подзадачи;
- какие исходные данные должны использоваться для решения;
- какие понятия и взаимосвязи между ними используются при решении задачи экспертами;
- какой вид имеет решение и какие концепции используются в нем;
- какие аспекты опыта эксперта существенны для решения задачи;
- какова природа и объем знаний, необходимых для решения задачи;
- какие препятствия встречаются при решении задач;
- как эти помехи могут влиять на решение задачи.

1.3. Определение необходимых ресурсов – временных, людских, материальных.

1.4. Определение целей.

В качестве целей, преследуемых при создании экспертных систем, могут быть: повышение скорости принятия решения, повышение качества решений, тиражирование опыта экспертов и т.п.

2. Концептуализация.

На этом этапе разработчики должны ответить на следующие вопросы:

- какие типы данных нужно использовать;

- что из данных задано, а что должно быть выведено;
 - имеют ли подзадачи наименования;
 - имеют ли стратегии наименования;
 - имеются ли ясные частичные гипотезы, которые широко используются.
3. Формализация.
 4. Реализация прототипной версии.
 5. Тестирование.
 6. Перепроектирование прототипной версии.

По степени готовности к использованию и распространению различают четыре прототипа экспертных систем:

1) демонстрационный; предназначен для демонстрации возможностей будущей экспертной системы, основных архитектурных решений, пользовательского интерфейса, для уточнения требований к пользовательскому интерфейсу и функциям, выполняемым экспертной системой, содержит демонстрационную далеко неполную базу знаний;

2) исследовательский; предназначен для исследования направлений дальнейшего совершенствования экспертной системы и для пополнения базы знаний, может использоваться для решения реальных задач в ограниченных пределах;

3) промышленный; предназначен для использования, как правило, в организации, где был разработан, в нем возможны ограничения, условности, специализация, свойственные для данной организации;

4) коммерческий; предназначен для широкого распространения, обладает гибкостью, удобством в эксплуатации, адаптируемостью к конкретным задачам и требованиям пользователя.

Жизненный цикл экспертной системы состоит из этапов разработки и сопровождения. На этапе разработки создается программное обеспечение и база знаний экспертной системы; на этапе сопровождения происходит исправление выявленных ошибок и пополнение базы знаний без участия разработчиков (если последнее допускается архитектурой экспертной системы).

Применение экспертной системы с базой знаний, неизменяемой в процессе эксплуатации, возможно при достаточно стабильной в течение длительного времени предметной области, в которой решаются задачи. Примерами таких предметных областей являются разделы математического анализа, описание правил диагностики различных заболеваний.

Примерами областей применения, требующих гибкости со стороны создания и пополнения базы знаний, являются: планирование производства, проектирование и диагностика в области электроники, вычислительной техники и приборостроения.

4.4. Инструментальные средства создания экспертных систем

Важную роль при создании ЭС играют инструментальные средства. Среди инструментальных средств для создания ЭС наиболее популярны такие языки программирования, как LISP и PROLOG, а также экспертные системы-оболочки (ЭСО): KEE, CENTAUR, G2 и GDA, CLIPS, АТ-ТЕХНОЛОГИЯ, предоставляющие в распоряжение разработчика - инженера по знаниям широкий набор для комбинирования систем представления знаний, языков программирования, объектов и процедур [17].

Разработка ЭС связана с определенными трудностями, которые необходимо хорошо знать, так же как и способы их преодоления [14]. Рассмотрим подробнее эти проблемы.

1. Проблема извлечения знаний экспертов. Ни один специалист никогда просто так не раскроет секреты своего профессионального мастерства, свои сокровенные знания в профессиональной области. Он должен быть заинтересован материально или морально, причем хорошо заинтересован. Никто не хочет рубить сук, на котором сидит. Часто такой специалист опасается, что, раскрыв все свои секреты, он будет не нужен компании. Вместо него будет работать экспертная система. Избежать эту проблему поможет выбор высококвалифицированного эксперта, заинтересованного в сотрудничестве.

2. Проблема формализации знаний экспертов. Эксперты-специалисты в определенной области, как правило, не в состоянии формализовать свои знания. Часто они принимают правильные решения на интуитивном уровне и не могут аргументированно объяснить, почему принято то или иное решение. Иногда эксперты не могут прийти к взаимопониманию (фраза «встретились два геолога, у них было три мнения» - не шутка, а реальная жизнь). В таких ситуациях поможет выбор эксперта, умеющего ясно формулировать свои мысли и легко объяснять другим свои идеи.

3. Проблема нехватки времени у эксперта. Выбранный для разработки эксперт не может найти достаточно времени для выполнения проекта. Он слишком занят. Он всем нужен. У него есть проблемы. Чтобы избежать этой ситуации, необходимо получить от эксперта, прежде чем начнется проект, согласие тратить на проект время в определенном фиксированном объеме.

4. Правила, формализованные экспертом, не дают необходимой точности. Проблему можно избежать, если решать вместе с экспертом реальные задачи. Не надо придумывать «игрушечных» ситуаций или задач. В условиях задач нужно использовать реальные данные, такие как лабораторные данные, отчеты, дневники и другую информацию, взятую из практических задач. Постарайтесь говорить с экспертом на одном языке, используя единую терминологию. Эксперт, как правило, легче понимает правила, записанные на языке, близком к естественному, а не на языке типа LISP или PROLOG.

5. Недостаток ресурсов. В качестве ресурсов выступают персонал (инженеры знаний, разработчики инструментальных средств, эксперты) и средства построения ЭС (средства разработки и средства поддержки).

Недостаток благожелательных и грамотных администраторов порождает скептицизм и нетерпение у руководителей. Повышенное внимание в прессе и преувеличения вызвали нереалистические ожидания, которые приводят к разочарованию в отношении экспертных систем. ЭС могут давать не самые лучшие решения на границе их применимости, при работе с противоречивыми знаниями и в рассуждениях на основе здравого смысла. Могут потребоваться значительные усилия, чтобы добиться небольшого увеличения качества работы ЭС. Экспертные системы требуют много времени на разработку. Так, создание системы PUFF для интерпретации функциональных тестов легковых автомобилей потребовало 5 человеко-лет, на разработку системы PROCPECTOR для разведки рудных месторождений ушло 30 человеко-лет, система XCON для расчета конфигурации компьютерных систем на основе VAX 11/780 потребовала 8 человеко-лет. ЭС последних лет разрабатываются более быстрыми темпами за счет развития технологий ЭС, но проблемы остались. Удвоение персонала не сокращает время разработки наполовину, потому что процесс создания ЭС - это процесс со множеством обратных связей. Все это необходимо учитывать при планировании создания ЭС.

6. Неадекватность инструментальных средств решаемой задаче. Часто определенные типы знаний (например, временные или пространственные) не могут быть легко представлены на одном ЯПЗ, так же как и разные схемы представления (например, фреймы и продукции) не могут быть достаточно эффективно реализованы на одном ЯПЗ. Некоторые задачи могут быть непригодными для решения по технологии ЭС (например, отдельные задачи анализа сцен). Необходим тщательный анализ решаемых задач, чтобы определить пригодность предлагаемых инструментальных средств и сделать правильный выбор.

4.5. Методика создания экспертных систем

Рассмотрим методику формализации экспертных знаний на примере создания экспертных диагностических систем (ЭДС) [17].

Целью создания ЭДС является определение состояния объекта диагностирования (ОД) и имеющихся в нем неисправностей.

Состояниями ОД могут быть: исправно, неисправно, работоспособно. Неисправностями, например, радиоэлектронных ОД являются обрыв связи, замыкание проводников, неправильное функционирование элементов и т.д.

Число неисправностей может быть достаточно велико (несколько тысяч). В ОД может быть одновременно несколько неисправностей. В этом случае говорят, что неисправности кратные.

Введем следующие определения. Разные неисправности ОД проявляются во внешней среде информационными параметрами. Совокупность значений информационных параметров определяет «информационный образ» (ИО) неисправности ОД. ИО может быть полным, то есть содержать всю необходимую информацию для постановки диагноза, или, соответственно,

неполным. В случае неполного ИО постановка диагноза носит вероятностный характер.

Основой для построения эффективных ЭДС являются знания эксперта для постановки диагноза, записанные в виде информационных образов, и система представления знаний, встраиваемая в информационные системы обеспечения функционирования и контроля ОД, интегрируемые с соответствующей технической аппаратурой.

Для описания своих знаний эксперт с помощью инженера по знаниям должен выполнить следующее.

1. Выделить множество всех неисправностей ОД, которые должна различать ЭДС.

2. Выделить множество информативных (существенных) параметров, значения которых позволяют различить каждую неисправность ОД и поставить диагноз с некоторой вероятностью.

3. Для выбранных параметров следует выделить информативные значения или информативные диапазоны значений, которые могут быть как количественными, так и качественными. Например, точные количественные значения могут быть записаны: задержка 25 нсек, задержка 30 нсек и т.д. Количественный диапазон значений может быть записан: задержка 25--40 нсек, 40--50 нсек, 50 нсек и выше. Качественный диапазон значений может быть записан: индикаторная лампа светится ярко, светится слабо, не светится.

Для более удобного дальнейшего использования качественный диапазон значений может быть закодирован, например, следующим образом:

- светится ярко P1 = +++ (или P1 = 3),
- светится слабо P1 = ++ (или P1 = 2),
- не светится P1 = + (или P1 = 1).

Процедура получения информации по каждому из параметров определяется индивидуально в каждой конкретной системе диагностирования. Эта процедура может заключаться в автоматическом измерении параметров в ЭДС, в ручном измерении параметра с помощью приборов, качественном определении параметра, например, светится слабо, и т.д.

4. Процедура создания полных или неполных ИО каждой неисправности в алфавите значений информационных параметров может быть определена следующим образом. Составляются диагностические правила, определяющие вероятный диагноз на основе различных сочетаний диапазонов значений выбранных параметров ОД. Правила могут быть записаны в различной форме. Ниже приведена форма записи правил в виде таблицы (Таблица 4.1).

Таблица 4.1 Диагностические правила

Номер	P1	P2	P3	Диагноз	Вероятность диагноза	Примечания
1		+++		Неисправен Блок А1	0.95	
2	12-15	+		Неисправен Блок А2	0.80	

Для записи правил с учетом изменений по времени следует ввести еще один параметр P0 - время (еще один столбец в таблице). В этом случае диагноз может ставиться на основе нескольких строк таблицы, а в графе Примечания могут быть указаны использованные тесты. Диагностическая таблица в этом случае представлена в Таблице 4.2.

Таблица 4.2 Динамические диагностические правила

Номер	P0	P1	P2	P3	Диагноз	Вероятность Диагноза	Примечания
1	13:00	+	+	+	См.Тест 1	См. Тест 1	Тест 1
2	13:20	++	++	+	Неисправен блок АЗ	0.90	

Для записи последовательности проведения тестовых процедур и задания ограничений (если они есть) на их проведение может быть предложен аналогичный механизм. Механизм записи последовательности проведения тестовых процедур в виде правил реализуется, например, следующим образом:

ЕСЛИ: P2 = 1

ТО: тест = T1, T3, T7 - где T1, T3, T7 - тестовые процедуры, подаваемые на ОД при активизации (срабатывании) соответствующей продукции.

В современных ЭДС применяются различные стратегии поиска решения и постановки диагноза, которые позволяют определить необходимые последовательности тестовых процедур. Однако приоритет в ЭС отдается прежде всего знаниям и опыту, а лишь затем логическому выводу. Данная методика нашла широкое применение при создании экспертных систем управления технологическими процессами.

4.6. Примеры экспертных систем

Ранние экспертные системы, в большинстве случаев, создавались на базе системы продукций (правил) и прямой цепочки рассуждений. Медицинская ЭС MYCIN разработана в Стэнфордском университете в середине 70-х годов для диагностики и лечения инфекционных заболеваний крови. MYCIN в настоящее время используется для обучения врачей. ЭС DENDRAL разработана в Стэнфордском университете в середине 60-х годов для определения топологических структур органических молекул. Система выводит молекулярную структуру химических веществ по данным масс-спектрометрии и ядерного магнитного резонанса. ЭС PROSPECTOR разработана в Стэнфордском университете в 1974 - 1983 годах для оценки геологами потенциальной рудоносности района. Система содержит более 1000 правил и реализована на INTERLISP. Программа сравнивает наблюдения геологов с моделями разного рода залежей руд. Программа вовлекает геолога в диалог для извлечения дополнительной информации. В 1984 году она точно предсказала существование молибденового месторождения, оцененного в многомиллионную сумму [17].

Рассмотрим экспертную систему диагностирования (ЭСД) цифровых и цифроаналоговых устройств, в которой использовались системы продукций и фреймы, а также прямая и обратная цепочка рассуждений одновременно. В качестве объекта диагностирования (ОД) в ЭСД могут использоваться цифровые устройства (ЦУ), БИС, цифро-аналоговые устройства. На рис. 4.2 показано, что такая ЭСД работает совместно с автоматизированной системой контроля и диагностирования (АКД), которая подает в динамике воздействия на ОД (десятки, сотни и тысячи воздействий в секунду), анализирует выходные реакции и дает заключение: годен или не годен. В случае, если реакция проверяемого ОД не соответствует эталонным значениям, то подключается основанная на знаниях подсистема диагностирования. ЭСД запрашивает значения сигналов в определенных контрольных точках и ведет оператора по схеме ОД, рекомендуя ему произвести измерения в определенных контрольных точках или подтвердить промежуточный диагноз и, в результате приводит его к месту неисправности. Исходными данными для работы ЭСД являются результаты машинного моделирования ОД на этапе проектирования. Эти результаты моделирования передаются в ЭСД на магнитных носителях в виде тысяч продукционных правил. Движение по контрольным точкам осуществляется на основе модели, записанной в виде сети фреймов для ОД.

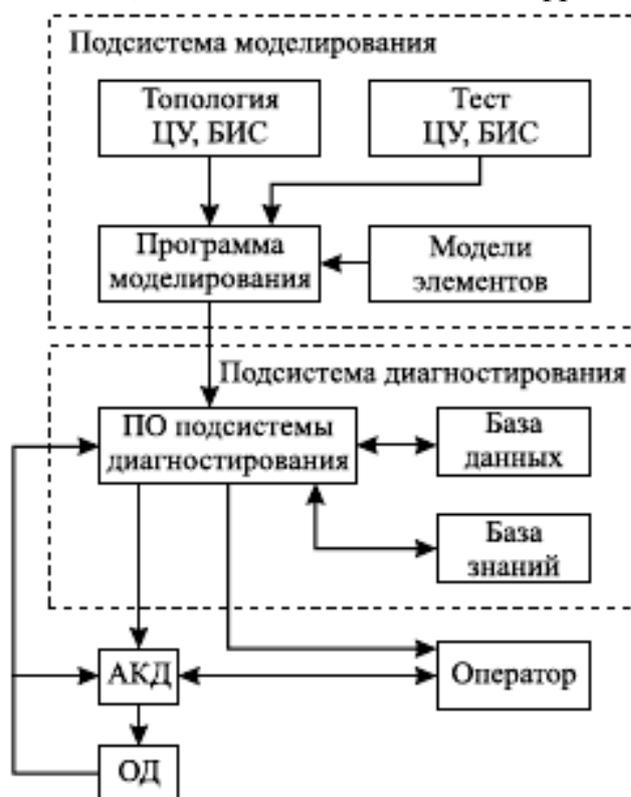


Рис. 4.2 Общая структура экспертной системы диагностирования

Такая ЭСД не была бы интеллектуальной системой, если бы она не накапливала опыт. Она запоминает найденную неисправность для данного типа ОД. В следующий раз при диагностике неисправности ОД этого типа она предлагает проверить сразу же эту неисправность, если реакция ОД говорит о том, что такая неисправность возможна. Так поступают опытные мастера

радиоэлектронной аппаратуры (РЭА), знающие «слабые» места в конкретных типах РЭА и проверяющие их в первую очередь. ЭСД накапливает вероятностные знания о конкретных неисправностях с целью их использования при логическом выводе. При движении по дереву поиска решений на очередном шаге используется критерий - максимум отношения вероятности (коэффициента уверенности) постановки диагноза к трудоемкости распознавания неисправности. Коэффициенты уверенности автоматически корректируются во время работы ЭСД при каждом подтверждении или не подтверждении диагноза для конкретных ситуаций диагностирования. Трудоемкости элементарных проверок первоначально задаются экспертом, а затем автоматически корректируются в процессе работы ЭСД. ЭСД не была реализована в виде интеллектуальной робототехнической системы по экономическим соображениям. Небольшая серийность проверяемой аппаратуры, недостаточная унификация и дешевая рабочая сила (последний фактор и в наше время играет в России немаловажную роль) помешали реализовать полностью автоматическое диагностирование.

Среди современных коммерческих систем необходимо выделить экспертную систему - оболочку G2 американской фирмы Gensym (USA) как непревзойденную экспертную коммерческую систему для работы с динамическими объектами. Работа в реальном времени с малыми временами ответа часто необходима при анализе ситуаций в корпоративных информационных сетях, на атомных реакторах, в космических полетах и множестве других задач. В этих задачах необходимо принимать решения в течение миллисекунд с момента возникновения критической ситуации. ЭС G2, предназначенная для решения таких задач, отличается от большинства динамических ЭС такими характерными свойствами, как:

- работа в реальном времени с распараллеливанием процессов рассуждений;
- структурированный естественно-языковой интерфейс с управлением по меню и автоматической проверкой синтаксиса;
- обратный и прямой вывод, использование метазнаний, сканирование и фокусирование;
- интеграция подсистемы моделирования с динамическими моделями для различных классов объектов;
- структурирование БЗ, наследование свойств, понимание связей между объектами;
- библиотеки знаний являются ASCII-файлами и легко переносятся на любые платформы и типы ЭВМ;
- развитый редактор для сопровождения БЗ без программирования, средства трассировки и отладки БЗ;
- управление доступом с помощью механизмов авторизации пользователя и обеспечения желаемого взгляда на приложение;
- гибкий интерфейс оператора, включающий графики, диаграммы, кнопки, пиктограммы и т.п.;
- интеграция с другими приложениями (по TCP/IP) и базами данных,

возможность удаленной и многопользовательской работы.

Создание экспертной системы управления технологическим процессом (ТП) может значительно ускорить процесс разработки сложной системы управления ТП, повысить качество решения задачи и дать экономию ресурсов за счет эффективного распределения функций центрального управления и локальных измерительных и управляющих подсистем. Такой эффект достигается за счет открытости системы представления знаний об объекте управления, адаптивности системы к условиям функционирования, автоматической коррекции управляющих воздействий при изменении существенных параметров в процессе функционирования.

5. Использование робототехнических систем с элементами искусственного интеллекта в приборостроении

5.1. Область робототехники

Область робототехники охватывает достаточно широкий класс машин, начиная от простейших игрушек до полностью автоматизированных производств (автоматически управляемые электростанции, беспилотные космические корабли, автоматические подводные аппараты, ЭВМ, играющая в шахматы - все эти системы можно считать роботами) [12]. Поэтому термин «робот» имеет весьма широкий подтекст. В представленном методическом пособии основное внимание уделяется промышленным роботам, в которых присутствуют элементы интеллектуальной деятельности.

Создание "разумных" роботов связано, как правило, с приданием им человеческих качеств. Это способность распознавать образы, участвовать в интерактивных операциях, ставить задачи и принимать решения. Поэтому в дальнейшем следует остановиться более подробно на детальном рассмотрении подсистем низшего уровня, выполняющих технологические операции обработки заготовок, и связь данных подсистем с подсистемами высшего уровня. В целом стоит отметить, что применение вычислительной техники в системах управления и программного обеспечения позволяет реализовать интеллектуальные способности человека и заменить его в сфере оценки ситуации и принятия решений. Совокупность интеллектуальных и механических способностей робототехнической системы позволяет заменить человека в сфере его производственной деятельности.

Основными чертами промышленных роботов являются их механическое совершенство и непрерывно растущий интеллект, что расширяет сферу применения робототехники, которая уже охватывает промышленность, сельское хозяйство, транспорт, медицину, научные исследования практически во всех областях знаний. Структура робота, взаимодействующего с окружающей средой, может быть упрощенно проиллюстрирована рис. 5.1.

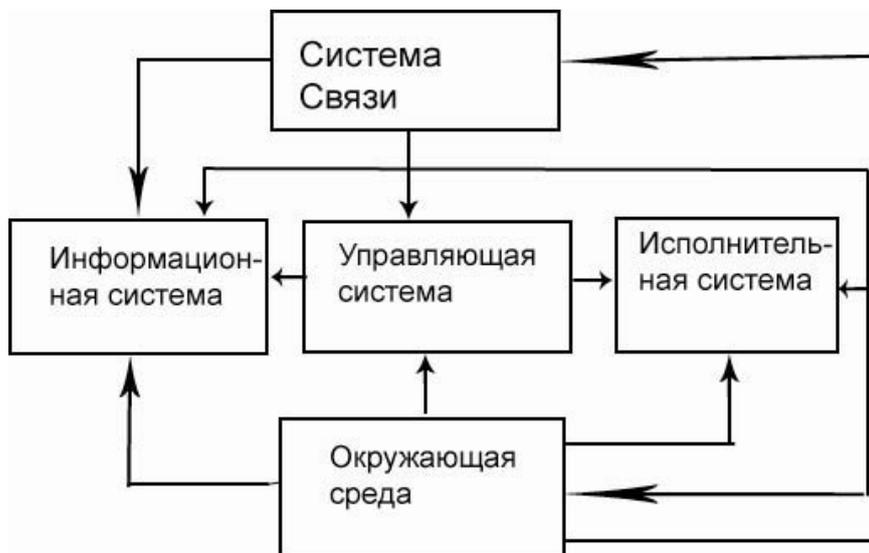


Рис. 5.1 Обобщённая структура робота, взаимодействующего с окружающей средой

В общем случае робот состоит из четырех систем [20]:

- исполнительной (манипуляционной) - для целенаправленного воздействия на окружающую среду;
- информационно-измерительной (сенсорной) - для обеспечения робота информацией о состоянии внешней среды, результатах воздействия на нее манипуляционной системы (или взаимодействия системы робот - объект - среда) и состояния самого робота в соответствии с требованиями управляющей системы;
- управляющей системы (интеллекта) - для выработки закона управления манипуляционной системой на основании данных, поступающих от информационной системы, а также для организации общения робота с человеком или другими функциональными устройствами, с которыми взаимодействует робот; интеллектуальные способности робота определяются главным образом алгоритмическим и программным обеспечением его управляющей системы;
- системы связи - для организации обмена информацией между роботом и человеком или другими функциональными устройствами (в том числе роботами) на некотором понятном им языке.

Уровни интеллекта и информационного обеспечения промышленного робота определяются характеристикой окружающей среды, с которой взаимодействует (на которую воздействует) исполнительная система робота. В робототехнических системах под окружающей средой понимаются объекты роботизации, которые могут находиться в неупорядоченном (и тогда среда является неподготовленной), упорядоченном (подготовленная среда) и частично упорядоченном состоянии. Снижение информационной нагрузки и уровня интеллекта робота возможно, если заранее подготовить среду путем упорядочения объектов по классам, ориентации в пространстве и относительно друг друга.

Одной из важных проблем робототехники является анализ роботизируемого производства, предусматривающий этапы подготовки среды

(упорядочения объектов) специальными средствами. Возможность упорядочения объектов рассматривается при этом как один из основных фактов оценки технологичности объектов под роботизированное изготовление. Особенно важным является использование средств и методов упорядочения среды при роботизации многономенклатурного мелкосерийного производства, когда отсутствие таких средств обуславливает слишком разветвленную систему информационного обеспечения, что удорожает робототехническую систему в целом, снижает надежность ее функционирования и повышает затраты на эксплуатацию роботов.

Гибкость автоматизированных производственных процессов, особенно в условиях мелкосерийного многономенклатурного производства, можно обеспечить путем их организации по модульному принципу. Основу последнего составляют элементы нижнего уровня структуры гибких автоматизированных производств - робототехнологические комплексы. Такие производства представляют собой комплексные объединения станков с числовым программным управлением, другого технологического (в том числе и сборочного) оборудования и роботов, управляемых микропроцессорами, с системами автоматизированной подготовки производства, включая его технологический анализ, проектирование изделий и разработку технологии.

5.2. Системы искусственного интеллекта в робототехнике

Искусственный интеллект промышленных комплексов заключается в возможности распознавать детали и их поверхности с точки зрения качества и соответствия заданным геометрическим размерам по чертежу, управлять технологическим процессом и принимать решения по его изменению [8]. В свою очередь, принятие решения включает формирование промежуточных целей для выполнения поставленной задачи.

Современная быстродействующая вычислительная техника позволила качественно изменить структуру технологического оборудования. Во-первых, благодаря высокому быстродействию вычислений появилась возможность осуществлять управление механизмами, в которых перемещения не совпадают с координатами изготавливаемой детали. Например, высоко скоростные прямолинейные перемещения можно выполнять с помощью вращательных пар. Во-вторых, быстродействующие средства контроля дали возможность построить системы оперативной настройки режимов обработки, получая информацию об обрабатываемой поверхности.

В робототехнике системы оучувствления и искусственного интеллекта нашли достаточно широкое применение. Следует выделить следующие направления развития интеллектуальных роботов.

1. Промышленные роботы, работающие в производственной сфере и заменяющие человека при выполнении технологических операций. Интеллект указанных роботов заключается в их способности автоматически распознавать качество обработанной поверхности, контролировать режимы обработки и

корректировать их в зависимости от поставленной цели, например, минимизировать погрешности, уменьшать энергозатраты, выбирать технологию обработки в зависимости от типа детали и требований к ее выходным характеристикам. В настоящее время это основной класс роботов, которому должно быть уделено особое внимание, так как замена человека в сфере производства качественно изменит его жизнедеятельность.

2. Игровые роботы;

3. Специальные роботы, способные работать в военной обстановке, а также в условиях особо опасных для жизнедеятельности человека.

Интеллектуальные робототехнические системы для выполнения производственных задач, так называемые роботы-станки, являются устройствами, которые полностью автоматизируют производство по выпуску определенного вида продукции. Данное оборудование оснащается системами контроля технологических и выходных параметров обрабатываемого изделия. К станочному оборудованию предъявляются достаточно высокие требования по точности, надежности и ответственности выполняемой операции. При выполнении операций обработки и сборки сложных изделий невозможно требовать вероятностного результата. Как правило, такие операции строго детерминированы. Поэтому вероятностные поисковые методы возможны только на стадии обработки результатов. Принятие окончательного решения должно обеспечивать детерминированный результат, обеспечивающий поставленную цель. Особенно высокие требования предъявляются при обработке поверхностей сложной формы. В этом случае необходимо более точное выполнение режимов обработки, контроль износа инструмента в процессе обработки и обеспечение одновременно нескольких параметров детали. В частности, для каждой точки поверхности нужно одновременно обеспечивать до шести геометрических параметров, не считая качества поверхности. Для сложных поверхностей, кроме требований к самим координатам, накладываются условия и на их производные. Для соблюдения высоких требований к точности изготовления деталей необходимо осуществлять постоянный контроль геометрических параметров станка, размеров звеньев, температурных изменений и других параметров. Применение механизмов параллельной структуры также качественно меняет подход к проектированию станочного робототехнического оборудования.

5.3. Современное робототехническое технологическое оборудование с элементами искусственного интеллекта

Понятие робот-станок было введено в 1992 году при описании станочного оборудования, построенного на механизмах параллельной структуры и позволяющего посредством одного и того же механизма выполнять транспортные операции и операции обработки [2]. Данные механизмы позволяют расширить функциональные возможности станочного оборудования и, при наличии системы управления, оснащенной элементами

искусственного интеллекта, делает данное оборудование близким к интеллектуальным роботам. На рис. 5.2. представлен робот-станок, предназначенный для выполнения механической обработки.

Совмещение функций особенно актуально для сложных высокоточных операций, когда требуется изготовление детали от одной базы. В данном случае получаем универсальное оборудование, позволяющее выполнять несколько различных технологических операций для широкой номенклатуры изделий. Главной отличительной особенностью робота-станка от обрабатывающего центра является универсальность, точнее, более богатые кинематические возможности перемещения механизмов. Безусловно, из набора роботов-станков можно построить распределенный обрабатывающий центр. Механизмы параллельной структуры расширили возможности исполнительных механизмов станков, сделали их более облегченными и универсальными. Наличие параллельных кинематических цепей позволяет управлять одним выходным звеном по нескольким параллельным каналам, обеспечивая одновременное управление по положению, скорости, более высоким производным, а также по силе.



Рис. 5.2. Робот-станок

В настоящее время промышленность уже освоила выпуск большой номенклатуры интеллектуальных робототехнических систем, которые довольно быстро заняли лидирующее место в сфере промышленной робототехники. При этом роботы всё чаще представляются в виде технологических систем, непосредственно выполняющих операцию обработки. Как уже было сказано выше, такие системы называются роботами-станками, так как их кинематическая схема позволяет выполнять транспортные операции и

непосредственно обработку. Применение механизмов параллельной структуры уже на низшем уровне позволяет расширить интеллектуальные возможности технологических машин.

5.4. Структура и состав интеллектуальной робототехнической системы

Интеллектуальная робототехническая система включает объект управления совместно со средой, в которой она работает. Структура робототехнической системы представлена на рис. 5.3. Объект управления представляет непосредственно механизмы перемещения инструмента и изделия. В состав манипуляторов входят исполнительные двигатели, которые осуществляют их перемещение по заданным законам R_d и R_i (Рис. 5.3). Информация о положении выходных звеньев манипуляторов определяется датчиками, расположенными в шарнирах звеньев манипуляторов, которые получают информации о выходных координатах механизмов перемещения, их скоростях, ускорениях и силах.

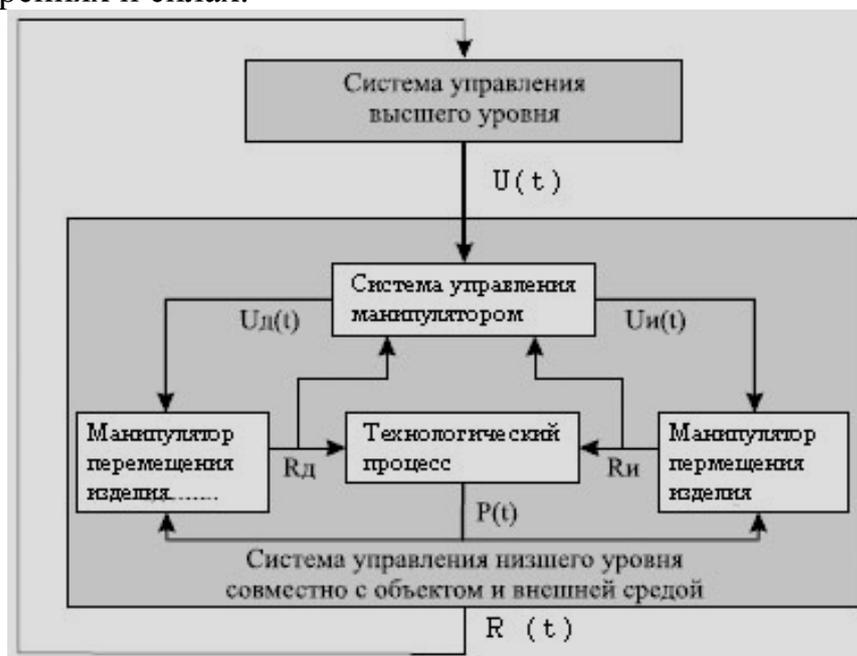


Рис. 5.3. Структура робототехнической системы, где $U(t)$ – управляющий сигнал, $U_d(t)$ и $U_i(t)$ – законы перемещения исполнительных манипуляторов, R_d и R_i – выходные координаты манипуляторов, $P(t)$ – создаваемое усилие и $R(t)$ – управляющий сигнал.

Основная функция системы управления манипуляторами состоит в формировании законов перемещения исполнительными механизмами манипуляторов в реальном времени $U_{i(t)}$ и $U_{d(t)}$ (Рис. 5.3). Данные системы обычно работают в следящем режиме, обеспечивающем выполнение каждой степенью подвижности манипуляторов заданной траектории перемещения с требуемыми точностью, скоростью и усилием. Выходными координатами манипуляторов являются R_d и R_i (Рис. 5.3). В результате взаимодействия инструмента с заготовкой создается усилие $P(t)$, которое воздействует на

исполнительные органы манипуляторов. Применительно к рассматриваемой системе в качестве объекта управления и внешней среды следует рассматривать манипуляторы перемещения изделия, инструмента и непосредственно сам технологический процесс.

На рис. 5.3. не раскрывается состав подсистемы управления высшего уровня. Ее структура и выполняемые функции подробно описаны в главе 1. Общим информационным управляющим каналом на систему управления низшего уровня является канал передачи управляющих сигналов $U(t)$ с обратной связью от системы низшего уровня - сигнал $R(t)$. Управляющее воздействие $U(t)$ представляет выбранную программу действия из некоторого множества $U \in U(t)$ и соответствующую заданной детали, либо обработке заданной поверхности детали. Какую из программ следует выбрать, решается системой высшего уровня как на основе информации от системы распознавания поверхности, так и на основе указаний оператора, управляющего робототехнической системой. Выбранная программа $U(t)$ задается непрерывно в реальном масштабе времени. Обратная связь $R(t)$ может нести полную информацию о работе системы управления низшего уровня в виде логических сигналов о ее состоянии, непрерывную информацию о геометрических размерах, качестве обработки поверхности детали и информацию о состоянии внешней среды, например, о температуре окружающей среды или двигателей, о состоянии сопутствующих обработке других устройств.

Представленная на рисунке 5.3. система является обобщенной для технологических машин широкого назначения. Более детальное представление данной системы рассмотрим на примере системы управления робота-станка (рис. 5.4).

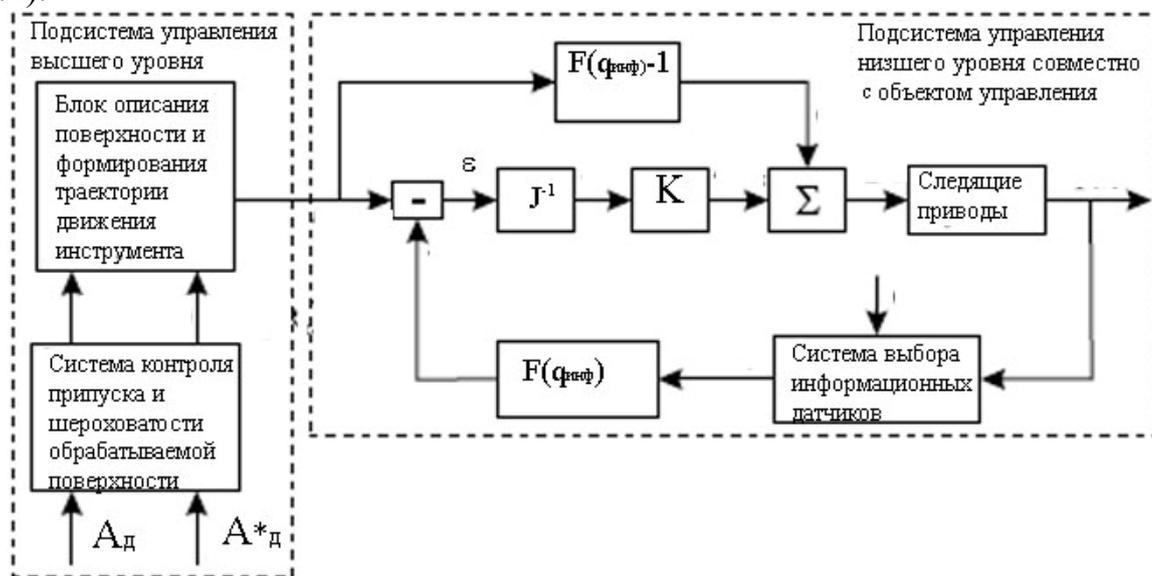


Рис. 5.4. Система управления робота-станка

Отличительной особенностью рассматриваемой следящей системы управления от существующих станочных систем является наличие главной обратной связи по результату обработки поверхности (вычисление $A_{дi}^*(t)$). Вычисление $A_{дi}^*(t)$ осуществляется в системе координат детали решением прямой задачи о положении $F(q_{инф.})$ по информации датчиков, располагаемых в

сочленениях звеньев механизма. Погрешность ε вычисляется сравнением программного значения управляющего воздействия $A_i(t)$ и вычисленного реального его значения $A_{дi}^*(t)$. Обратный Якобиан J и устройство K выполняют функции преобразования и решения линейной задачи вычисления приращений обобщенных координат q_i . Суммируя приращения на каждом шаге вычисления с предыдущим значением, формируется управляющее воздействие на исполнительные приводы q_i .

В качестве электродвигателей приводов манипуляторов применяются безредукторные и высокомоментные электродвигатели. Это требует применения методики синтеза приводов с учетом переменности моментов инерции, а для многостепенной механической системы требуется также учитывать взаимовлияние приводов по степеням подвижности. Подсистема управления высшего уровня выполняет следующие функции. Получая информацию от оптической системы о состоянии обрабатываемой поверхности и ее геометрических размерах, данная подсистема выбирает требуемую программу обработки из некоторого детерминированного множества программ либо при ее отсутствии на основе анализа принимает наиболее близкую по критерию точности воспроизведения требуемой поверхности.

Оптические средства контроля геометрических размеров припуска и качества обработки (шероховатости) поверхности детали позволяют оптимизировать режимы резания. В работе [6] приведено описание оптической системы, построенной с применением специальной решетки и источника монохроматического света. В данной работе дается описание системы, рассматриваются вопросы построения системы распознавания зон с заданным качеством обработки и формирования на этой основе новой программы обработки поверхности.

Формирование программной траектории перемещения инструмента относительно обрабатываемой поверхности $A_{дi}^*(t)$, производится на основе информации, полученной от оптической системы контроля поверхности и экспертной оценки при выборе режимов обработки. Информация о геометрических размерах полученной после обработки поверхности контролируется оптической системой контроля. Эта система формирует также данные о качестве обрабатываемой поверхности. В зависимости от этой информации выбирается ограниченная область обработки поверхности.

Математическая модель объекта управления совместно с окружающей средой, формируемая в системе высшего уровня на основе информации, получаемой от датчиков, включает: чертеж детали с реальными геометрическими размерами, чертеж требуемой идеальной детали и набор параметров, определяющих режимы обработки. Указанная модель позволяет, проигрывая различные ситуации, представляющие набор процедур для выполнения обработки, выбирать цель и формировать программу обработки $U(t)$.

6. Применение в приборостроении современного технологического оборудования с элементами искусственного интеллекта и интеллектуальных систем

6.1. Особенности современного технологического оборудования

Детали, имеющие сложную конфигурацию, составляют большую группу и в приборостроении и в машиностроении. Это в первую очередь высокоточные мехатронные изделия, детали станков и различных компонентов космических и военных приборов. Технологическое оборудование, применяемое в настоящее время для механической обработки сложных поверхностей, это, как правило, обрабатывающие центры, оснащенные числовой системой управления, либо специализированные станки, выполняющие перемещение заготовки или инструмента по жесткой, не перестраиваемой траектории. Для выполнения операций по переноске заготовок и готовых деталей станки оснащаются специальными манипуляционными механизмами. Одним из недостатков процесса создания оборудования является традиционно принятая последовательность, когда первоначально конструируется механическая часть, затем для разработанной механики подбирается система управления, обеспечивающая требуемые законы перемещения и технологические режимы обработки. При этом получаем систему, у которой оптимально, но отдельно проектируемые части не всегда оказываются оптимальными в целом для всей системы. Таким образом, основная тенденция развития современного технологического оборудования заключается в единстве механики и управления.

Современная вычислительная техника, обладающая высоким быстродействием, позволяет по-новому подойти к созданию технологической машины. Система управления совместно с датчиками информации способна исправлять "недостатки" механической части технологической машины. Поэтому технологическую машину необходимо рассматривать как единую систему, включающую механическую часть, технологический процесс и непосредственно систему управления. Обычно в такой схеме управления манипулятор изделия осуществляет его перемещение по заданным координатам, манипулятор инструмента перемещает инструмент по своим заданным координатам, и, в результате их взаимного перемещения, выполняется обработка и технологический процесс воздействует на оба манипулятора.

В настоящее время наиболее перспективными конструкциями манипуляторов, обеспечивающих требуемые функциональные возможности и необходимую зону обслуживания, являются подвижные стержневые

механизмы. Подвижные стержневые механизмы манипуляторов перемещения инструмента и обрабатываемой заготовки, построенные на основе разомкнутой или замкнутой кинематической цепи, позволяют создать облегченные конструкции манипуляторов, обладающие высокой мобильностью при одновременном выполнении транспортных и обрабатывающих операций [6].

Широкое распространение в приборостроении получили так называемые «многорукие» манипуляторы (Рис. 6.1).

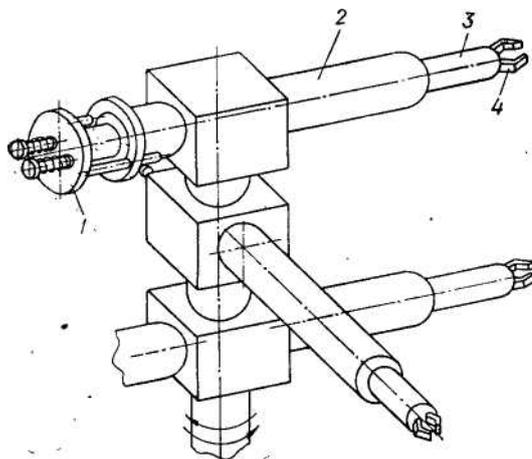


Рис. 6.1 «Многорукий» манипулятор

В «многоруком» манипуляторе, предназначенном для одновременного захвата, передачи и установки нескольких изделий, использованы своеобразные упоры, ограничивающие перемещения его рук, образованных пневмоцилиндрами 2. Эти упоры гасят кинетическую энергию не всего подвижного звена - штока 3, несущего захватное устройство 4, а лишь небольшого по массе диска 1, служащего заслонкой смонтированного в штоке пневматического усилителя. Тем самым резко снижаются перегрузки манипулятора и его отдельных элементов, повышается качество его работы и точность позиционирования захватного устройства. Особенностью манипулятора является также то, что шток служит преобразователем, который не только обрабатывает управляющие команды, но и вырабатывает эти команды, что позволяет упростить всю конструкцию и сделать ее более компактной (Рис. 6.2).



Рис. 6.2. 5-ти координатный манипулятор R17

Представленный на рис. 6.2. пятикоординатный манипулятор отличается не только высоким уровнем гибкости, но и позволяет, благодаря сочленённой конструкции, максимально увеличить зону охвата робототехнического комплекса.

Также широко представлены на робототехническом рынке универсальные 4-х координатные цилиндрические манипуляторы, которые отличаются высокой надёжностью и универсальностью. Представленный на рис.6.3. манипулятор может использоваться практически в любом робототехническом комплексе - от сборочных комплексов до покрасочных автоматов.

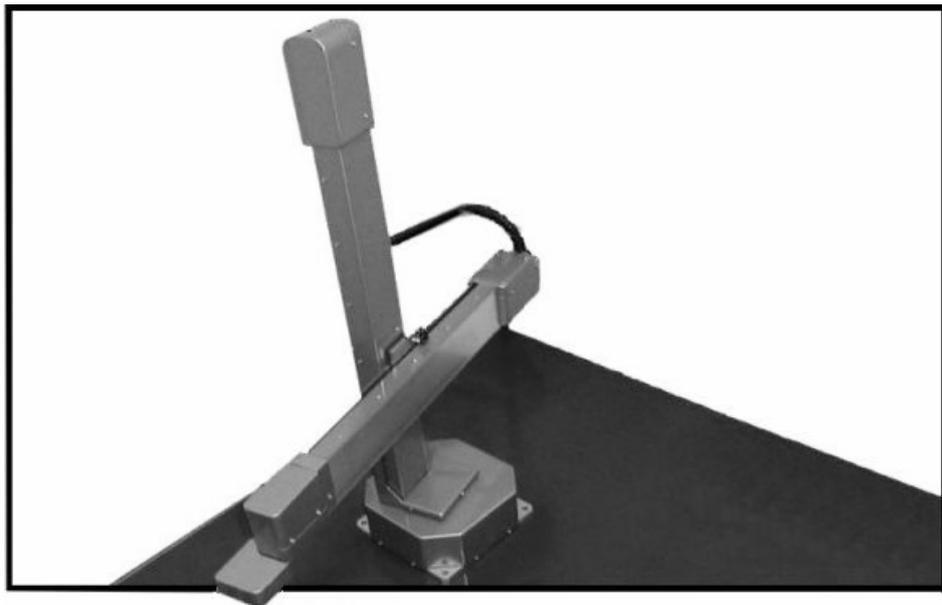


Рис. 6.3. 4-х координатный цилиндрический манипулятор

Механизмы параллельной структуры, которые всё чаще используются при конструировании робототехнических комплексов, характеризуются тем, что выходное звено соединено с основанием несколькими кинематическими цепями, работающими параллельно в смысле передачи движения [13]. Такие механизмы позволяют параллельно управлять усилием, скоростью и перемещением по одной координате выходного звена. Кроме того, в каждой кинематической цепи имеются свободные, не содержащие приводов, сочленения, в которых могут быть установлены дополнительные датчики, а также дополнительные приводы, работающие параллельно с основными.

На рис. 6.4. представлен манипулятор с использованием параллельной структуры. Стоит отметить, что дополнительные датчики и приводы не прибавляют интеллекта системе в целом, но существенно расширяют ее функциональные возможности.

Таким образом, технологические машины, построенные на механизмах относительного манипулирования и оснащенные современными управляющими вычислительными системами, позволяют выполнять механическую обработку заготовок со сложными поверхностями, имеют существенно меньшую металлоемкость в сравнении с оборудованием,

применяемым в настоящее время для этих целей, позволяют одними и теми же механизмами выполнять транспортные и обрабатывающие операции.

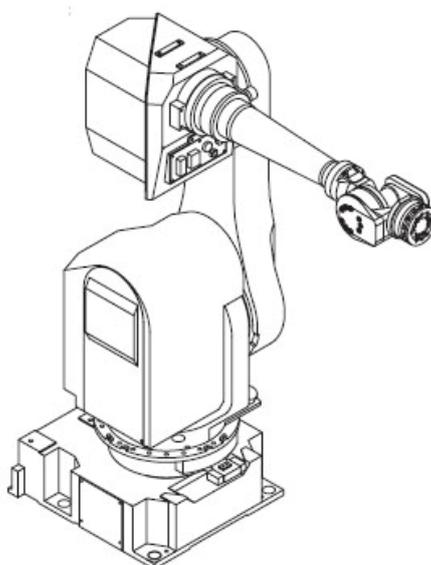


Рис. 6.4. Манипулятор робототехнической системы, с использованием параллельной структуры компоновки элементов

6.2. Особенности разработки технологических процессов для интеллектуальных робототехнических систем

Технологический процесс механической обработки рассматривается как подсистема (рис. 6.5), входом которой являются перемещения манипуляторов R_d и R_i , а выходом - результат данных перемещений, сила резания $P_{рез}$ [2]. Для механической обработки в качестве упрощенной модели может рассматриваться зависимость между скоростью перемещения инструмента относительно заготовки V_p и усилием резания $P_{рез}$.



Рис. 6.5 Система управления интеллектуальным робототехническим комплексом

Для оптимального управления технологическим комплексом как единой системой необходимо задать критерии, характеризующие качество выполнения технологической операции. Такими критериями, например, для абразивной обработки являются шероховатость поверхности R_z , глубина прожигов поверхностного слоя обработанной поверхности h_n и максимальный сьем материала в направлении перемещения инструмента относительно детали $Q_{рез}$. (рис. 6.6 (а)).

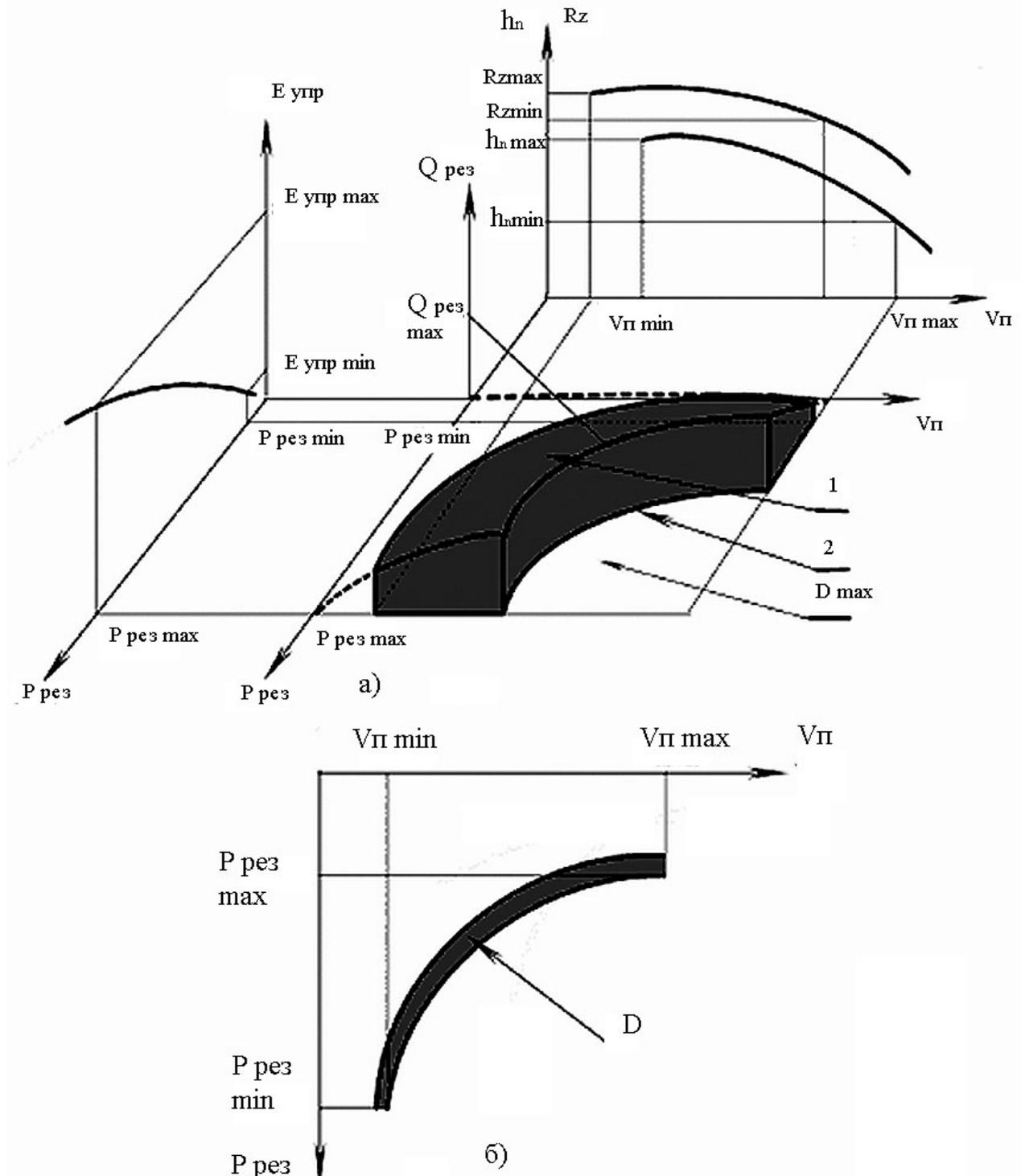


Рис. 6.6 а) Пример обрабатываемый поверхности; б) области изменения $P_{рез}$ и $V_{п}$, где $P_{рез}$ – сила резания, а $V_{п}$ - скорость перемещения инструмента относительно заготовки

Исходным требованием к поверхности обработки является требование к шероховатости поверхности R_z и к возможным прижогам h_p , что особенно критично при обработке титановых сплавов. Данные критерии определяются в большей степени скоростью относительного перемещения инструмента и обрабатываемой заготовки $V_{п}$. При этом существуют ограничения скорости $V_{п.min}$ и $V_{п.max}$, когда обеспечиваются допустимые $h_{п.min}$ и $h_{п.max}$, $R_z min$ и $R_z max$. Область ограничений по $V_{п}$ связана с ограничениями по силе резания $P_{рез.min}$ и $P_{рез.max}$. Данные ограничения обусловлены упругими допустимыми деформациями, определяемыми данными силами, $E_{упр.min}$ и $E_{упр.max}$. Ограничение на $P_{рез}$ может быть определено также конструктивными допустимыми размерами, при которых наступало разрушение инструмента либо других конструкций станка.

Область $D_{max} \in (P_{рез.min} \div P_{рез.max}, V_{п.min} \div V_{п.max})$ накладывает ограничения на $P_{рез}$ и $V_{п}$ с целью обеспечения качества, точности и прочности конструкции. Однако в области D_{max} существует область, в которой обеспечивается максимальный съём материала, что тождественно мощности, затрачиваемой на резание, и определяется формулой (6.1).

$$Q_{рез} = V_{п} * P_{рез}. \quad (6.1)$$

На рисунке 6.6(а) $Q_{рез}$ представляет поверхность 1, которая описывается уравнением (6.1). Так как $Q_{рез}$ имеет максимум на границе зоны (2), то область D (рис.6.5.(б)) является той областью изменения D ($P_{рез}, V_{п}$), где обеспечивается требуемое качество, точность и максимальная производительность. На рис.6.6(б) приведены области изменения $P_{рез}$ и $V_{п}$ равные соответственно составляющим силы $P_{рез}$ и скорости $V_{п}$ в направлении движения инструмента относительно заготовки и обеспечивающие требуемую шероховатость поверхности, глубину прижогов (h_p) обработанного поверхностного слоя и упругие допустимые деформации $E_{упр}$ от действия силы резания. Нахождение $P_{рез}$ и $V_{п}$ в области D дополнительно к требуемому качеству обеспечивает максимальный съём материала в единицу времени $Q_{рез max}$. Данная область ограничена погрешностью измерения $P_{рез}$ и $V_{п}$. Поэтому управление исполнительными приводами манипуляторов перемещения инструмента и детали осуществляется таким образом, чтобы находиться в области D ($P_{рез}, V_{п}$), так как в этом случае обеспечивается требуемое качество поверхности после обработки и максимальная производительность.

Подобная система адаптации технологического процесса к условиям работы интеллектуального робототехнического оборудования широко используется современной промышленностью и позволяет существенно снизить издержки производства и повысить качество выпускаемой продукции.

7. Управление производственным робототехническим комплексом с использованием элементов интеллектуального управления

7.1 Основные функции и особенности интеллектуальных систем управления

Основными требованиями к управлению технологическими системами, построенными на подвижных стержневых манипуляторах, являются, во-первых, обеспечение технологических режимов для выполняемой операции, точности и качества получения поверхности при ее обработке и, во-вторых, выполнение указанных требований при наличии упругих деформаций исполнительных механизмов [21]. Это возможно только в том случае, если работает система контроля технологических параметров, геометрических размеров обрабатываемой поверхности, положения исполнительных механизмов и упругих передвижений звеньев манипуляторов перемещения инструмента и изделия. Указанные системы контроля оснащаются датчиками контроля положения звеньев механизма, упругих перемещений, режимов обработки (датчики измерения силы резания, подачи, скорости резания, износа инструмента и др.), качества обрабатываемой поверхности и ее геометрических размеров.

Система управления технологической машиной в целом представляет сложную систему, способную решать отдельные интеллектуальные задачи. Система управления предназначена для формирования законов управления исполнительными приводами, обработки информации систем контроля, задания траектории перемещения инструмента относительно обрабатываемой заготовки и обеспечения требуемых режимов обработки. Основными функциями интеллектуальной системы управления являются следующие.

1. Описание поверхности, которую требуется получить после обработки на каждом переходе, а также после окончательной обработки. Эта информация хранится в виде массива опорных точек поверхности.

2. Формирование траектории движения инструмента. Траектория рассчитывается исходя из снимаемого припуска на каждом переходе как непрерывное перемещение подвижного трехгранника в системе координат детали.

3. Сравнение программной траектории перемещения инструмента с реальным его положением в системе координат детали. На основе данного сравнения определяются погрешности линейных и угловых координат.

4. Определение реальных координат заготовки. Оптическая система контроля поверхности определяет реальные координаты поверхности заготовки

в системе координат детали. Сравнивая реальные координаты с идеальными, формируется массив распределения припуска по обрабатываемой поверхности.

5. Вторым функциональным предназначением оптической системы контроля является определение шероховатости обработанной поверхности и ее распределение. В зависимости от дискретной градации уровня шероховатости формируются зоны на поверхности с заданным уровнем микронеровностей.

6. Выбор информационных датчиков контроля положения. Информационные датчики выбираются из суммарного количества датчиков, определяющих перемещения в сочленениях звеньев механизма параллельной структуры. Критерием, по которому выбираются данные датчики, является минимум погрешности вычисления выходного звена при заданной погрешности датчиков.

7.2. Состав системы управления и функциональные характеристики ее элементов

В состав системы управления входят сепаратные приводы, представляющие замкнутые по положению следящие системы по каждой управляемой координате механизма. Кроме этого, система управления в целом также представляет следящую систему, в которой осуществляется сравнение программного положения режущей кромки инструмента с реальным его положением в системе координат детали. Для этих целей применяются специальные математические расчёты, целью которых является нахождение параметров, обеспечивающих устойчивость системы и требуемую точность. Решение прямой и обратной задач кинематики подвижных стержневых механизмов параллельной структуры осуществляется с использованием дополнительных датчиков. Для этого датчики положения устанавливаются в сочленениях звеньев, содержащих и не содержащих исполнительные приводы. Это позволяет оперативно вычислять управление исполнительными приводами и сокращает вычислительные ресурсы. Однако при этом необходимо решать задачу выбора группы датчиков для соответствующей конфигурации механизма, которые с наибольшей точностью определяют положение его выходного звена. Например, два датчика (рис.7.1.), имеющие одинаковую погрешность определения углового положения $\Delta_1 = \Delta_2$ с разной точностью определяют линейные перемещения в направлении оси X. Датчик D_1 определяет значение X более точно, чем D_2 , и $\Delta x_1 < \Delta x_2$. При другом положении точки i на плоскости значимость точности датчика может поменяться.

Манипулятор перемещения изделия специального робота-станка для обработки пера лопаток (рис.7.2) содержит дополнительные датчики. Данный манипулятор имеет четыре управляемых двигателя D_1, D_2, D_3, D_4 для перемещения выходного звена по четырем координатам: двум линейным и двум угловым. Кроме датчиков контроля углов поворота двигателей q_1, q_2, q_3 и q_4 , в механизме установлены датчики измерения углов взаимного положения звеньев, расположенные в сочленениях $\varphi_A, \varphi_B, \varphi_C$.

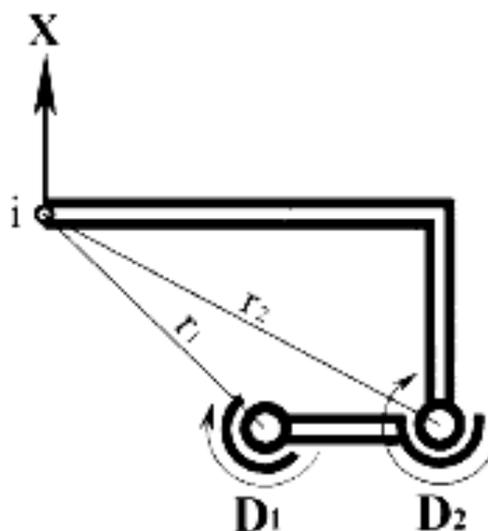


Рис. 7.1. Кинематическая схема установки дополнительных датчиков D_1 и D_2

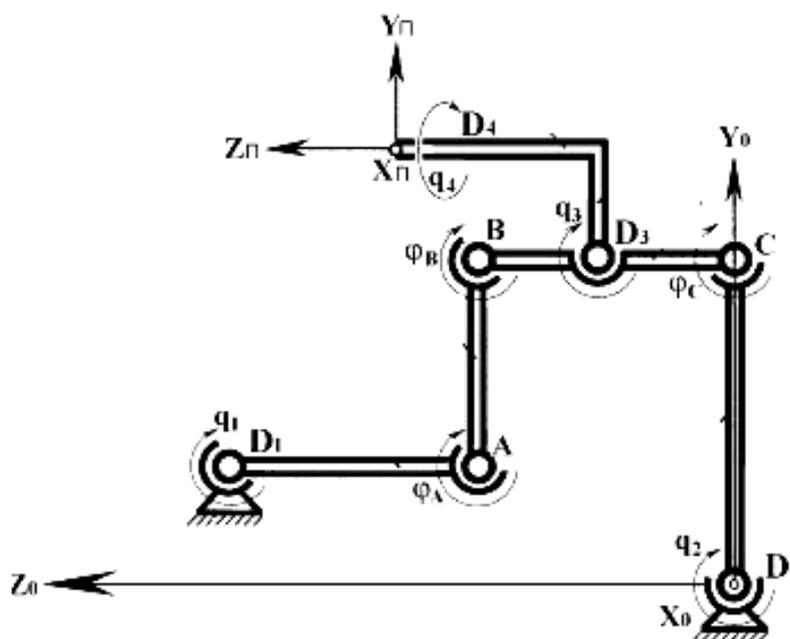


Рис. 7.2. Кинематическая схема расположения дополнительных датчиков робота-станка

Для определения положения платформы робота-станка относительно базовой системы координат $(XYZ)_0$ достаточно знать длины всех звеньев и четыре угла поворота. При наличии семи датчиков контроля углового положения звеньев требуется найти такое сочетание четырех из семи информационных углов, которое обеспечит минимальную погрешность определения координат выходного звена относительно базовой системы координат.

В настоящее время самым распространённым элементом интеллектуальной системы управления является встроенная оптическая система контроля. Встроенная оптическая система контроля поверхности позволяет оперативно изменять траекторию относительного перемещения инструмента и изделия. Система контроля обрабатываемой поверхности дает

информацию о топологии распределения припуска по всей обрабатываемой поверхности и формирует зоны с разной величиной припуска и шероховатости, а это позволяет планировать перемещение инструмента, изменяя закон его движения в зависимости от обрабатываемой зоны. Таким образом, речь идёт об интеллектуальной системе управления, которая осуществляет настройку, выбор режимов и траекторий движения инструмента для отдельных зон поверхности.

7.3. Пример расчёта параметров встроенной оптической системы контроля на примере абразивной обработки поверхности

Следует отметить, что при обработке абразивным инструментом, который характеризуется стабильным съёмом материала в достаточно широком диапазоне изменения прижимающего усилия и подачи, можно получать высокоточные геометрические размеры поверхности изделия при недостаточной точности механизмов станка [4]. Это возможно только при высокоточной системе контроля. В данном случае инструмент снимает слой материала, толщина которого меньше допустимой погрешности на окончательный размер поверхности. Для реализации данных возможностей в системе управления должны быть заложены интеллектуальные алгоритмы выбора законов перемещения инструмента по информации системы контроля. Система контроля обрабатываемой поверхности выдает информацию о величине припуска и шероховатости поверхности. Рассмотрим интеллектуальную задачу, решаемую при ручном способе обработки только человеком. Это выбор траекторий относительного перемещения инструмента и обрабатываемого изделия для различных зон, задаваемых системой контроля поверхности. При этом выделяются зоны на обрабатываемой поверхности с соответствующей величиной припуска или шероховатости (рис.7.3.).

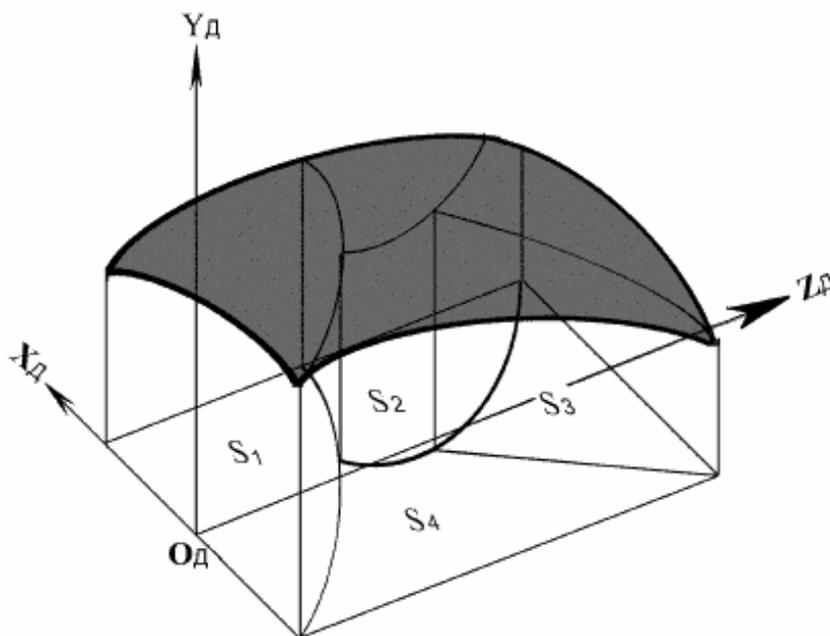


Рис. 7.3. Обрабатываемая поверхность, где S1,S2,S3 и S4 – зоны обработки

Требуется провести анализ каждой зоны и выбрать для нее соответствующую траекторию перемещения инструмента и режимы обработки. Анализ зон включает: определение ее границ, вписывание ее в одну из фигур, хранящихся в базе данных, выбор начальной точки движения инструмента и назначение траектории его движения. Таким образом, требуется решить задачу распознавания образа зоны поверхности и выбрать варианты обработки. Данную процедуру необходимо представить на языке формальной логики. Рассмотрим детерминированный подход на основе обучаемых классификаторов. Будем считать, что оптическая система, просматривая всю поверхность с заданной дискретностью, формирует координаты зоны S_i , в которой отклонения припуска или шероховатости лежат в пределах заданной величины Δ_i . Каждая зона представляет массив точек поверхности, характеризуемых двумя координатами $R_i=(x_i z_i)$. Третья координата может быть вычислена через две известные, так как поверхность считается заданной и следовательно имеется зависимость $y_i=F(x_i z_i)$, представляемая в виде сплайнов либо двумерных полиномов.

С целью сокращения объема информации достаточно для каждой зоны S_i хранить только координаты ее границы, так как координаты и режимы обработки внутренних точек полностью определяются через координаты границы. Зная границы зоны, требуется отнести ее к одной из регулярных фигур, в которую она вписывается полностью с минимальными отклонениями. В качестве регулярных фигур принимаются известные геометрические фигуры, в нашем случае это окружность и прямоугольник. Близость поверхностей будем оценивать по минимуму площадей $S_{\phi}-S_i = \min$, где S_{ϕ} — площадь типовой фигуры, S_i — площадь анализируемой зоны.

Первоначально необходимо оптимальным образом вписать зону в каждую из типовых фигур. Затем сравнить площади фигуры с площадью зоны и причислить ее к фигуре, для которой разность площадей минимальна. Сопоставление зоны поверхности с одной из фигур выполняется по расстоянию между множествами точек границы анализируемой зоны и типовой фигуры. Первоначально определяются координаты центра зоны как центра ее тяжести (рис. 7.4.).

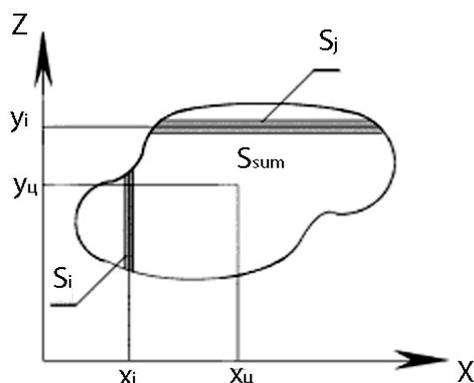


Рис. 7.4. Определение координат центра одной зоны обработки

$$X_u = \frac{\sum_{i=1}^n s_i x_i}{S_{sum}} \quad Y_u = \frac{\sum_{j=1}^n s_j y_j}{S_{sum}} \quad (7.1)$$

Координаты $X_u Y_u$ принимаются за координаты центра описанной окружности. Радиус окружности определяется как расстояние до максимально удаленной точки границы зоны:

$$R = \max_{x_i \in X_u, y_i \in Y_u} = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2} \quad (7.2)$$

Для прямоугольника координаты $X_u Y_u$ также принимаются за центр его тяжести. Требуется вписать анализируемую зону в типовую поверхность с минимальной площадью (рис. 7.5). Для этого осуществляется вращение прямоугольника вокруг точки с координатами $X_u Y_u$ и определяется касание сторон ABCD с границей зоны. Для прямой В точки зоны лежат выше этой прямой и точка касания 2 должна удовлетворять уравнению прямой.

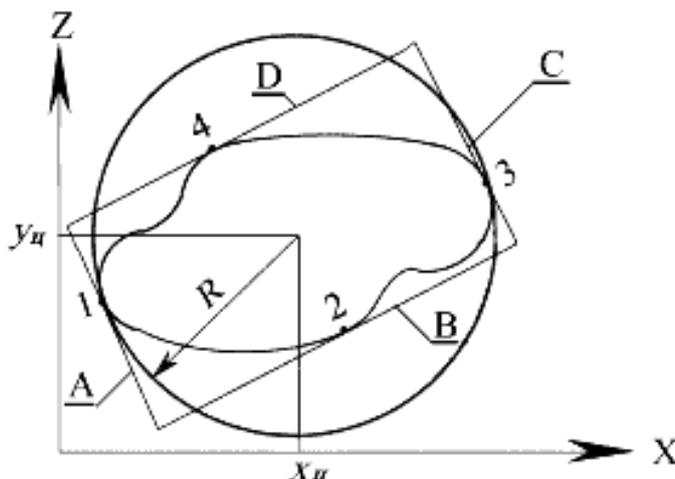


Рис. 7.5. Включение анализируемой зоны в типовую поверхность

Если допустить, что прямая В (рис. 7.5) описывается уравнением

$$y_B = K_i x_B + b_i \quad (7.3)$$

то для поиска точек касания 2 и 4 при фиксированном K_i изменяется $b_i = \text{var}$ и определяются расстояния от точек границы до прямой. В точке касания расстояние до прямой равно 0. Одновременно определяется уравнение перпендикуляра к прямой (7.3) Аналогично определяются точки касания 1 и 3 с данной прямой.

Рассматриваемая процедура повторяется при повороте прямых на угол 180° с заданной дискретностью ($K_i = \text{var}$), и для каждого прямоугольника определяется его площадь. В качестве прямоугольника, в который вписывается анализируемая зона, принимается прямоугольник, имеющий минимальную площадь.

Затем определяется разность между площадью анализируемой зоны и площадями описанного круга и прямоугольника. Фигура, для которой разность площадей наименьшая, принимается как наиболее близкая по своей конфигурации к анализируемой зоне. Для зоны, наиболее близкой к кругу, принимается траектория движения инструмента по спирали (рис. 7.6(a)),

начиная от центра. Для прямоугольника в качестве траектории движения инструмента принимается ломаная линия с началом, например, в точке Н (рис. 7.6 (б)).

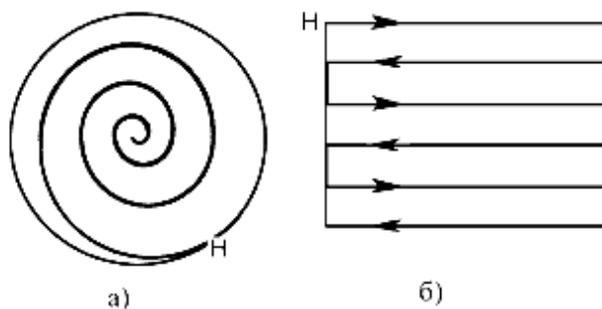


Рис. 7.6. Траектории движения инструмента при обработке поверхности

Также рассмотренная оптическая система контроля поверхности обладает способностью по интенсивности отраженного света от поверхности определять качество поверхности, ее шероховатость.

8. Воспроизведение сложных поверхностей деталей приборов при помощи систем искусственного интеллекта

При создании системы управления процессом механической обработки заготовок с использованием многозвенных манипуляционных механизмов, которые управляются системами с элементами искусственного интеллекта, возникает необходимость экономного описания этого процесса, в том числе описания геометрического контура обрабатываемой поверхности. При этом следует отметить, что метод описания геометрии обрабатываемой поверхности определяет принципы построения систем управления манипуляционными механизмами, а также системы распознавания геометрических параметров. Геометрия обрабатываемой поверхности определяет также требуемое количество степеней подвижности манипулятора, необходимое для выполнения данной операции.

Поверхности реальных промышленных деталей, имеющих сферическую, цилиндрическую, коническую или иную форму, могут быть заданы следующими параметрами: радиусами кривизны, кручения и координатами опорных точек или линий. Перемещение инструмента по траектории на обрабатываемой поверхности от одной опорной точки к другой может быть задано как перемещение по траектории с заданными радиусами кривизны и кручения. Форма поверхности образуется режущей кромкой инструмента, который перемещается по траектории на обрабатываемой заготовке. Эта траектория выбирается таким образом, чтобы при движении по ней режущей кромки инструмента образовывалась требуемая поверхность (Рис. 8.1).

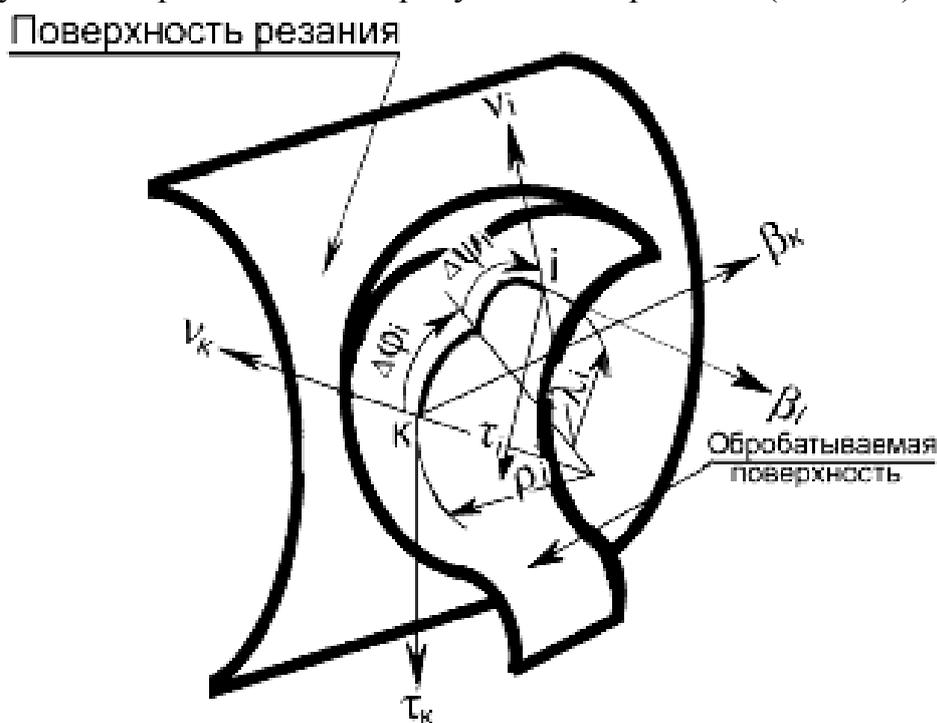


Рис. 8.1. Обрабатываемая поверхность заготовки

При движении инструмента по траектории, расположенной на поверхности обрабатываемой заготовки (рис. 8.1), ось сопровождающего трехгранника (трехгранника Фрэнэ) τ_i направлена по касательной к траектории, ν_i — по нормали, а β_i - по бинормали к поверхности, на которой расположена траектория. Основные параметры, характеризующие траекторию в пространстве, - это текущие линейные координаты траектории, радиус кривизны ρ_i и радиус кручения λ_i . Радиус кривизны ρ_i в i -й точке траектории направлен вдоль оси ν_i и определяется как производная вектора τ_i по дуге s_i :

$$\frac{1}{\rho_i} \nu_i \rightarrow = \frac{\partial \tau_i \rightarrow}{\partial s_i} \quad (8.1)$$

Величина радиуса кривизны ρ_i в каждой i -й точке поверхности вычисляется через модуль производной:

$$\rho_i = \frac{1}{\left| \frac{\partial \tau_i \rightarrow}{\partial s_i} \right|} \quad (8.2)$$

Цилиндрическая поверхность полностью может быть задана координатами опорных точек траектории, радиусами кривизны ρ_i и направляющими косинусами осей ν_i и β_i , причем для цилиндрической поверхности направляющие косинусы оси β_i постоянны для всех опорных точек. Радиус кручения λ_i (рис.8.1) также направлен по оси ν_i и определяется как производная вектора β_i по дуге s_i :

$$\frac{1}{\lambda_i} \nu_i \rightarrow = \frac{\partial \beta_i \rightarrow}{\partial s_i} \quad (8.3)$$

Величина радиуса кручения λ_i в каждой i -й опорной точке поверхности вычисляется через модуль производной:

$$\lambda_i = \frac{1}{\left| \frac{\partial \beta_i \rightarrow}{\partial s_i} \right|} \quad (8.4)$$

При одновременном повороте с заданными радиусами кривизны ρ_i и кручения λ_i получается произвольная форма обработанной поверхности.

Для записи обрабатываемой поверхности воспользуемся методом, описывающим перемещение подвижного трехгранника относительно выбранной системы координат, в которой задается поверхность. Положение трехгранника задается матрицей, которая является универсальной как для описания поверхности, так и для описания кинематической схемы сложного пространственного механизма.

Для задания траектории на поверхности также используется перемещающийся трехгранник, в котором направление движения по траектории совпадает с осью τ_i (Рис. 8.1). В зависимости от способа задания системы координат, относительно которой описывается обрабатываемая поверхность, и, соответственно, способа крепления заготовки, определяются алгоритмы управления манипуляционной системой. Кроме того, способ

закрепления заготовки в захватном устройстве задает количество степеней подвижности манипулятора перемещения заготовки, необходимое для получения требуемой поверхности.

9. Контроль работы производственного комплекса с использованием интеллектуальной системы управления качеством изделия

В настоящее время при обработке поверхностей сложной формы, к точности исполнения которых предъявляются высокие требования, применяются универсальные или специальные измерительные средства, производящие дискретный контроль положения точек или отдельных сечений обрабатываемых поверхностей. В настоящее время существуют два типа измерительных устройств – контактные и оптические. Контактные измерительные приборы представляют собой сложные механические устройства, оснащенные щупом, который движется по поверхности изделия от точки к точке и при этом фиксируется абсолютное отклонение головки щупа от некоторого базового положения (Рис. 9.1.). Иногда вместо щупа применяется набор стержней. Универсальные устройства контроля, построенные на этой основе, требуют создания специальных механических конструкций, хорошей защиты от вибраций. Как правило, данные устройства сложны в эксплуатации, особенно при измерении внутренних размеров или расстояний между отверстиями.

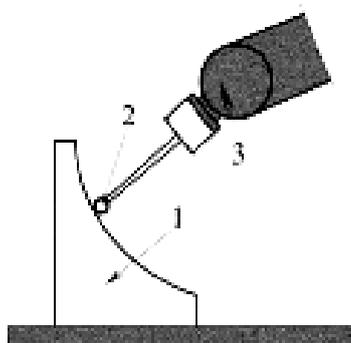


Рис. 9.1 Принципиальная схема контактного измерительного устройства

На рис. 9.1 приведена принципиальная схема наиболее универсального, высокоточного контактного измерительного устройства. В данном устройстве щуп 2 передвигается с помощью электромеханической системы 3 вдоль контролируемой поверхности 1. В щупе размещается сенсорный датчик, который регистрирует относительные отклонения контролируемой поверхности при движении от точки к точке. Система перемещения щупа представляет собой следящую систему, выходные координаты которой определяют геометрические размеры измеряемой поверхности. В силу наличия механических узлов данные системы обладают малым быстродействием.

Оптические системы контроля, в сравнении с электромеханическими, обладают более высоким быстродействием. Данные системы, как правило, реализуют два метода - теневой и контроль в отраженном свете. В первом

случае измеряются размеры тени от предмета, во втором - измеряются амплитудно-фазовые характеристики отраженного света, по которым определяются параметры контролируемой поверхности. Для контроля изделий с точностью, соизмеримой с длиной волны света, прибегают к использованию эффекта интерференции и дифракции света или к применению методов голографии. Однако реализация данных методов требует сложных технических устройств. Поэтому в настоящее время предпочтение все чаще отдается оптическим приборам бесконтактного контроля. Они во многом лишены перечисленных недостатков, просты в реализации, обладают большой универсальностью и быстродействием. Роль оптических методов контроля резко возросла с появлением персональных компьютеров и малогабаритных компьютерных телекамер - TV. Они позволили полностью автоматизировать все необходимые расчеты, исключив при этом подготовительные операции. Применение компьютеров и TV дало возможность с помощью одного и того же прибора измерять различные характеристики поверхности путем выбора соответствующего программного обеспечения.

Метод оптического контроля, который в наибольшей степени подходит для высокоточного бесконтактного измерения геометрических размеров поверхностей деталей, представлен на рис. 9.2. При этом следует отметить, что речь здесь идет только о тех элементах поверхности изделия, до которых доходит световая волна. Принципиальная схема оптического контроля геометрии поверхности (рис. 9.2) включает источник монохроматического света (1), зеркала (2), линзы (3), дифракционная решетка (4), корректирующие зеркала (5), деталь (6), телевизионная камера (7), плата сопряжения (8), компьютер (9). Устройство закрыто от стороннего света и пыли и не чувствительно к вибрации. Свободный доступ обеспечен только к месту установки детали. Компьютер по результатам обработки поверхности выдает в виде таблиц на экран и на печать отклонения во всех сечениях поверхности. Система оптического контроля имеет в общем случае следующие характеристики: время контроля и расчета одной детали - порядка 20 с (данный параметр в основном определяется быстродействием компьютера); точность контроля - в среднем 6 мкм; минимальный радиус контролируемой поверхности - около 50 мкм.

Данные, полученные от измерительной системы, позволяют корректировать траекторию перемещения инструмента относительно заготовки. Телекамера фиксирует изображение в виде массива чисел, элементы которого характеризуют интенсивность светового потока, попадающего на каждый элемент фотоматрицы. Телекамера соединена с компьютером, в котором осуществляется обработка полученной информации. Масштабирование изображений, фиксируемых на телекамере, выполняется с использованием эталонной фигуры, в качестве которой может быть принята, например, высокоточная фасонная поверхность на захватном устройстве робота. Изображение эталонной фигуры фиксируется в условных единицах.

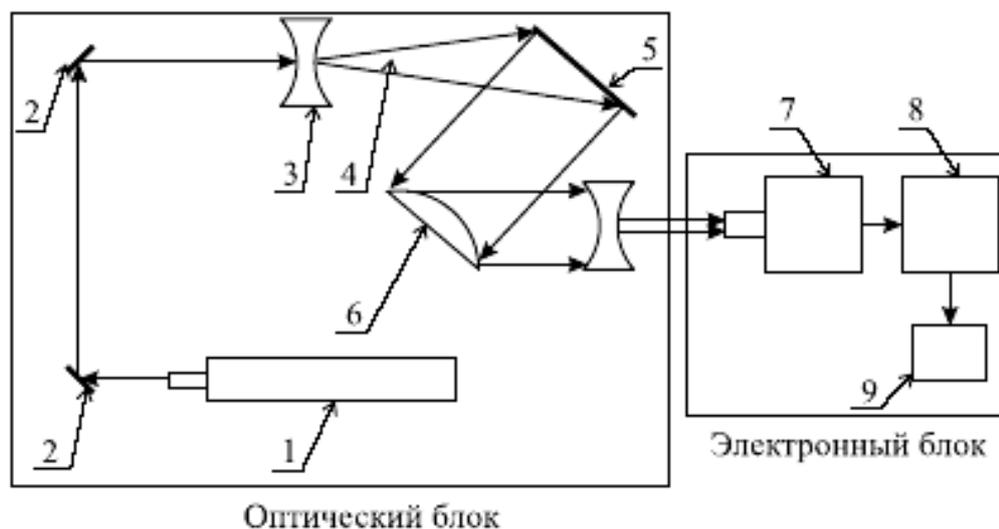


Рис. 9.2 Оптическая система контроля

Зная размеры эталонной фигуры в миллиметрах, можно определить масштаб для базы крепления. При работе с объектами выпуклой формы необходимо учитывать свойство трехмерной аберрации объектива. Изображение в телекамере фиксируется за $0,01 \div 0,001$ сек. Это сводит к минимуму влияние механических вибраций и позволяет получить нужное быстродействие. Кроме телекамеры, система контроля (рис. 9.2) включает:

1. источник освещения, который крепится на жесткой базе таким образом, чтобы свет от него попадал на контролируемый объект;
2. измеряемое изделие, которое может крепиться в захватном устройстве; в этом случае эталонная фигура также размещается на захватном устройстве.

10. Использование встраиваемых технологических систем контроля работы интеллектуальных производственных систем

В предыдущих главах уже упоминалось о возможности расширения функциональных характеристик и интеллектуальных свойств технологической системы путем установки дополнительных датчиков и приводов. Дополнительные встраиваемые элементы контроля в механизмах параллельной структуры существенно упрощают вычисление положения выходного звена, а также дают возможность повысить точность определения его положения [16]. Рассмотрим данную задачу на примере механизма, представляющего манипулятор перемещения изделия специального робота-станка для обработки сложных изделий (рис.10.1).

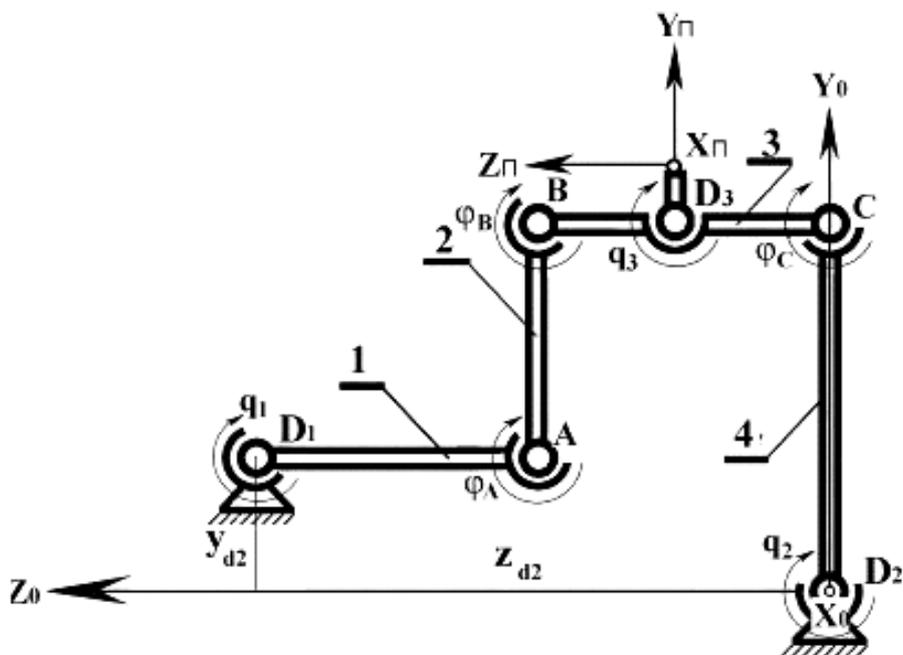


Рис. 10.1 Кинематическая схема манипулятора перемещения робота-станка, состоящего из 4-х звеньев (1, 2, 3 и 4) и 3-х двигателей D_1 , D_2 и D_3

Манипулятор имеет три управляемых двигателя D_1 , D_2 , D_3 для перемещения выходного звена по трем координатам: двум линейным и одному углу поворота выходного звена в плоскости Y_0OZ_0 . Кроме датчиков контроля углов поворота двигателей q_1, q_2 и q_3 , в механизме установлены датчики измерения углов взаимного положения звеньев φ_A , φ_B и φ_C (Рис.10.1), расположенные в сочленениях. Для определения положения выходного звена $(XYZ)_n$ относительно базовой системы координат $(XYZ)_0$ достаточно знать длины звеньев L_1 , L_4 , y_{d2} , z_{d2} и три угла поворота (Рис. 10.1). При наличии шести датчиков контроля углового положения звеньев q_1 , q_2 , q_3 , φ_A , φ_B и φ_C

требуется найти такое сочетание трех из шести информационных датчиков углового положения, которое обеспечит минимальную погрешность определения координат выходного звена механизма относительно координат $(XYZ)_0$. Следует отметить, что при рассмотрении только линейных перемещений выходного звена достаточно из пяти датчиков $q_1, q_2, \varphi_A, \varphi_B$ и φ_C выбрать только два.

Рассмотрим решение данной задачи в общем случае. Связь между вектором, определяющим погрешности выходного звена механизма, - ΔR (в общем случае размерности 6×1) и погрешностями информационных датчиков q_j (размерности $n \times 1$, где $n > 6$) однозначно выражается линейной зависимостью:

$$\Delta R = K_j * \Delta q_j; \quad (10.1),$$

где K_j - матрица весовых множителей $6 \times n$.

Вектор ΔR включает погрешность линейных координат $\Delta x, \Delta y$ и Δz в точке i конечного звена механизма и погрешность угловых координат $\Delta \varphi, \Delta \Phi$ и $\Delta \theta$ вокруг данных осей:

$$\Delta R = [\Delta r_i]^T = [\Delta x \Delta y \Delta z \Delta \varphi \Delta \Phi \Delta \theta]^T \quad (10.2)$$

Погрешности определяются в неподвижной или связанной с деталью системах координат. Общее возможное количество датчиков, с помощью которых можно определить положение выходного звена механизма параллельной структуры, для механизма, имеющего m степеней свободы, равно $n = m^2$. Максимальное число возможных сочетаний информационных датчиков, которое может быть взято для определения положения выходного звена, вычисляется соотношением:

$$C_n^m = (n(n-1)(n-2)...(n-m+1))/(1 \cdot 2 \cdot 3 \cdot \dots \cdot m) \quad (10.3)$$

Чаще всего требуется обеспечить высокую точность вычисления линейных координат. В этом случае число возможных вариантов определения трех линейных координат составляет $C = 7140$ и одной координаты $C = 36$.

Для механизма, представленного на рисунке 10.1, выходным звеном плоского пятизвенника D_1, A, B, C, D_2 является звено L_3 , на котором располагается привод q_3 . Две линейные координаты, определяющие положение центра привода q_3 , могут быть определены датчиками углов $q_1, q_2, \varphi_A, \varphi_B$ и φ_C различным их сочетанием по два: $C = 10$. Поэтому число способов вычисления координат выходного звена в этом случае также равно 10.

Вычисление и анализ элементов матрицы K_j является основной задачей при выборе информационных датчиков, определяющих координаты выходного звена механизма. Элементы данной матрицы являются весовыми множителями при погрешностях датчиков и определяют вклад погрешности каждого датчика в погрешность соответствующей координаты выходного звена механизма.

11. Моделирование технологического комплекса позиционно-силового управления в системе Simulink

Установка дополнительных приводов в совокупности с дополнительными датчиками контроля положения позволяет системе управления «низшего» уровня лучше приспособляться к изменению силового воздействия и более точно управлять перемещением [4]. Расширяется зона обслуживания робота за счет возможности проходить особые положения механизма, в которых система становится неуправляемой. Безусловно, эти функциональные дополнительные возможности могут быть реализованы различными способами. В настоящее время наибольшее распространение получили робототехнические технологические машины, построенные на основе механизмов параллельной структуры, что вносит особую специфику в схему построения данных систем. С точки зрения механики движения невозможно одновременно по одной координате независимо управлять усилием, перемещением и скоростью данного перемещения. Однако механизмы параллельной структуры позволяют частично разделять данные функции и независимо управлять перемещением и усилием.

Для расширения кинематических и динамических возможностей технологического оборудования, построенного на механизмах параллельной структуры, устанавливают приводные элементы на основания механизма и в сочленениях, выполняющих функции только передачи движения от основания к выходному звену. В этом случае дополнительные приводы играют роль устройств силовой разгрузки основных приводов, обеспечивающих заданный закон перемещения. Дополнительные приводы позволяют разгружать основные приводы от статических нагрузок и реакций связей в сочленениях, что в свою очередь повышает быстродействие при движении выходного звена по заданной траектории и, как следствие, повышает точность воспроизведения программных траекторий.

Работа системы позиционно-силового управления с реальным объектом управления описывается сложным математическим аппаратом и представляет некоторую сложность для вычислений обычными способами. Для таких расчётов целесообразно использовать специальные программные средства, например, пакет Simulink среды Matlab. Simulink сочетает в себе наглядность аналоговых машин и точность цифровых вычислительных машин. Simulink обеспечивает пользователю доступ ко всем возможностям пакета MatLab, в том числе к большой библиотеке численных методов. При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым, пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты.

Simulink обеспечивают интерактивную среду для моделирования, при

этом поведение модели и результаты ее функционирования отображаются в процессе работы. Существует возможность изменять параметры модели даже в тот момент, когда она выполняется. Simulink позволяет создавать собственные блоки и библиотеки блоков с доступом из программ на MatLab, Fortran или C, связывать блоки с разработанными ранее программами на Fortran и C, содержащими уже проверенные модели.

На рис. 11.1 представлен пример моделирования системы позиционно – силового управления роботом – манипулятором в системе Simulink.

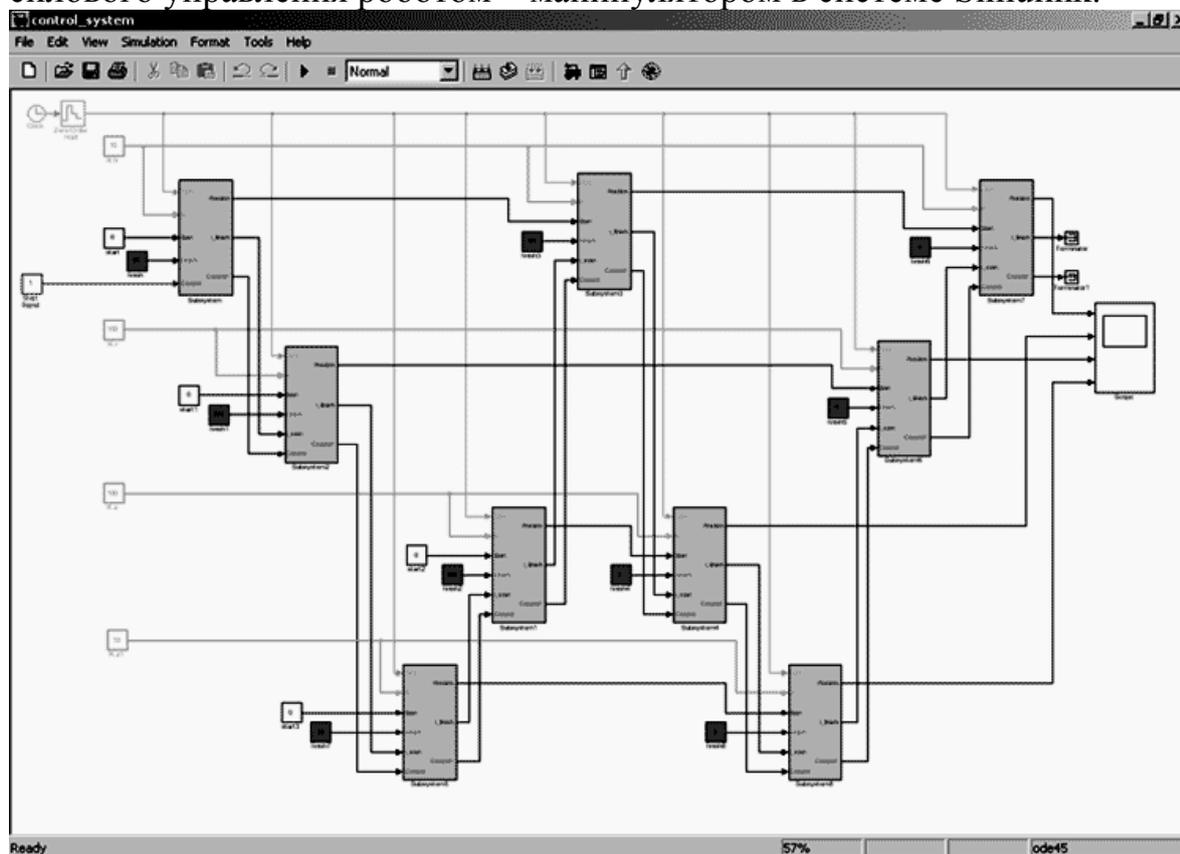


Рис. 11.1 Пример моделирования системы позиционно – силового управления роботом – манипулятором в системе Simulink

Результаты моделирования имеют вид временной циклограммы работы, при этом по оси ординат откладывается значение координаты перемещения вдоль той или иной степени подвижности (рис. 11.2).

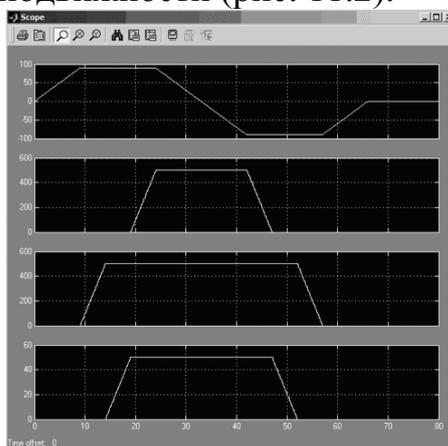


Рис. 11.2 Результат моделирования системы управления

Результат моделирования может быть представлен также в виде анимационной картинки цикла работы манипулятора, фрагмент которой показан на рис. 11.3.

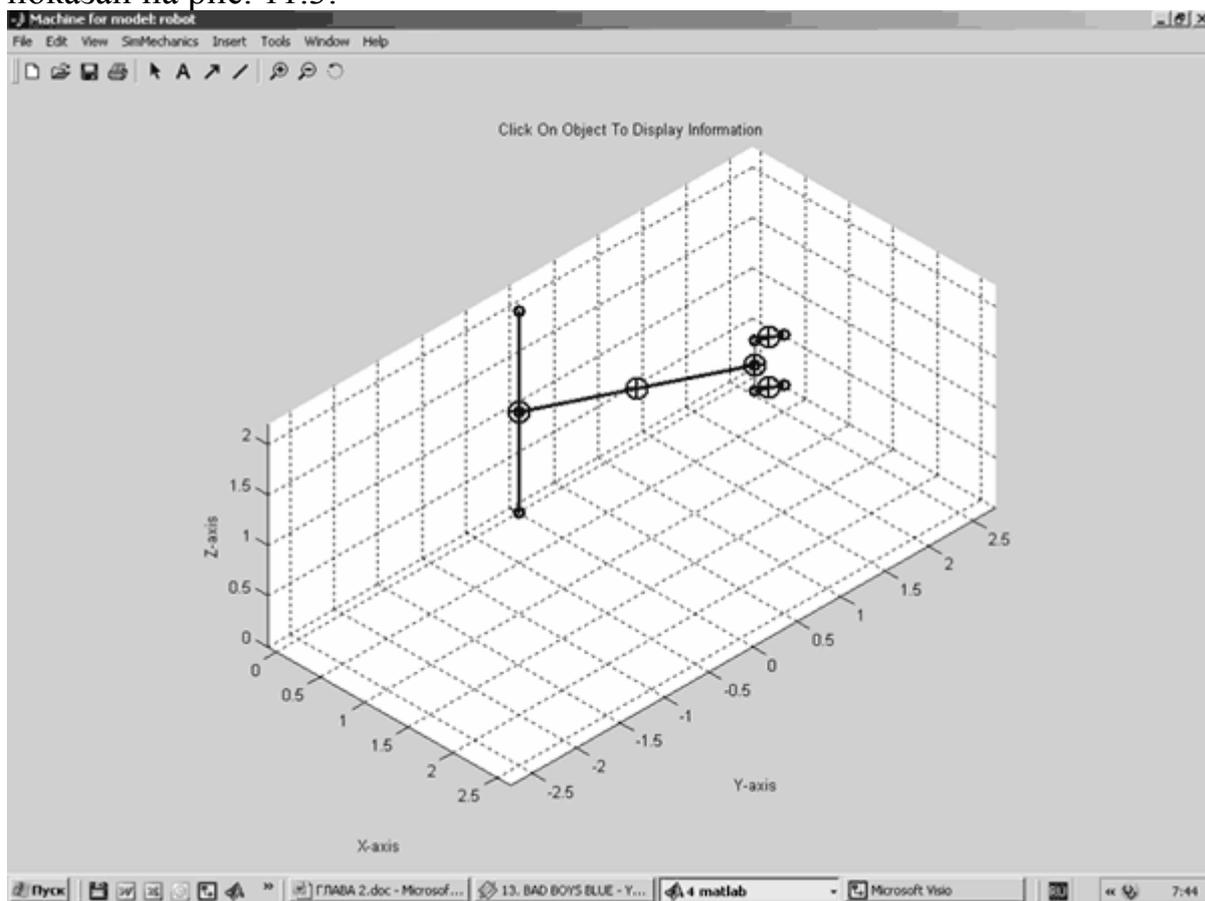


Рис. 11.3 Фрагмент анимационной картины цикла работы системы позиционно – силового управления роботом

Таким образом, среда Simulink позволяет не только максимально упростить расчёты элементов системы позиционно-силового управления роботом, но и визуализировать весь процесс работы устройства.

СПИСОК ЛИТЕРАТУРЫ

1. Аверкин А. Н., Батыршин И. З., Блишун А. Ф., Силов В. Б., Тарасов В. Б. Нечеткие множества в моделях управления и искусственного интеллекта М.: Наука, 1986.
2. Афонин В. Л., Морозов А. В. Управление технологическими роботами для механической обработки М.: ИМАШ РАН, 1995, 156 с.
4. Афонин В. Л., Смоленцев А. Н. Позиционно-силовое управления в механизмах параллельной структуры Справочник, Инженерный журнал, №5, 2003, с. 14-19.
6. Бурдаков С.Ф., Дьяченко В.А., Тимофеев А.Н. Проектирование манипуляторов промышленных роботов и робототизированных комплексов М.: Высшая школа, 1986, 264с.
7. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем СПб: Питер, 2001. 384 с.
8. Девятков В.В. Системы искусственного интеллекта: Учеб. пособие для вузов.- М.: Изд-во МГТУ им. Баумана, 2001, 352с.
11. Люгер Д. Искусственный интеллект М.: Мир, 2003. 690 с.
12. Люгер Д. Искусственный интеллект. Стратегии и методы решения сложных проблем М.: Издательство Вильямс, 2003. 864 с.
12. Макаров И. М., Топчиев Ю. И. Робототехника. История и перспективы М.: Наука, МАИ, 2003. 350 с.
13. Мачульский И.И., Запятой В.П., Майров Ю.П. Робототехнические системы и комплексы. М.: Транспорт, 1999, 446с.
14. Попов Э. В., Фоминых И. Б., Кисель Е. Б., Шапт М. Д. Статические и динамические экспертные системы М.: Финансы и статистика, 1996, 320 с.
16. Тимофеев А.В. Адаптивные робототехнические комплексы Л.: Машиностроение, 1988, 322с.
17. Уотерман Д. Руководство по экспертным системам М., Мир, 1989, 388 с.
19. Шахинпур М. Курс робототехники Пер. с англ. М.: Мир, 1990, 527с.
20. Ямпольский Л.С., Яхимович В.А., Вайсман Е.Г. Промышленная робототехника К.: Техника, 1984, 264с.
21. Ямпольский Л.С., Банашак З., Хасегава К. и др. Управление дискретными процессами в ГПСЛ К.: Техника, 1992, 251с.

В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

КАФЕДРА ТЕХНОЛОГИИ ПРИБОРОСТРОЕНИЯ

Кафедра технологии приборостроения относится к числу ведущих кафедр института со дня его основания в 1931 году. Тогда она называлась кафедрой механической технологии и возглавлялась известным ученым в области разработки инструмента профессором А.П. Знаменским. Позже она была переименована в кафедру технологии приборостроения.

За время своего существования кафедра выпустила из стен института более тысячи квалифицированных инженеров, более сотни кандидатов и докторов наук. В разные годы ее возглавляли известные ученые и педагоги профессора Николай Павлович Соболев и Сергей Петрович Митрофанов.

Кафедра имеет выдающиеся научные достижения. Заслуженными деятелями науки и техники РСФСР, профессором СП. Митрофановым были разработаны научные основы группового производства, за что он был удостоен Ленинской премии СССР. Методы группового производства с успехом применяются в промышленности и постоянно развиваются его учениками. Заслуженным деятелем науки и техники РСФСР, Заслуженным изобретателем СССР Юрием Григорьевичем Шнейдером разработаны метод и инструментальный нанесения регулярного микрорельефа на функциональной поверхности.

Основными научными направлениями кафедры являются: научные основы организации группового производства (руководители: Д.Д. Куликов, Е.С. Падун); автоматизация технологической подготовки производства (руководители: Д.Д. Куликов, Б.С. Падун, Е.И. Яблочников); регуляризация микрорельефа поверхностей деталей машин и приборов (руководитель Ю.П. Кузьмин); управление функциональными свойствами поверхностного слоя деталей (руководитель В.А. Валетов). Последнее направление организовано в 1988 году и в настоящее время детально разработаны не только теоретические

основы управления характеристиками поверхностного слоя деталей, но и методики проведения экспериментальных исследований по определению влияния различных факторов на характеристики поверхностного слоя.

Разработаны современные установки - измерительно-вычислительные комплексы для анализа микрогеометрии поверхностей и технологических остаточных напряжений в поверхностном слое. По этому направлению за последние годы защищено три кандидатских диссертации и подготовлена одна докторская. В настоящее время на кафедре в рамках этого направления работают не только преподаватели, но и молодые аспиранты.

Кафедра имеет тесные научные и учебные связи с университетами Германии, Франции, Китая. Наиболее способные студенты и аспиранты проходят стажировку по интересующим их проблемам в Техническом университете г. Ильменау (Германия).

Кафедра технологии приборостроения была создана как ведущая и обслуживала чтением лекций по курсу "Технология приборостроения" все выпускающие кафедры ВУЗа. На заведование кафедрой был приглашен профессор Знаменский А.П. - главный инженер завода ГОМЗ, автор первого "Справочника Металлиста", известного во многих странах мира. Преподавателями и сотрудниками кафедры были приглашены высококвалифицированные специалисты, в основном работники промышленности - проф. Соколов - главный инженер инструментального завода им. Васкова, Буталов В.И., Бельфир, Казак и др. Нужно отметить, что до 1961 года кафедра была слабо оснащена оборудованием и поэтому большое время уделялось прохождению практики непосредственно на заводах "ГОМЗ", "Красногвардеец", "Пишмаш" и др., где давались хорошая подготовка, как по работе на станках, так и по разработке грамотных технологических процессов. В 1951г. в кафедру влились кафедры станков и теории резания. По своей оснащённости она не удовлетворяла качества подготовки инженеров-технологов. На кафедре ученое звание профессора имел только Н.П.Соболев - заведующий кафедрой.

В настоящее время на кафедре работают 7 профессоров и 9 доцентов.

С 1998 года кафедру возглавлял проф. Н.Д. Фролов. В настоящий момент кафедру возглавляет Е.И. Яблочников.

**Вячеслав Алексеевич Валетов
Сергей Дмитриевич Третьяков
Анна Алексеевна Орлова**

**ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ ИЗГОТОВЛЕНИЯ
ПРИБОРОВ И СИСТЕМ**

Учебное пособие

В авторской редакции

Зав.редакционно-издательским отделом

Н.Ф. Гусарова

Заказ № 1141

Объем 134 стр. Тираж 100 экз. Подписано к печати

Редакционно-издательский отдел

Санкт-Петербургского государственного университета информационных
технологий механики и оптики

Лицензия ИД №00408 от 05.11.99