

## Принципы построения и иерархическая структура адаптивных робототехнических систем

Адаптивные робототехнические системы - сложные системы, способные выполнить поставленную перед ними задачу в недетерминированной внешней среде. Это означает, что в процессе выполнения задания возникают ситуации, которые не могут быть точно описаны на этапе формирования задания, но ориентировочно они могли быть предусмотрены. Этим адаптивные робототехнические системы отличаются от систем с искусственным интеллектом. Определяющей характеристикой адаптивных систем является их способность получать информацию о внешней среде и использовать ее далее для формирования управления.

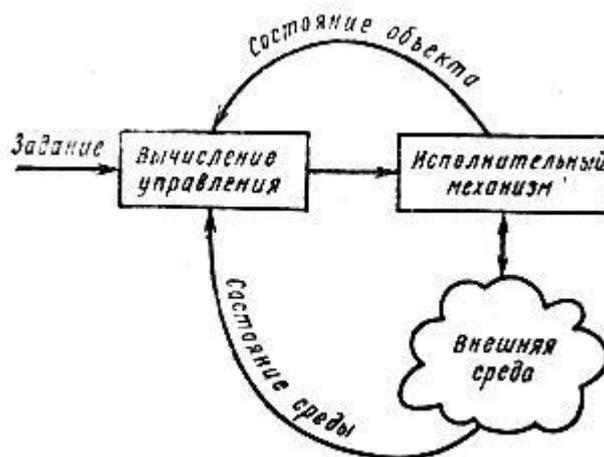


Рис. 3.6. Общая схема адаптивной робототехнической системы

На рис. 3.6 показана общая схема робототехнической адаптивной системы. На ее вход поступает каким-либо образом сформулированное задание, для исполнения которого вычисляется управление на основании информации о внешней среде и текущем состоянии механизма. Сформированное управление изменяет состояние механизма, который, возможно, в свою очередь изменяет состояние внешней среды. Эти изменения состояний механизма и внешней среды учитываются при формировании управления таким образом, чтобы обеспечить выполнение задания исполнительным механизмом. Параметры внешней среды, которые необходимы роботу в связи с выполнением им задания, чаще всего поступают с информационной подсистемы, однако поскольку внешняя среда также может оказывать влияние на работа, то информацию о ней можно получить косвенно с помощью датчиков, поставляющих информацию о состоянии механизма. Поэтому можно строить системы с элементами адаптации без развитых сенсорных подсистем.

**Пример простой адаптивной робототехнической системы.** Для выяснения общих принципов построения адаптивных систем рассмотрим простую адаптивную робототехническую систему (рис. 3.7). Пусть на конвейере, движущемся с постоянной скоростью  $v$ , перемещаются блоки, расположение которых заранее известно (например, они находятся в углублениях ленты конвейера). Задача манипуляционного робота - взять блок с ленты конвейера и поставить его в заранее известное место на неподвижной поверхности, которое может служить питателем для дальнейших операций технологического процесса или использоваться другим манипулятором как некоторое промежуточное положение. В этом случае, если скорость  $v$  конвейера известна, это задание может быть выполнено с помощью простого программного робота. Пусть исполнительный механизм представляет собой антропоморфный манипулятор с 6-ю степенями свободы. Для данной задачи эта кинематическая схема является избыточной, т. е. для ее решения достаточно трех степеней свободы. Но одной из характерных особенностей робота является его универсальность, т. е. способность выполнять широкий круг задач, и тогда могут потребоваться более сложные движения. При сделанных предположениях относительно кинематики манипулятора его состояние можно описать выражением

$$S_q = (q_1, q_2, \dots, q_6), \quad (3.1)$$

где  $q_i$  - углы поворота или перемещения подвижных сочленений.

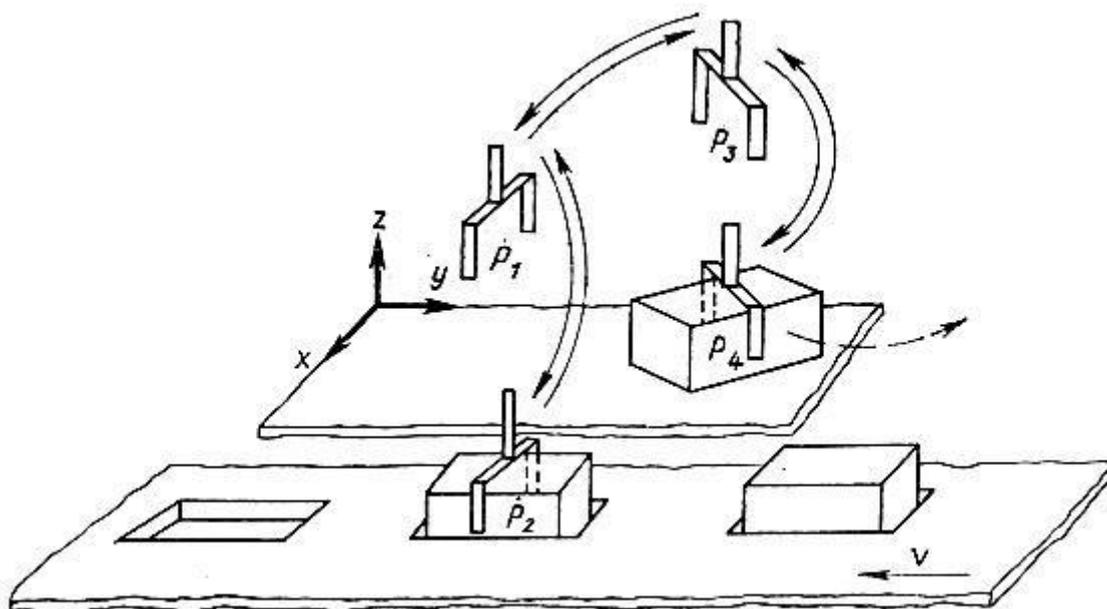


Рис. 3.7. Простая адаптивная робототехническая система:  $P_1$ - $P_4$  - точки позиционирования

Описание каждой из точек позиционирования может содержать кроме  $S_q$  еще и набор параметров  $\gamma$ , определяющих дополнительные характеристики

точки, не связанные с ее пространственными характеристиками: например, способ перемещения к точке, состояние захвата, время пребывания в точке и т. д. Тогда полное описание 1-й точки позиционирования представляет вектор

$$P_q^i = (S_q^i, r^i), \quad (3.2)$$

где  $S_q^i$  определяется формулой (3.1):

$$r^i = (r_1^i, r_2^i, \dots, r_l^i). \quad (3.3)$$

Для программного робота, выполняющего сформулированную выше транспортную задачу, достаточно каким-либо образом определить координаты точек  $P_1—P_4$  (чаще всего это делается на этапе обучения робота) и далее заставить его выполнять соответствующую последовательность движений. В целом вся эта процедура может быть описана с помощью программы, приведенной на рис. 3.8. На самом деле для промышленного неадаптивного робота нецелесообразно представлять программу движения на некотором проблемно-ориентированном языке: это потребовало бы не только более развитой аппаратной части комплекса (например, управляющую ЦВМ, устройства ввода-вывода), но и развитого математического обеспечения (в частности, транслятора с языка программирования), в то время как можно обойтись гораздо более простыми и, следовательно, дешевыми средствами. Поэтому такую программу надо рассматривать лишь как алгоритм, в соответствии с которым происходит функционирование программного робота.

```

1:      P1=Q11, Q12, Q13, Q14, Q15, Q16, R11, P12, R13;
2:      P2=Q21, Q22, Q23, Q24, Q25, Q26, R21, R22, R23;
3:      P3=Q31, Q32, Q33, Q34, Q35, Q36, R31, R32, R33;
4:      P4=Q41, Q42, Q43, Q44, Q45, Q46, R41, R42, R43;
5:      ПЕРЕЙТИ_В_ТОЧКУ P1;
6:      ПЕРЕЙТИ_В_ТОЧКУ P2;
7:      ЗАКРЫТЬ_СХВАТ;
8:      ПЕРЕЙТИ_В_ТОЧКУ P1;
9:      ПЕРЕЙТИ_В_ТОЧКУ P3;
10:     ПЕРЕЙТИ_В_ТОЧКУ P4;
11:     ОТКРЫТЬ_СХВАТ;
12:     ПЕРЕЙТИ_В_ТОЧКУ P3;
13:     ПОВТОРИТЬ_С_МЕТКИ 5;

```

Рис. 3.8. Программа для робота с программным управлением

Далее даже небольшое отклонение положения блока в момент захватывания от ожидаемого, вызванное, например, неравномерностью

скорости конвейера, приведет к тому, что блок будет захвачен манипулятором не точно в центре тяжести или не будет захвачен вовсе. При этом робот тем не менее будет продолжать выполнять запрограммированную последовательность движений до тех пор, пока его не остановит человек-оператор. Таким образом, применение робота с программным управлением предъявляет ряд требований к технологическому процессу, выражающихся в его детерминированности, т. е. в постоянстве описывающих его параметров.

Заканчивая рассмотрение специфических особенностей функционирования программного робота, рассмотрим схему (рис. 3.9), в соответствии с которой строится его алгоритм управления. На вход системы поступают координаты точек позиционирования, которые являются входными данными для вычисления управляющих сигналов на приводы подвижных сочленений. Модуль, вычисляющий управление, осуществляет преобразование координат точек позиционирования в сигналы управления.

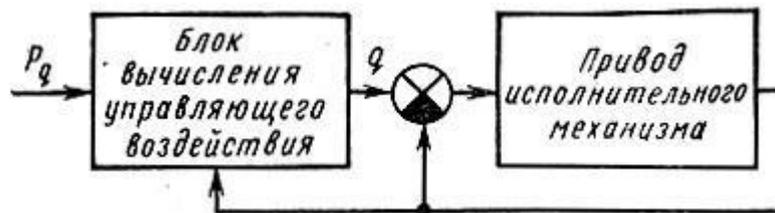


Рис. 3.9. Схема управления программным роботом:  $P_q$  - координаты точки позиционирования;  $q$  - управление

Пусть теперь скорость конвейера непостоянна, в этом случае для успешного выполнения роботом сформулированного выше задания его необходимо оснастить системой, измеряющей скорость конвейера, и, кроме того, изменить алгоритм управления. Главное отличие от программного робота - невозможность заранее вычислить координаты точки  $P_2$ : они определяются в процессе исполнения роботом задания. Таким образом, алгоритм управления роботом, адаптирующимся к скорости движения конвейера, позволяет вычислить положение блока на основании показаний датчика скорости движения, который нами рассматривается как система, поставляющая информацию о внешней среде. Пусть по аналогии с выражением (3.1)  $S_a$  представляет собой вектор, описывающий координаты захвата в декартовой системе координат:

$$S_a = (x, y, z, \alpha, \beta, \gamma), \quad (3.4)$$

а  $P_a^i$  - вектор, описывающий  $i$ -ю точку позиционирования:

$$P_a^i = (S_a^i, r^i). \quad (3.5)$$

Тогда программу, описывающую последовательность действий адаптивного робота, можно представить в виде, показанном на рис. 3.10. Так же как и в программе для программного робота, операторы 1-3 определяют по сути дела режим обучения. Отличие в этом случае заключается в том, что вычисление параметров точки позиционирования  $P_2$  происходит не на этапе обучения, а на этапе исполнения (оператор 5). Эти координаты вычисляются на основании определения скорости конвейера  $v$ ; далее полученные координаты преобразуются в вектор  $P_q$ , т. е. в форму, более удобную для использования в целях управления (рис. 3.11). Остальная часть схемы полностью совпадает со схемой функционирования программного робота, приведенной на рис. 3.8.

```

1:      P1=Q11, Q12, Q13, Q14, Q15, Q16;
2:      P3=Q31, Q32, Q33, Q34, Q35, Q36;
3:      P4=Q41, Q42, Q43, Q44, Q45, Q46;
4:      ПЕРЕЙТИ_В_ТОЧКУ P1;
5:      ВЫЧИСЛИТЬ P2;
6:      ПЕРЕЙТИ_В_ТОЧКУ P2;
7:      ЗАКРЫТЬ_СХВАТ;
8:      ПЕРЕЙТИ_В_ТОЧКУ P1;
9:      ПЕРЕЙТИ_В_ТОЧКУ P3;
10:     ПЕРЕЙТИ_В_ТОЧКУ P4;
11:     ОТКРЫТЬ_СХВАТ;
12:     ПЕРЕЙТИ_В_ТОЧКУ P3;
13:     ПОВТОРИТЬ_С_МЕТКИ 5;

```

*Рис. 3.10. Программа для адаптивного робота*

Таким образом, даже такое несложное устройство, как датчик скорости, позволило реализовать простой адаптивный робот. Используя эту информационную систему, мы можем теперь не требовать постоянства скорости конвейера, продолжая предъявлять тем не менее довольно жесткие требования к расположению блоков на конвейере. Чтобы освободиться и от этого ограничения, используют другую информационную систему - систему технического зрения. Пусть над конвейером расположена телевизионная камера (на самом деле для нас, конечно, не существенно, какой сенсор мы используем: это может быть фотоматрица или другое аналогичное устройство), позволяющая получать зрительную информацию о рабочем пространстве. Тогда, обрабатывая эту информацию программно с помощью ЦВМ или аппаратно, можно определить центр тяжести и ориентацию блока, попавшего в поле зрения камеры, доопределяя, таким образом, недостающие данные для исполнения задания (координаты точки позиционирования  $P_2$ ). Заметим, что программа движения для адаптивного робота, снабженного системой технического зрения, будет в точности совпадать с программой, приведенной на рис. 3.10, хотя, конечно, алгоритмы, положенные в основу работы модулей, обрабатывающих сенсорную информацию, будут

существенно различаться. На самом деле использование такой информационно-емкой сенсорной подсистемы, как система технического зрения, позволяет решать гораздо более сложные задачи: в частности, можно было бы допустить наличие на конвейере разнородных деталей, и тогда адаптивная робототехническая система могла бы решать задачу сортировки, собирая вместе детали одного вида.

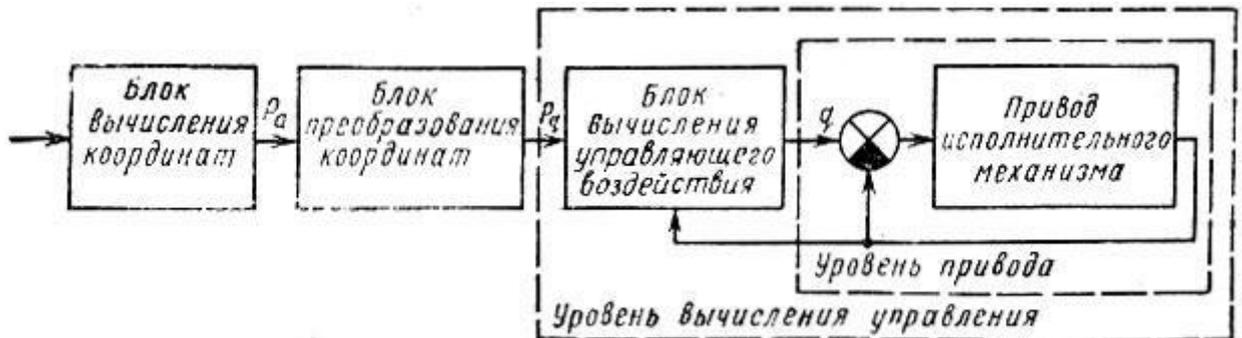


Рис. 3.11. Схема управления адаптивным роботом:  $P_a$  - координаты точки позиционирования;  $v$  - скорость конвейера

Рассмотренный выше простой пример позволил проиллюстрировать основные особенности построения адаптивных систем - наличие сенсорной подсистемы и использование поступающей с нее информации о внешней среде для формирования управления исполнительным механизмом. Оставаясь в рамках того же примера, обратим теперь внимание на то, что приведенные на рис. 3.9 и 3.11 схемы систем управления программного и адаптивного робота частично совпадают. Этот факт не является случайным, а отражает иерархическую структуру адаптивного робота.

**Иерархическая организация адаптивных систем.** Иерархия - это такой многоуровневый набор взаимодействующих подсистем (называемых уровнями иерархии системы), каждая из которых ответственна за решение строго определенной для подсистемы части задачи и имеет доступ к ограниченной информации, необходимой для решения только этой задачи. Каждый уровень иерархия находится в подчиненном положении по отношению к вышестоящему уровню, принимая и исполняя поступающие от него команды и данные, являясь в то же время источником команд и данных для стоящего ниже уровня. Для верхнего уровня иерархии источником команд, как правило, является человек-оператор, формирующий цели на языке, понятном для данного уровня. Это может быть нажатие соответствующей кнопки на пульте, ввод команды на некотором проблемно-ориентированном языке и т. д.

На примере простейших робототехнических систем (программных промышленных роботов) можно утверждать, что в основе их организации лежат элементы иерархии. Действительно, в качестве нижнего уровня

иерархии можно выделить уровень привода, который отвечает за адекватную обработку управляющих сигналов, поступающих к нему от вышестоящего уровня. Уровень привода является самостоятельной законченной подсистемой, которая гарантирует вышестоящему уровню исполнение генерируемых им команд. Следующий уровень иерархии занят вычислением управляющих сигналов на основе данных, которые были подготовлены ранее (на этапе обучения), и команд, вводимых человеком-оператором (как говорилось выше, это может быть нажатие кнопки "Исполнение" на пульте оператора). Данный уровень занят выборкой описателей требуемых точек позиционирования, если необходимо - их анализом и преобразованием, построением интерполирующего многочлена и, наконец, передачей для исполнения нижестоящему уровню привода. На рис. 3.11 выделены эти уровни иерархии, а также показано, что является входом и выходом соответствующего уровня. Реализация такой простейшей двухуровневой структуры (уровни привода и вычисления управления; последний еще называют тактическим уровнем) может быть самой разнообразной. В частности, управляющая микро-ЭВМ или микропроцессор в каждом сочленении исполнительного механизма может выполнять функции вычисления управления, а также брать на себя частично функции привода, реализуя цифровую коррекцию.

Вернемся теперь к рассмотрению адаптивных робототехнических систем. Отметим сразу, что основное свойство адаптивных систем - выполнение задания в условиях недетерминированной внешней среды - требует использования в структуре системы, по крайней мере, двух модулей: обработки информации о внешней среде и генерации последовательности управляющих сигналов и данных для нижестоящего уровня вычисления управления.

Такое "наращивание" структуры приводит к тому, что переход от простых систем (например, программных роботов) к более сложным (адаптивным роботам) не влечет за собой коренных изменений ни в структуре их аппаратной реализации, ни в программном обеспечении, а дополняет их. Например, робот "Пума" (Puma), являясь программным роботом, тем не менее при подключении к нему телевизионной системы выполняет функции адаптивного робота.

Многоуровневая иерархическая организация адаптивной робототехнической системы тесно связана и во многом определяется той последовательностью задач, которые решает система на пути выполнения исходного задания, формируемого человеком оператором. На рис. 3.12 изображена функциональная организация адаптивной системы, разработанная в МВТУ им. Н. Э. Баумана, где показаны потоки информации и управления, а также фрагмент примера, иллюстрирующий расщепление

задачи на подзадачи. Что касается потоков информации, заметим следующее: каждый уровень иерархии может потребовать информацию о внешней среде, однако ее содержание безусловно разнится от уровня к уровню. Так, для верхнего уровня необходимая информация о рабочей сцене может заключаться в выяснении того, присутствует ли требуемый объект или нет, тогда как для нижних уровней могут быть необходимы данные о его расположении и ориентации. Этот факт выглядит совершенно естественно: каждый уровень потребляет только ту информацию, которая необходима для решения его задачи. Информационные модули  $I_1$ - $I_6$ , изображенные на рис. 3.12, занимаются приемом, переработкой и выдачей информации о рабочей сцене как своему собственному уровню иерархии, так и вышестоящему уровню. Рассмотрим кратко функции каждого уровня иерархии, его входы и выходы, начиная с верхнего уровня.

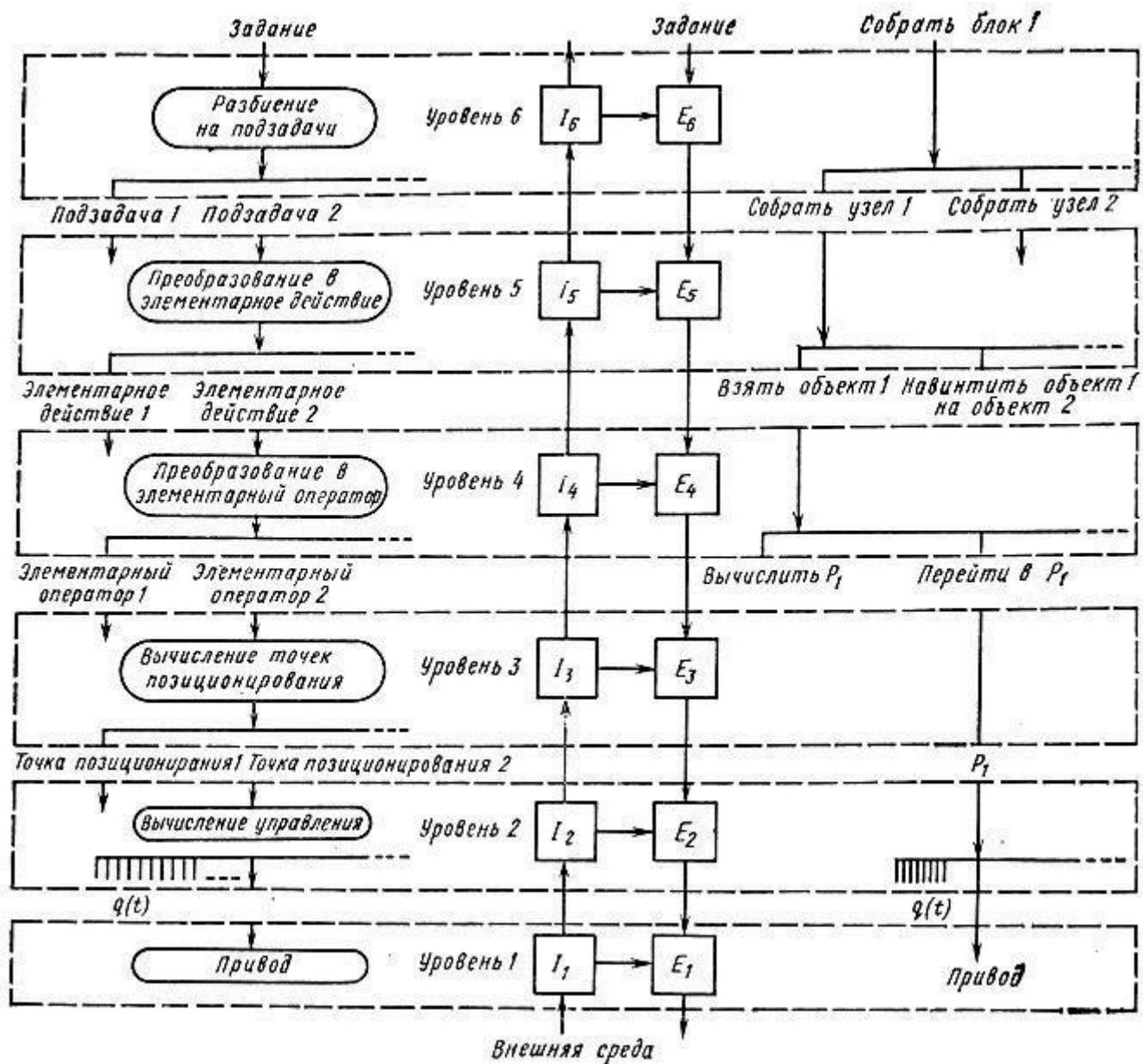


Рис. 3.12. Иерархическая структура адаптивного робота:  $I_1$ - $I_6$  - информационные модули;  $E_1$ - $E_6$  - исполнительные модули

### Уровень 6

**Вход:** задание, сформулированное человеком-оператором на некотором языке, понятном системе. В приведенном на рис. 3.12 примере это задание формулируется так:

## СОБРАТЬ БЛОК

**Выход:** последовательность подзадач, решающих в совокупности сформулированное задание. В примере эти подзадачи выглядят так:

СОБРАТЬ УЗЕЛ\_\_1

СОБРАТЬ УЗЕЛ\_\_2

*Функции:* разбиение сложной задачи на подзадачи.

*Инструментовка:* только ЦВМ; используемые методы - методы теории искусственного интеллекта.

*Уровень 5*

*Вход:* подзадача.

*Выход:* последовательность элементарных действий (или макрооператоров), выполнение которых обеспечит решение подзадачи. Для нашего примера действия, направленные на решение подзадачи 1, выглядят так:

ВЗЯТЬ ОБЪЕКТ\_\_1

НАВИНТИТЬ ОБЪЕКТ\_\_1 НА ОБЪЕКТ\_\_2

•  
•  
•

*Функции:* преобразование подзадачи в последовательность действий.

*Инструментовка:* только ЦВМ; используемые методы - методы теории искусственного интеллекта.

*Уровень 4*

*Вход:* элементарное действие.

*Выход:* последовательность элементарных операторов, реализующих это действие. Для нашего примера расширение макрооператора **ВЗЯТЬ ОБЪЕКТ\_\_1** имеет вид:

ВЫЧИСЛИТЬ P1  
ПЕРЕЙТИ\_\_В\_\_ТОЧКУ P1  
ЗАКРЫТЬ\_\_СХВАТ  
·  
·  
·

Заметим, что набор элементарных операторов имеет тот же смысл и принадлежит тому же уровню иерархии, о котором шла речь выше в связи с обсуждением рис. 3.10.

*Функции:* расширение макрооператоров на элементарные операторы.

*Инструментовка:* только ЦВМ; один из возможных способов получения элементарных операторов заключается в использовании методов макропроцессирования.

### Уровень 3

*Вход:* элементарный оператор.

*Выход:* описатели точек позиционирования, либо некоторые команды и данные, управляющие внешними устройствами.

*Функции:* трансляция (либо интерпретация) элементарного оператора.

*Инструментовка:* только ЦВМ, входные элементарные операторы представляют собой инструкции языка управления движением (мы их будем рассматривать ниже), поэтому используемые на этом уровне методы - по сути дела методы теории проектирования компиляторов.

### Уровень 2

*Вход:* описатель точки позиционирования.

*Выход:* управляющие сигналы на приводы подвижных сочленений.

*Функции:* декодировка входа; возможно, преобразование координат (например, преобразование из декартовых координат в обобщенные для антропоморфной кинематики исполнительного механизма, рассмотренного выше); построение интерполяционного многочлена; генерация управляющих сигналов.

*Инструментовка:* возможно, ЦВМ, хотя часть этих функций может быть реализована аппаратурно (например, генерация управляющих сигналов может быть реализована с помощью цифровых интеграторов).

### *Уровень 1*

*Вход:* управляющие сигналы на приводы подвижных сочленений.

*Выход:* изменение окружающей среды.

*Функции:* фактическое выполнение задания, сформулированного на уровне 6.

*Инструментовка:* функции цифровой коррекции может взять на себя ЦВМ (микро-ЭВМ или микропроцессор), хотя это не является обязательным.

Итак, мы рассмотрели ряд вопросов, связанных с организацией адаптивных робототехнических систем. Уровень адаптации робота может быть различным: от адаптации к параметрам среды (как, например, в примере с конвейером) до адаптации к структуре среды. Это и определяет сложность организации соответствующей адаптивной системы. Простая адаптивная система может содержать лишь несколько уровней иерархии; чем сложнее система, тем выше уровень ее адаптации, тем большее число уровней иерархии должно быть задействовано при ее реализации.

Зачем нужны уровни иерархии и стоит ли в отношении робототехнических систем пользоваться этими понятиями? В качестве ответа на этот вопрос можно было бы привести следующие доводы.

1. Иерархия объективно присуща робототехническим системам в такой мере, в какой она свойственна биологическим системам и, в частности, такой сверхсложной системе, какой является человек. Не следует забывать, что возлагая на робота некоторые функции человека, пусть самые простые, следует в какой-то мере придерживаться их прототипа.
2. Разбиение системы на подсистемы в соответствии с ее иерархической организацией существенно облегчает ее анализ. Действительно, никому, вероятно, не придет в голову учитывать динамические эффекты, возникающие в процессе движения исполнительного механизма, при решении задачи планирования его действий на уровне макрокоманд. Необходимо просто быть уверенным в том, что эти макрокоманды будут отработаны и об этом позаботятся нижестоящие уровни иерархии и, в частности, уровень привода. Заметим, что в отношении двух соседних уровней этого сказать нельзя. Например, уровень вычисления управления, о котором шла речь выше, должен знать возможности подчиненного ему уровня привода с тем, чтобы не подать ему на вход управляющие сигналы,

характеризующиеся такими скоростями и ускорениями, которые он не будет в состоянии отработать, аналогично уровень генерации макрокоманд должен быть понят уровнем их интерпретации (или компиляции), т. е., вообще говоря, уровень требований должен быть адекватен возможностям исполнителя.

3. Иерархия является конструктивным свойством робототехнических систем (в том числе и адаптивных) в том смысле, что ее используют при проектировании их программного обеспечения. Как будет показано ниже, процессы, развивающиеся в операционных системах адаптивных роботов, часто осуществляют те функции, которые выполняют соответствующие уровни иерархии. Это и понятно: уровень иерархии занят самостоятельной, четко очерченной задачей, в процессе решения которой он обменивается минимальной информацией с соседними ему уровнями, и этот набор характеристик является определяющим для выделения последовательности вычислительных операций в самостоятельный процесс.
4. Иерархический принцип организации робототехнических систем может обеспечить преемственность в их проектировании. Это означает, что переход от программного робота к адаптивному требует не коренной ломки их структур, а лишь надстройки необходимых модулей, которые придают им соответствующие свойства. Этот факт был отмечен выше и проиллюстрирован на рис. 3.9, 3.11 и 3.12, где показаны структурные схемы программного и адаптивного робота.

**Другой пример иерархической структуры.** Разработка адаптивных систем управления движением роботов — это сложная научно-техническая задача, решение которой складывается в основном из следующих этапов:

1. структурного синтеза адаптивной системы, включая выбор основных элементов (модулей), и распределения функций между ними;
2. алгоритмического построения элементов (модулей) с учетом их функционального назначения;
3. программно-аппаратурной реализации адаптивной системы на базе средств вычислительной техники.

Принципиальное значение первого этапа заключается в выборе и обосновании структуры адаптивной системы управления в целом, определяющей облик и функциональные возможности робота. На втором этапе синтезируются алгоритмы обработки сенсорной информации и адаптивного управления с учетом реальных характеристик имеющихся средств очувствления и конструктивных особенностей исполнительных механизмов робота. Тем самым создается специализированное алгоритмическое обеспечение модульной адаптивной системы управления для роботов определенного типа. Наконец, на третьем этапе решаются

вопросы программно-аппаратурной реализации синтезированных алгоритмических модулей на базе ЭВМ и микропроцессоров.

Рассмотрим пример разработанного в ЛГУ принципа построения адаптивной системы управления робота. Прежде всего заметим, что во многих случаях оказывается возможным свести сложную задачу адаптивного управления к совокупности более простых задач. Такая редукция составляет сущность принципа расщепления процесса управления, который лежит в основе структурного синтеза адаптивной системы. Она "подсказывает" иерархическую организацию процесса адаптивного управления; при этом принцип расщепления позволяет "расчленивать" трудности. Так, общая задача адаптивного управления движением робота естественным образом разделяется на четыре самостоятельные задачи [17, 61, 62].

Первая задача - формирование и оптимизация программного движения исполнительного механизма. Ее решение возлагается на автомат наведения, называемый еще программатором. Этот автомат реализует алгоритм, который с учетом цели движения, конструктивных ограничений и заданного критерия качества, автоматически строит нужное (в частности, оптимальное) программное движение. Тем самым осуществляется самопрограммирование движений робота.

Вторая задача - аналитический синтез идеального закона управления программным движением, гарантирующего желаемый характер переходных процессов в условиях, когда параметры робота и действующие на него возмущения известны. Эту задачу решает автомат стабилизации, называемый кратко регулятором.

Третья задача - построение алгоритмов адаптивной настройки параметров регулятора с целью компенсации неконтролируемого дрейфа параметров и внешних возмущений. Решение данной задачи возлагается на специальный автомат адаптации - адаптатор. Этот автомат реализует тот или иной алгоритм адаптации и осуществляет самонастройку параметров регулятора к фактическим условиям эксплуатации.

Синтез и оптимизация алгоритмов адаптации невозможны без контроля за качеством процесса управления. Поэтому четвертая задача сводится к разработке эффективных критериев качества адаптации. Такие критерии разумно задавать в форме неравенств, связывающих состояния, управления и настраиваемые параметры. Эти неравенства, определяющие, по существу, цель адаптации, будем называть эстиматорными неравенствами. Таким образом, функции оперативного контроля за качеством адаптации возлагаются на автомат, реализующий эстиматорные неравенства и

проверяющий их справедливость. Такой автомат будем кратко называть эстиматором.

Перечисленные выше "расщепленные" задачи неразрывно связаны друг с другом. Поэтому, подчеркивая важность взаимозависимости и соподчиненности приведенных выше задач, будем говорить об иерархической организации процесса их решения. Определим данное понятие с помощью иерархической схемы модульной системы адаптивного управления программным движением робота, показанной на рис. 3.13, которая выражает, по существу, общую концепцию адаптивного управления движением в условиях неполной информации и наличия разного рода возмущений. Программатор, регулятор, адаптатор, эстиматор выступают как основные элементы адаптивной системы управления, определяющие его структуру и функциональные возможности. На вход этих элементов-модулей подаются сигналы обратной связи, формируемые информационно-измерительной (сенсорной) системой.

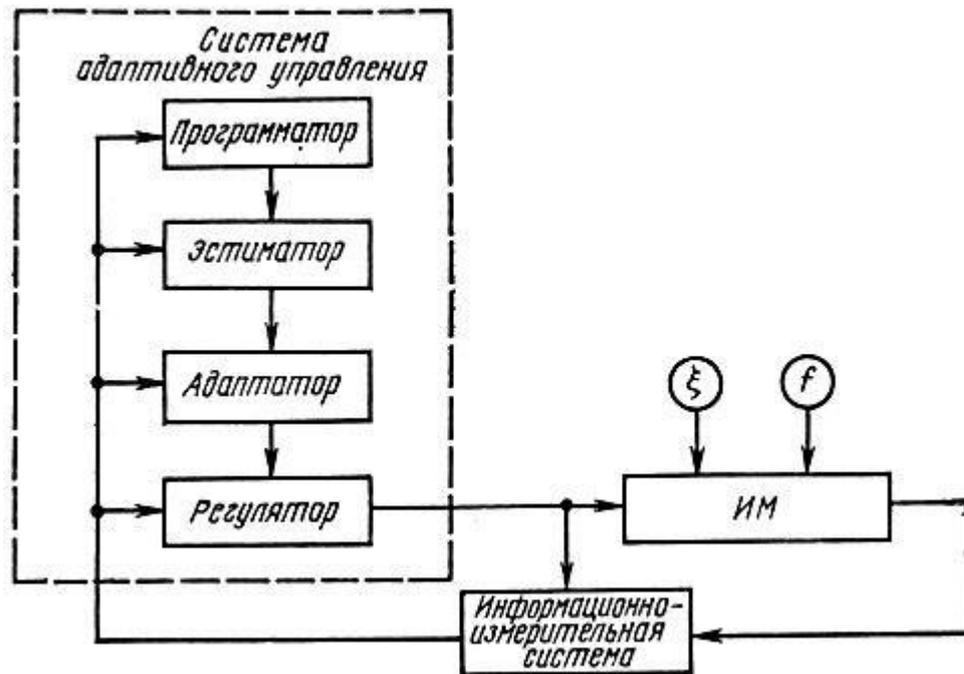


Рис. 3.13. Модульная структура системы адаптивного управления движением робота:  $\xi$  - отклонения параметров объекта и рабочей нагрузки;  $f$  - возмущающие воздействия; ИМ - исполнительный механизм