

В. Н. ДРОЗДОВ
И. В. МИРОШНИК
В. И. СКОРУБСКИЙ

Системы
автоматического
управления
с микроЭВМ



ЛЕНИНГРАД
«МАШИНОСТРОЕНИЕ»
ЛЕНИНГРАДСКОЕ ОТДЕЛЕНИЕ
1989

ББК 32.965

Д75

УДК [681.3—181.4 + 681.326.32].004.14:658.512.011.56

Рецензент проф. *В. А. Бесекерский*

Дроздов В. Н. и др.

Д75 Системы автоматического управления с микроЭВМ/
В. Н. Дроздов, И. В. Мирошник, В. И. Скорубский. — Л.:
Машиностроение. Ленингр. отд-ние, 1989. — 284 с.: ил.
ISBN 5-217-00557-2

В книге рассмотрены общие принципы построения локальных систем управления на базе микропроцессоров и микроЭВМ, аспекта этой проблемы: синтез алгоритмов управляющих микроЭВМ на основе временных методов; аппаратную реализацию цифровых регуляторов на микроЭВМ и микропроцессорах; программное обеспечение управляющих микроЭВМ. Приведенные в книге методы синтеза алгоритмов в значительной степени формализованы и могут быть использованы в автоматизированных системах проектирования систем автоматического управления. Основные положения проиллюстрированы примерами.

Книга предназначена для инженерно-технических работников, занимающихся разработкой цифровых систем управления.

Д $\frac{240500000-051}{038(01)-89}$ 51—89

ББК 32.965

ISBN 5-217-00557-2 © Издательство «Машиностроение», 1989

ПРЕДИСЛОВИЕ

В научно-технической литературе последних десятилетий наблюдался определенный разрыв между исследованиями в области теории автоматического управления и прикладными разработками. Внедрение математических методов обеспечило опережающее развитие теории дискретных систем, оптимального управления, методов фильтрации и наблюдения, теории устойчивости, инвариантности. Фактически была создана научная база современных САУ, с самого начала ориентированная на разработку алгоритмического обеспечения управляющих ЭВМ.

Появление в 70-х гг. микропроцессоров и микроЭВМ справедливо оценивается специалистами как революционное событие, позволившее перейти к активному использованию обширного теоретического задела. Этому в немалой степени способствуют возросший математический и системотехнический уровень инженерно-технических работников и, что наиболее важно, потребности технического прогресса и в первую очередь комплексной автоматизации производства. Особую актуальность управление на основе микроЭВМ приобретает в связи с развитием гибкого автоматизированного производства, базирующегося на использовании станков-автоматов, роботов и автоматических транспортных средств, а также с переводом на цифровое управление энергетических и больших приборных систем.

Усложнение автоматизируемого оборудования и повышение требований к его техническим характеристикам определяет необходимость создания нового поколения цифровых САУ, обеспечивающих достижение оптимальных качественных показателей управляемых процессов с учетом особенностей современных управляющих микроЭВМ, т. е. ограничений, накладываемых вычислительной аппаратурой и программным способом формирования управляющего воздействия. Игнорирование специфики технических средств приводит либо к ухудшению характеристик САУ, либо к необоснованному завышению требований к быстродействию, разрядности и объему памяти ЭВМ, что в конечном итоге сказывается на технико-экономических показателях разрабатываемого оборудования.

Естественно возникает задача приспособления современных методов теории автоматического управления к существующим и

перспективным средствам вычислительной техники, т. е. задача синтеза эффективных машиноориентированных алгоритмов управления, составляющих алгоритмическое обеспечение цифровой САУ. Для решения этой задачи наряду с методами теории дискретных систем используется принцип декомпозиции. Широкое распространение получила *пространственная декомпозиция* — разбиение сложной динамической системы на локальные САУ, основными элементами которых становятся серийные управляющие микроЭВМ (программируемые микроконтроллеры).

Концепция *временной декомпозиции* предусматривает выделение последовательно сменяющих друг друга локальных целей управления и соответствующих режимов работы САУ, а затем построение комплекса регуляторов (алгоритмов управления), обслуживающих систему в этих режимах. В микропроцессорных системах переключение алгоритмов управления с изменением режима работы САУ осуществляется программными методами и не требует расширения технических средств. Концепция временной декомпозиции может оказаться весьма продуктивной для повышения качества САУ технологического оборудования и при решении некоторых базовых задач управления. Так, в задачах терминального управления оптимальное быстродействие САУ может быть достигнуто за счет последовательной оптимизации управления по более простым локальным критериям.

Разработка алгоритмов управления является первым этапом проектирования цифровой системы, за которым следует комплектование технических средств и подготовка прикладных программ. От того, в какой мере существующие средства вычислительной техники соответствуют потребностям управленческих задач, зависит практическая эффективность синтезированных алгоритмов. Необходимость развития и проблемной ориентации цифровой аппаратуры, операционных систем реального времени и средств автоматизации программирования требуют объединения знаний и усилий специалистов по вычислительной технике и автоматическому управлению, преодоления расхождений в концепциях, методологии и терминологии ведущих кибернетических дисциплин. При написании книги авторы преследовали следующие цели: предложить читателю методы разработки алгоритмического обеспечения современных цифровых систем управления; изложить принципы построения и программирования микропроцессорных САУ, организации взаимодействия аппаратуры и программных средств управляющих микроЭВМ, а также наметить пути повышения эффективности алгоритмического, программного и аппаратного обеспечения.

Центральное место в книге занимают вопросы синтеза машинно-ориентированных алгоритмов управления — формализованных правил расчета управляющих воздействий в реальном масштабе времени. Алгоритмы позволяют решать три базовые задачи автоматизации: стабилизации, слежения (программного управления) и

терминального управления. На их основе могут быть построены и более сложные многорежимные системы управления технологическим оборудованием.

Использование метода пространства состояний в наибольшей степени отвечает специфике рассматриваемых задач и способам организации вычислений в микропроцессорных системах. Предлагаемые процедуры синтеза ориентированы на широкое применение ЭВМ и служат основой алгоритмического обеспечения САПР систем управления. Рациональное перераспределение функций между процедурами синтеза и алгоритмами управления позволяет упростить расчеты реального времени, уменьшить интервал квантования и требования к быстродействию аппаратных средств САУ.

Наряду с широким спектром линейных алгоритмов в книге представлены нелинейные алгоритмы терминального управления, обеспечивающие оптимальное функционирование САУ в условиях больших начальных рассогласований, ограничений, накладываемых на состояние объекта управления и его входные воздействия. Основным теоретическим аппаратом для решения подобных задач служит принцип максимума Понтрягина и метод Гамкредидзе, которые не позволяют однако непосредственно перейти к синтезу замкнутых алгоритмов управления. В связи с этим в книге сформулированы дополнительные теоретические положения, необходимые для нахождения оптимальных решений.

Значительное внимание в книге уделяется принципам организации технических и программных средств управляющих микроЭВМ; анализируются характеристики аппаратуры, наиболее важные для решения задачи управления — разрядность, комплектность ЭВМ и устройств сопряжения с объектом; рассматриваются особенности программной реализации локальных алгоритмов управления и переключения структуры, а также системные средства реального времени и автоматизации программирования.

Изложение материала базируется на программах технического вузов по курсам математики, теории автоматического управления и вычислительной техники. При этом авторы стремились к разумному компромиссу между математической строгостью основных результатов, обеспечивающей необходимую корректность процедур синтеза, а в конечном итоге — адекватное качество управления, и доступностью изложения для специалистов-разработчиков. Лучшему пониманию материала должны способствовать также примеры, иллюстрирующие основные положения работы.

В основу книги легли теоретические и прикладные исследования, проведенные авторами на кафедрах автоматки и телемеханики и вычислительной техники Ленинградского ордена Трудового Красного Знамени института точной механики и оптики. Предисловие и гл. 1 написаны авторами совместно; гл. 2 и 3 — В. Н. Дроздовым, гл. 4 — И. В. Мирошником, а гл. 5 и 6 — В. И. Скорубским.

СПИСОК ОСНОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

x — вектор состояния объекта управления	ЗУ — запоминающее устройство
\bar{x} — редуцированный вектор состояния или вектор продольного движения	ИРПР — интерфейс радиальный параллельный
u — вектор управления	ИРПС — интерфейс радиальный последовательный
y — вектор регулируемых или выходных переменных	КГ — командный генератор
ξ — вектор состояния командного генератора	МП — микропроцессор
g — вектор задающих воздействий	ОЗУ — оперативное ЗУ
e — вектор отклонений по состоянию или вектор относительного движения	ОП — основная память
e — вектор отклонений по регулируемым переменным	ОС — операционная система
r — символ режима работы	ОС РВ — ОС реального времени
S — многообразие или гиперповерхность	ОУ — объект управления
T — интервал квантования	ПДП — прямой доступ к памяти
АЛБ — арифметико-логический блок	ПП — планирующая программа
АЦП — аналого-цифровой преобразователь	ПЗУ — постоянное ЗУ
БИС — большая интегральная схема	РОН — регистр общего назначения
	Р — регулятор
	САУ — система автоматического управления
	УСО — устройства сопряжения с объектом
	ЦАП — цифроаналоговый преобразователь
	ЦВВ — цифровые входы — выходы
	ША — шина адресная
	ШИ — шина информационная

Глава 1

ПРИНЦИПЫ ПОСТРОЕНИЯ И ПРОЕКТИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ С МИКРОЭВМ

1.1. Задачи автоматического управления

Современное производство характеризуется разнообразием технологических процессов, оснащенных автоматическим оборудованием. Стремление учесть как можно больше особенностей конкретных процессов ориентирует разработчиков на создание узкоспециализированных систем управления, плохо приспособляющихся к изменяющимся условиям производства, модернизации технологического оборудования и использованию в смежных областях. Значительные затраты на проектирование, изготовление и эксплуатацию уникальных систем обуславливают их низкую экономическую эффективность. Этим объясняется возросший интерес специалистов к унифицированным средствам, позволяющим решать широкий класс качественно однородных задач управления в различных производственных объектах. Элементной базой таких средств служат микроЭВМ и микропроцессорные наборы, обладающие практически неограниченными возможностями изменения режимов управления.

Анализ наиболее общих системных свойств технологического оборудования позволяет выявить основные типы *объектов управления* (ОУ), сформулировать характерные задачи *систем автоматического управления* (САУ) и требования, предъявляемые к качеству их функционирования. С этой точки зрения проанализируем некоторые образцы наиболее типичного для современного производства технологического оборудования.

Станки-автоматы. Основу механообрабатывающего производства составляют автоматические металлорежущие станки (токарные, фрезерные, шлифовальные и универсальное оборудование) и обрабатывающие центры [16]. Использование сменного инструмента, хранящегося в револьверных головках и инструментальных магазинах, позволяет осуществлять многооперационную обработку однотипных деталей. Система управления автоматического станка обеспечивает выполнение в заданной последовательности технологических и вспомогательных операций, таких как съем детали и инструмента, внеоперационный контроль размеров, перемещение детали и инструмента в зону обработки. Элементы

станка (рис. 1.1) приводятся в движение исполнительными устройствами, в состав которых входят электроприводы главного движения и подачи, а также механизмы и муфты вспомогательных операций. Управляемый электропривод содержит усилители мощности, исполнительные электродвигатели и различного рода механические передачи.

Электропривод главного движения станка поддерживает необходимую для данной операции скорость вращения заготовки или инструмента. В функции электроприводов подачи входит обеспечение позиционирования и перемещения режущего инструмента относительно детали, в ходе которого образуется заданный профиль изделия. Эти операции

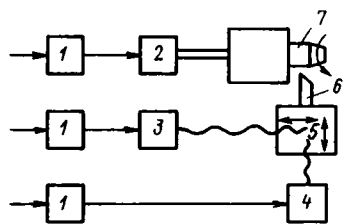


Рис. 1.1. Схема электропривода токарного станка:

1 — усилители мощности; 2 — электродвигатель главного движения; 3 — электродвигатель продольной подачи; 4 — электродвигатель поперечной подачи; 5 — суппорт; 6 — инструмент; 7 — деталь

осуществляются перемещением суппорта (в станках токарной группы) и стола (в фрезерных, шлифовальных станках). Работа привода подачи определяет в конечном итоге скорость резания и форму обрабатываемой поверхности. Функции закрепления — смены инструмента и заготовок выполняются гидро-, пневмо- или электроприводом соответствующих механизмов, а также электромагнитными муфтами. Эти устройства осуществляют ряд простейших движений и работают в стартстопном режиме.

Все функции управления исполнительными механизмами автоматических станков выполняются системой управления. Можно выделить следующие задачи, решаемые САУ одновременно или последовательно в ходе многооперационной обработки детали: быстрое перемещение инструмента (детали) в зону обработки; стабилизация угловых и линейных скоростей движения элементов станка, а также их положения в пространстве; контурное перемещение инструмента (детали); включение, реверс и выключение стартстопных механизмов.

Объектами управления в данном случае служат инерционные массы элементов станка, приводимые в движение соответствующими исполнительными устройствами. Влияние процессов металлообработки на объект управления проявляется в виде моментов и сил сопротивления, приложенных к исполнительным механизмам.

Кроме перечисленных выше локальных задач, САУ осуществляет автоматическое переключение режимов работы исполнительных устройств в соответствии с технологическим процессом. Многорежимный характер работы является основной особенностью рассматриваемого класса объектов управления и во многом определяет организацию САУ. Для измерения переменных, ха-

рактизирующих текущее состояние объекта управления, в автоматических станках используются измерители скоростей, линейных и угловых перемещений (аналоговые, цифровые, импульсные), датчики моментов, дальномеры и фотоэлектрические преобразователи технического зрения, а также измерители электрических величин электроприводов (токов, напряжений). Система автоматического управления осуществляет обработку текущей информации о состоянии объекта и формирование управляющих сигналов, поступающих на исполнительные элементы оборудования.

Непрерывные технологические линии. В различных отраслях промышленности применяются агрегаты непрерывного действия для обработки ленточного материала [27]. К ним относятся станы тонколистной прокатки, бумагоделательные машины, поточные линии химической и текстильной промышленности. Эти агрегаты осуществляют формирование профиля изделия, нанесение покрытия, термообработку полосы и т. д. Несмотря на широкий диапазон назначения и конструктивные особенности таких систем их объединяет принцип последовательного расположения зон обработки, основных исполнительных механизмов (рис. 1.2) и необходимость транспортировки ленточного материала. В прокатных станах и бумагоделательных машинах перемещение является технологической операцией, за счет которой осуществляется формирование профиля изделия. В других случаях движение ленты обеспечивает необходимый режим работы технологического оборудования в зонах термообработки, нанесения покрытия и т. д. Это позволяет выделить класс непрерывных технологических линий, объединенных общим функциональным назначением и, следовательно, принципами построения САУ.

Исполнительными механизмами технологических линий служат электроприводы намоточных устройств и валков, а также приводы и механизмы вспомогательных операций (прижима валков, фиксации рулонов и т. д.). Электроприводы осуществляют намотку и размотку рулонов, регулировку скорости движения ленты, поддержание натяжения отдельных ее участков на заданном уровне. Основными функциями САУ таких систем является управление электроприводами и стартстопными механизмами, что предусматривает: быстрый разгон и остановку ленты; стабилизацию или регулирование по заданному закону скорости перемещения ленты; стабилизацию или программное регулирование натяжения ленты; переключение стартстопных механизмов. При осуществлении многооперационной и циклической (многократной) обработки ленточного материала, характерной для современных поточных линий, система управления обеспечивает поддержание заданной последовательности режимов работы, в каждом из которых электропривод может выполнять различные функции.

Объектом управления технологической линии является многодвигательный электропривод, отдельные каналы которого связаны через обрабатываемый материал. Это определяет многосвяз-

ность задач управления и в общем случае исключает возможность автономного управления электродвигателями. Нагрузкой электропривода служат инерционные массы валков и рулонов. Причем изменение радиусов рулонов в процессе функционирования системы влечет за собой изменение инерционных свойств нагрузки электроприводов намоточных устройств, т. е. параметров электроприводов. Возмущающее воздействие технологического процесса в зонах обработки и взаимное влияние отдельных каналов проявляется в качестве сил и моментов сопротивления, приложенных к исполнительным механизмам. Для съема информации о состоянии системы транспортировки как объекта управления

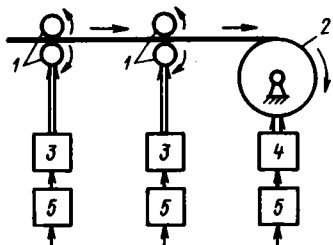


Рис. 1.2. Схема непрерывной технологической линии:

1 — валки; 2 — намоточное устройство; 3 — электродвигатели валков; 4 — электродвигатель намоточного устройства; 5 — усилители мощности

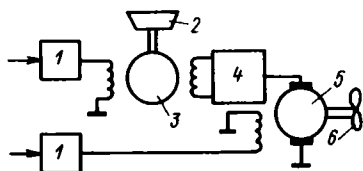


Рис. 1.3. Схема гребной электрической установки:

1 — усилители мощности; 2 — тепловой двигатель; 3 — генератор; 4 — выпрямитель; 5 — гребной электродвигатель; 6 — гребной винт

используются измерители угловых и линейных скоростей, датчики напряжений, усилий, моментов, а также измерители электрических величин электроприводов.

Гребные электрические установки. Одно из наиболее перспективных направлений в судостроении предполагает использование для движения судов *гребных электрических установок* (ГЭУ) [81]. Они позволяют обеспечить широкий диапазон изменения скорости гребного вала, экономичность и маневренность судна; что особенно важно для объектов, работающих в тяжелых эксплуатационных условиях.

Основным компонентом ГЭУ является электропривод (постоянного, переменного или двойного рода тока). В его состав входит тепловой двигатель, приводящий в движение электромашинный усилитель, электродвигатель и усилительно-преобразовательные устройства. На рис. 1.3 представлен электропривод двойного рода тока, усилитель главной цепи которого выполнен на базе генератора переменного напряжения, связанного с якорной обмоткой двигателя постоянного тока через преобразователь. Электропривод содержит также тиристорные усилители обмоток возбуждения двигателя и генератора.

Электропривод ГЭУ осуществляет регулирование угловой скорости гребного вала в условиях волнения, заклинивания винта и изменения сопротивления движению судна. Перечисленные факторы могут служить причиной перегрузки первичного двигателя по мощности и значительного (до 10 крат) превышения допустимого значения тока в главной цепи электродвигателя. При перегрузках возникает необходимость введения дополнительных режимов работы ГЭУ — режима стабилизации мощности на гребном валу и тока главной цепи (тока якоря для электродвигателей постоянного тока). Для судов, требующих частых переключений скорости гребного вала, например при швартовке или движении во льдах, особое значение приобретает возможность осуществлять быстрые изменения скорости.

В функции системы управления ГЭУ входят [96]: наискорейший разгон гребного вала до заданной угловой скорости и его реверс; стабилизация скорости вращения вала, мощности на гребном валу и тока главной цепи. Кроме этого, система управления организует своевременное переключение режимов работы по командам с поста управления или измерителей, а также функции контроля и аварийного отключения ГЭУ в экстремальных ситуациях.

Управление электроприводом ГЭУ может осуществляться как по главной цепи, так и по каналу возбуждения двигателя, причем в различных режимах используются разные цепи управления. Отметим, что для ГЭУ с нереверсивным каналом якорной цепи реверс электродвигателя производится только по цепи обмотки возбуждения.

Нагрузкой ГЭУ, как объекта управления, является гребной вал с приложенным к нему моментом сопротивления среды. Характерной особенностью рассматриваемых систем является многообразие типов возмущающих воздействий в различных режимах движения судна. Это, в первую очередь, моменты вязкого трения, зависящие от угловой скорости вала, скорости движения судна и волнения. Кроме того, выходной вал испытывает дополнительное сопротивление при движении во льдах, принимающее иногда скачкообразный характер.

Необходимая для управления информация о состоянии ГЭУ обеспечивается комплексом контрольно-измерительной аппаратуры, в состав которого входят импульсные измерители угловой скорости, датчики токов и напряжений. В ряде случаев целесообразно измерение моментов нагрузки и использование информации о скорости движения судна.

Электропривод промышленных установок. В современном производстве автоматизированный электропривод является основным регулируемым источником механической энергии и, следовательно, важным объектом изучения. Главными компонентами электропривода являются электродвигатели, связанные друг с другом через внешнюю среду. Управление двигателями осуществляется через

усилительно-преобразовательные устройства главных цепей и обмоток возбуждения. Как динамический объект электропривод описывается системой дифференциальных уравнений, которая для приводов с двигателями постоянного тока включает [97]:

уравнения динамики электродвигателя

$$\frac{d\alpha}{dt} = \omega; \quad (1.1)$$

$$J \frac{d\omega}{dt} = k_d I_\alpha I_B - M; \quad (1.2)$$

уравнение якорной цепи

$$T_\alpha \frac{dI_\alpha}{dt} + I_\alpha = \frac{1}{r_\alpha} (U_\alpha - k_e I_B \omega); \quad (1.3)$$

уравнение усилительно-преобразовательного устройства якорной цепи

$$T_\gamma \frac{dU_\gamma}{dt} + U_\gamma = k_{\gamma 1} v_1; \quad (1.4)$$

уравнение цепи обмотки возбуждения

$$T_B \frac{dI_B}{dt} + I_B = k_{\gamma 2} v_2, \quad (1.5)$$

где α — угол поворота выходной оси; J — приведенный момент инерции; ω — угловая скорость выходной оси; I_α — ток якоря; I_B — ток обмотки возбуждения двигателя; M — приведенный момент сопротивления среды; U_γ — выходное напряжение усилителя якорной цепи; v_1, v_2 — входные (управляющие) сигналы цепи тока якоря и возбуждения соответственно; T_α — постоянная времени усилителя якорной цепи; T_B — постоянная времени обмотки возбуждения электродвигателя; r_α — сопротивление якорной цепи; $k_d, k_e, k_{\gamma 1}, k_{\gamma 2}$ — постоянные коэффициенты.

Если в соответствии с кинематической схемой вращательное движение выходной оси преобразуется в поступательное движение l , то описание электропривода дополняется уравнением связи:

$$l = k_p \alpha, \quad (1.6)$$

где k_p — коэффициент редукции.

Влияние на электродвигатель приводимых в движение элементов промышленного оборудования и внешних факторов учитывается через моменты сопротивления M , которые могут быть представлены постоянной составляющей $M = M_0$ или изменяться в функции угловой скорости $M = M(\omega)$, времени $M = M(t)$ и т. д. В многодвигательных электроприводах в состав возмущающих факторов часто входят составляющие моментов сопротивления, вызванные взаимодействием приводов через обрабатываемый материал или среду.

Переменные уравнений (1.1)—(1.6) имеют различную физическую природу и должны быть выражены в абсолютных единицах измерения угловых или линейных перемещений, скоростей, токов и т. д. В цифровых САУ информация об этих переменных, полученная с помощью измерительных устройств, преобразуется в двоичные коды, с которыми и оперирует управляющая микро-ЭВМ. Тем самым физические переменные объекта управления преобразуются к кодированным машинным переменным, выраженным в относительных единицах измерения. Для аналогичного преобразования уравнений динамики электропривода введем в рассмотрение безразмерные величины: $x_1 = \alpha/\alpha_6$ ($x_1 = l/l_6$), $x_2 = \omega/\omega_6$, $x_3 = I_{я}/I_{я6}$, $x_4 = U_y/U_{y6}$, $x_5 = I_{в}/I_{в6}$, $u_1 = v_1/v_{16}$, $u_2 = v_2/v_{26}$, $\mu = M/M_6$, где индексом «6» отмечены базовые значения соответствующих переменных, в качестве которых часто выбирают номинальные параметры электропривода. Тогда уравнения динамики привода можно записать в виде

$$\dot{x}_i = f_i(x_1, \dots, x_5, u_1, u_2, \mu), \quad i = \overline{1, 5},$$

где функции $f_i(\cdot)$ нетрудно получить из уравнений (1.1)—(1.5).

Минимальный набор относительных переменных состояния x_i , необходимый для описания поведения объекта управления в динамике, образует *вектор состояния* $OУ$. Для однодвигательного электропривода он формируется как $\mathbf{x} = \{x_i\}$, $i = \overline{1, 5}$. Управляющие воздействия u_1 и u_2 образуют *вектор управления* $\mathbf{u} = \{u_j\}$, $j = 1, 2$. Для многодвигательных систем размерности векторов \mathbf{x} и \mathbf{u} соответственно увеличиваются, а возмущающие воздействия, приложенные в различных точках объекта, образуют *вектор возмущения* μ .

Связь вектора состояния \mathbf{x} с векторами управления \mathbf{u} и возмущения μ задается векторным дифференциальным *уравнением состояния* $OУ$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mu), \quad (1.7)$$

где $\mathbf{f}(\cdot) = \{f_i(\cdot)\}$, $i = \overline{1, 5}$.

Регулируемой переменной объекта управления в зависимости от назначения электропривода может служить угол поворота α , линейное перемещение l или угловая скорость ω . Регулируемая (или выходная) переменная y , выраженная в относительных единицах, определяется, соответственно, как $y = x_1$ или $y = x_2$. В многодвигательном электроприводе регулируемые переменные отдельных каналов образуют *вектор регулируемых (или выходных) переменных*, который связан с вектором состояния *уравнением выхода*

$$y = \mathbf{c}(\mathbf{x}). \quad (1.8)$$

Таким образом, полная модель объекта управления задается системой уравнений (1.7), (1.8) (рис. 1.4). Наличие достаточно точ-

ных моделей управляемого процесса и возмущающих факторов является необходимым условием построения высококачественной САУ.

Качество управления зависит также от полноты и достоверности информации о состоянии ОУ. На практике доступной измерению оказывается лишь часть переменных системы, причем измерения физических величин осуществляются с погрешностью, вызванной неточностью измерителей и влиянием шумов измерения. Отсутствие адекватной информации о состоянии ОУ и внешней среды вызывает необходимость построения *устройств оценивания* — фильтров, позволяющих восстановить

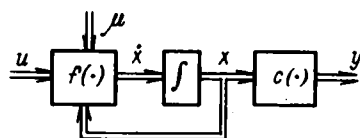


Рис. 1.4. Структурная схема объекта управления

векторы x и μ .

Задачи систем автоматического управления. Рассмотренные примеры позволяют составить достаточно полное представление о характере задач управления, решаемых САУ. К ним относятся:

задача стабилизации угловой скорости ω , угла поворота α или перемещения l на заданном уровне $\omega = \omega_y$, $\alpha = \alpha_y$ или $l = l_y$, в общем случае формализуемая как *задача стабилизации выхода на уровне g*

$$y = g = \text{const}$$

или вектора состояния в заданной точке x^*

$$x = x^* = \text{const};$$

задача управления переменными ω , α или l по заданной программе ($\omega = \omega(t)$, $\alpha = \alpha(t)$, $l = l(t)$) или *задача программного управления* общего вида

$$y = g(t) \text{ или } x = x^*(t);$$

задача терминального управления, заключающаяся в переводе объекта управления из начального состояния $x(0) = x_0$ в заданное конечное $x(t_f) = x_f$ за фиксированный или произвольный отрезок времени t_f .

Кроме управления непрерывными процессами в промышленных системах могут быть выделены также задачи дискретной автоматики, связанные с управлением стартовыми механизмами и переключением режимов работы оборудования. Требования к поведению объектов управления задаются расписаниями, циклограммами или временными диаграммами, устанавливающими последовательность и правила переключения дискретных состояний. При постановке задачи управления необходимо принять во внимание *ограничения* на переменные состояния объекта и управляющие воздействия. Необходимость введения ограничений на координаты x_i вектора состояния x обусловлена ограниче-

ниями рабочего пространства механизмов по углам поворота α или перемещениям l

$$x_{1 \min} \leq x_1 \leq x_{1 \max}, \quad (1.9)$$

ограничениями технологического характера на скорости (резания, обработки ленты) движения ω

$$x_{2 \min} \leq x_2 \leq x_{2 \max}, \quad (1.10)$$

ограниченными перегрузочными способностями электродвигателей по току якоря $I_{я}$

$$|x_3| \leq x_{3 \max}, \quad (1.11)$$

току обмотки возбуждения $I_{в}$

$$|x_5| \leq x_{5 \max}, \quad (1.12)$$

мощности $P(x_2, x_3, x_5) = x_2 x_3 x_5$

$$P(x_2, x_3, x_5) \leq P_{\max}, \quad (1.13)$$

и т. д.

Ограниченные энергетические возможности элементов объекта управления определяют необходимость ограничения управляющих воздействий неравенствами типа

$$|u_j| \leq U_j, \quad j = 1, 2, \dots \quad (1.14)$$

Выполнение ограничений в ходе функционирования объекта управления является одной из первостепенных задач САУ.

Рассмотрим далее наиболее общие требования к процессам управления и критерии оценки качества системы в ходе решения типовых задач. Качество САУ определяется комплексом точностных и динамических показателей. Точностные показатели устанавливают предельно допустимые значения ошибок управления в установившихся режимах, т. е. после завершения переходного процесса. Динамические показатели определяют характер переходного процесса. Для систем стабилизации и программного управления вводится понятие *отклонения движения (ошибки)* по выходной переменной $e = g - y$ или по состоянию $e = x^* - x$. Точностные требования устанавливаются неравенствами вида: $|e| \leq \epsilon_{\max}$ или $\|e\| \leq \epsilon_{\max}$. К динамическим свойствам относится в первую очередь устойчивость САУ, т. е. ее способность возвращаться к установившемуся режиму при малых начальных отклонениях, вызванных кратковременными воздействиями различных возмущающих факторов. Обеспечение устойчивости является необходимым условием нормального функционирования системы.

Высокое качество технологических процессов, необходимое для большинства образцов современного оборудования, предполагает введение дополнительных динамических показателей, к которым относятся время переходного процесса, перерегулирование и различного рода интегральные критерии. Научно обоснованная формулировка показателей качества и возможность их формали-

зации являются необходимыми предпосылками оптимизации процессов управления в рамках имеющихся информационных и энергетических ресурсов системы.

Характерной особенностью системы стабилизации и программного управления является работа в области малых отклонений ϵ или ϵ . Это позволяет использовать линеаризованное описание объекта и пренебречь ограничениями, накладываемыми на координаты вектора состояния и управляющие воздействия. Жесткие требования по точности вызывают необходимость серьезного анализа возмущающих воздействий.

В терминальных задачах обычно предполагается, что начальные отклонения от заданного состояния объекта велики. Поэтому решение этих задач требует ощутимых затрат времени и энергетических ресурсов. Естественными критериями качества таких систем становятся быстродействие (время переходного процесса) и затраты энергии или топлива, причем в полной мере проявляются ограничения физических переменных объекта (скоростей, токов и т. д.), а также управляющих воздействий.

1.2. Принципы построения систем управления

Использование аналоговых средств реализации и частотных методов синтеза определило широкое распространение систем автоматического управления с пропорционально-интегрально-дифференциальными законами регулирования (ПИД-регуляторами). Усложнение технологического оборудования, повышение требований к качеству управляемых процессов, а также появление дешевых микропроцессоров и разработка новых разделов теории управления обусловили развитие нового поколения цифровых систем управления. Эти системы отличает широкое использование преимуществ иерархических структур и принципов многорежимного управления, применение значительно более сложных типов локальных регуляторов (многоконтурных, нелинейных, адаптивных). Изменение принципов построения систем управления с ростом технического уровня автоматизируемого оборудования, средств реализации и развитием теории автоматического управления является закономерным процессом.

Одноконтурные и многоконтурные системы. На рис. 1.5 представлена структурная схема одноконтурной системы управления. В ее состав входят элемент сравнения, регулятор (P) и объект управления. Система обеспечивает стабилизацию регулируемой величины y на заданном уровне или обработку простейших типов задающих воздействий $g(t)$. В функции регулятора входит преобразование информации об ошибке $\epsilon = g - y$ в управляющий сигнал объекта u в соответствии с законом (алгоритмом) управления:

$$u = u_{\epsilon}(\epsilon), \quad (1.15)$$

где $u_e(\cdot)$ — оператор преобразования, т. е. правило, в соответствии с которым рассчитывается текущее значение u .

Обычно используемые ПИД-регуляторы реализуют закон управления

$$u = K(p) e, \quad (1.16)$$

где $K(p)$ — дробно-рациональная функция p ; $p = d/dt$ — оператор дифференцирования.

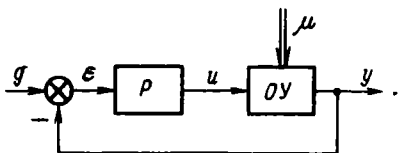


Рис. 1.5. Структурная схема одно-контурной САУ

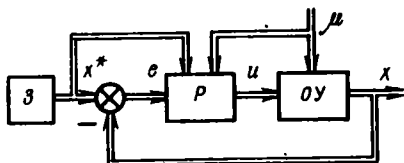


Рис. 1.6. Структурная схема много-контурной САУ

Его каноническая структура при $K(p) = k_1 + k_2/p + k_3p$ определяется выражением

$$u = k_1 e + k_2 \int_0^t e \, d\tau + k_3 \frac{de}{dt}, \quad (1.17)$$

т. е. управляющее воздействие кроме текущей информации о расхождении содержит информацию о «прошлом» системы (интегральная составляющая) и о поведении объекта в будущем (дифференциальная компонента).

Такое воздействие на объект позволяет частично компенсировать влияние возмущающих воздействий и предупредить резкие изменения управления u , приводящие к колебаниям системы.

Повышение требований к точности и динамическим показателям качества управляемых процессов, в частности при обработке сложных программных законов $x^* = x^*(t)$, обуславливает необходимость построения *многоконтурных САУ*. Такие системы кроме информации об основной регулируемой переменной используют измерения других координат вектора состояния и внешних воздействий. В технике наибольшее распространение получили системы подчиненного регулирования, в которых управление по нескольким переменным осуществляется по замкнутой схеме, с использованием одноконтурных ПИД-регуляторов.

Современная теория управления предоставляет широкие возможности синтеза многоконтурных регуляторов, обеспечивающих достижение оптимальных качественных показателей САУ. Среди них регуляторы состояния, динамические и адаптивные регуляторы [28, 75]. За счет дополнительной информации о состоянии объекта управления, а также о внешних воздействиях появляется возможность существенно улучшить точностные и динамические показатели системы. Общая структурная схема

многоконтурной системы приведена на рис. 1.6. Она содержит задающее устройство (З), предназначенное для воспроизведения программных сигналов x^* , многоконтурное устройство сравнения и регулятор, объект управления. Закон управления, реализуемый регулятором, определяется выражением

$$u = u(e, x^*, \mu), \quad (1.18)$$

где $e = x^* - x$ — вектор ошибки по состоянию; μ — вектор возмущений.

Многорежимные системы управления. Важнейшей системной особенностью современного технологического оборудования является необходимость последовательного обеспечения нескольких задач управления, возникающих по мере переключения режимов его работы. Рассмотрим типовые многорежимные технологические процессы.

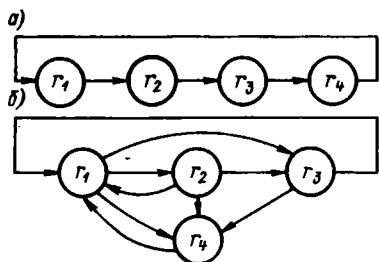


Рис. 1.7. Графы переключения режимов электроприводов: а — токарного станка; б — ГЭУ

Цикл работы электропривода токарного станка-автомата предусматривает следующую последовательность механических операций: установка инструмента (r_1); подача инструмента в рабочую зону (r_2); профильная обработка заготовки (r_3); вывод инструмента из рабочей зоны (r_4).

Граф-схема цикла приведена на рис. 1.7, а. Каждая операция (режим r_i) предъявляет свои требования к работе управляющей системы. Переключение режимов производится по жесткой схеме в соответствии с командами верхнего уровня управления и текущим состоянием электропривода (положением, скоростью), фиксируемым датчиками и концевыми переключателями. Граф-схема работы электропривода ГЭУ приведена на рис. 1.7, б, где выделены следующие режимы: разгон — торможение до нужной скорости (r_1); стабилизация скорости выходного вала (r_2); стабилизация мощности при разгоне и движении судна (r_3); стабилизация тока якоря и мощности при перегрузках (r_4).

Работа ГЭУ отличается отсутствием жесткой последовательности чередования режимов, что обусловлено влиянием внешних факторов и недетерминированным характером действия оператора при маневрах судна, приводящим к перегрузкам электродвигателей. Переключение режимов работы осуществляется при достижении определенных состояний (скорости, мощности, моментов сопротивления) привода и внешней среды с учетом командных сигналов с пульта.

Многорежимный характер работы технологического оборудования вызывает необходимость построения систем, которые удовлетворяли бы меняющимся от режима к режиму требованиям.

От того, насколько успешно выполняются все локальные требования, зависит качество управляемого процесса в целом. Поэтому системы фиксированной структуры, настраиваемые по некоторому обобщенному критерию, обычно не обеспечивают в каждом из режимов наилучшего качества управления и, следовательно, не обеспечивают заданных свойств технологического процесса.

Многорезимной системой будем называть САУ, поведение которой качественно изменяется с изменением частных целей управления (А. с. № 955475) [96]. Такая система обеспечивает повышение качества работы технологического оборудования за счет последовательного достижения локальных целей, формулируемых в соответствии с индивидуальными для каждого режима требованиями к процессу управления.

Многорезимную систему можно рассматривать как устройство (автомат) с конечным числом дискретных состояний (*режимов*) $r_1, r_2, \dots, r_s, \dots, r_N$, в каждом из которых к процессу управления предъявляются специфические требования. Переключение режимов осуществляется по текущей информации о состоянии объекта x и внешней среды μ , командам на переключение r_c , что аналитически можно выразить в виде *алгоритма переключения структуры*:

$$r = \rho_r(x, \mu, r_c), \quad (1.19)$$

где r — символьная переменная, принимающая значения из области (алфавита) $\Omega = \{r_1, r_2, \dots, r_s, \dots, r_N\}$; ρ_r — оператор квантования, который непрерывным величинам x, μ и дискретным сигналам r_c ставит в соответствие значения символа $r \in \Omega$.

Так как каждому автоматному состоянию САУ ставится в соответствие локальная цепь управления, то многорезимная система должна содержать группу многоконтурных *локальных регуляторов* P_s , каждый из которых реализует собственный алгоритм управления

$$P_s : u = u_s(x, x^*, \mu). \quad (1.20)$$

Локальный алгоритм, т. е. структура и параметры локального регулятора, выбираются в соответствии с требованиями, предъявляемыми к системе в текущем режиме. Таким образом, многорезимные системы представлены алгоритмом переключения структуры (1.19) и группой поочередно подключаемых регуляторов (1.20). Порядок их подключения в общем случае определяется не только командами оператора и другими внешними факторами, но и текущим состоянием управляемого объекта. Структурная схема многорезимной САУ приведена на рис. 1.8. Кроме локальных регуляторов P_s и объекта управления схема содержит коммутатор (K), осуществляющий переключение структуры при изменении режима работы r .

Необходимость переключения структуры САУ возникает не только при обслуживании многорезимных технологических объектов, но и при решении достаточно сложных локальных задач и, в частности, задачи терминального управления. Обычно выде-

ляются режимы: соответствующие большим и малым отклонениям от конечного состояния x ; регулирования промежуточных переменных при достижении ими предельных значений; движения по оптимальным многообразиям и т. д. (А. с. № 964938) [22, 23, 58, 60]. Тем самым каждому режиму ставится в соответствие некоторое множество X_i пространства состояния системы, что отвечает частной реализации алгоритма переключения (1.19). Использование группы локальных регуляторов, последовательно подключаемых по мере прохождения системой множеств X_i , позволяет обеспечить оптимальное функционирование системы без существенного усложнения алгоритма управления (см. гл. 4).

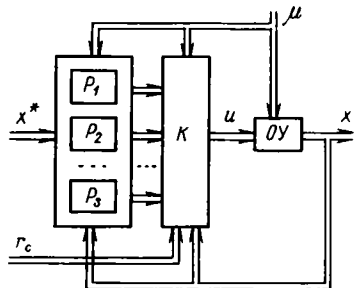


Рис. 1.8. Структурная схема многорежимной САУ

Системы управления, структура которых изменяется в зависимости от текущего состояния объекта, хорошо известны и относятся к *мультиструктурным САУ*, или *системам с переменной структурой* [31, 58, 74, 86].

Микропроцессорные (цифровые) системы управления. Рассматривая различные структурные решения САУ, мы не затрагивали вопросов их практической реализации. Структура системы отражает решение задачи управления и формально не зависит от используемых для ее построения технических средств. Однако выбор элементной базы в значительной мере определяет сложность схемных решений и даже саму возможность реализации разрабатываемой САУ.

В традиционных системах компоненты управляющего устройства (регуляторы, коммутаторы и т. д.) строятся на аналоговых схемах и коммутационных элементах. Такая элементная база ограничивает возможности унификации аппаратуры и препятствует реализации современных алгоритмов управления, характеризующихся значительной вычислительной сложностью. Кроме того, аналоговые схемы многорежимных систем аппаратурно избыточны, так как содержат элементы, лишь эпизодически участвующие в процессе управления (рис. 1.8).

Универсальную элементную базу современные системы управления получили с развитием цифровых средств обработки данных и особенно с появлением серийных микропроцессоров и микроЭВМ [70, 90]. Структура аппаратных средств микропроцессорной системы управления представлена на рис. 1.9. Система содержит микроЭВМ, объект управления (ОУ) и *устройство связи с объектом (УСО)*, которое обеспечивает преобразование цифровых кодов на выходе микроЭВМ в сигналы, воспринимаемые объектом, и преобразование выходных сигналов датчиков в двоичные коды, поступающие в машину.

Управляющая микроЭВМ с тактом (интервалом квантования) T на основании информации о текущем состоянии объекта x , а также задающих воздействий и командных сигналов Γ_c , поступающих с верхнего уровня, в соответствии с программой рассчитывает управляющее воздействие u . При построении микропроцессорной САУ следует учитывать следующие особенности программной реализации алгоритмов управления: временное и амплитудное квантование сигналов; запаздывание, вносимое микроЭВМ; возможность реализации логических, алгебраических и сводящихся к ним трансцендентных процедур; простое изменение структуры САУ с использованием условных переходов на другие ветви

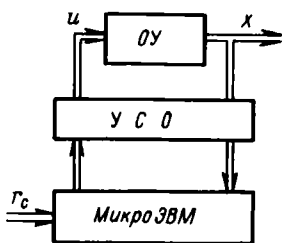


Рис. 1.9. Структурная схема системы управления с микроЭВМ

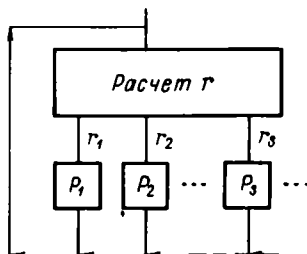


Рис. 1.10. Программная реализация переменной структуры

программы; возможность хранения в запоминающих устройствах больших информационных массивов программ, исходных и промежуточных данных и результатов.

Последовательный характер выполнения операций и связанное с ним запаздывание в обработке информации для реальных микроЭВМ с ограниченным быстродействием вносит определенные ограничения на сложность используемых алгоритмов управления. С другой стороны, микроЭВМ предоставляет широкие возможности для построения систем с переменной структурой ввиду простоты программной реализации алгоритмов переключения (1.19) и возможности использования в качестве локальных регуляторов (1.20) различных ветвей общей программы (рис. 1.10). Не вызывает дополнительных затруднений также вопрос хранения и воспроизведения программных законов управления, а при необходимости и их генерации.

Архитектура и алгоритмическое обеспечение САУ. Современная система управления строится как иерархическая система [28, 52, 77, 80], содержащая уровни управления (регулирования), оптимизации и организации управления (рис. 1.11). Для большинства многоканальных САУ характерна децентрализованная стратегия управления, предусматривающая, что управление каждым каналом (ОУ1, ОУ2 и т. д.) осуществляется собственной локальной САУ (САУ1, САУ2 и т. д.), а координация работы ка-

налов производится с *уровня организации управления*. Координация обычно выполняется путем согласования задающих воздействий g или x^* , поступающих на локальные САУ, а в более общих случаях с использованием специальных координирующих сигналов, учитывающих взаимодействие каналов [52]. Кроме этого, уровень организации вырабатывает команды r_c , инициирующие работу отдельных САУ и осуществляющие переключение режимов работы оборудования.

Уровень оптимизации локальных САУ осуществляет выбор наилучшего режима работы системы $r = r_s$, путем подключения

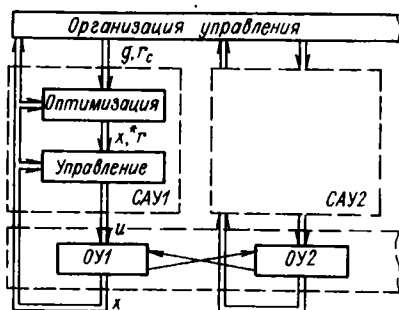


Рис. 1.11. Иерархическая структура САУ

нужного регулятора P_s , и формирование оптимальных программных воздействий x^* . Уровень управления представлен локальным многоконтурным регулятором (1.18) или группой регуляторов (1.20).

САУ реализуется на базе единой микроЭВМ, последовательно осуществляющей выработку управляющих воздействий на локальные объекты, или ЭВМ, рассредоточенными по объектам. Верхний уровень иерархии может быть представлен пультом оператора

или центральной ЭВМ, обеспечивающей координацию локальных САУ.

Основным функциональным элементом цифровой системы управления является *алгоритмический модуль*, т. е. обособленный формализованный фрагмент общего алгоритма формирования управляющего воздействия. Алгоритмическое обеспечение согласовано со структурой технических средств и содержит локальные регуляторы, задающие устройства, алгоритмы переключения структуры и т. д. Алгоритмические модули, объединенные процедурами обмена информацией, составляют основу алгоритмического обеспечения цифровых систем и используются при разработке прикладного программного обеспечения микроЭВМ.

Модульный принцип построения алгоритмов упрощает процедуру синтеза конкретной САУ, обеспечивает расширяемость ее функциональных возможностей, создает предпосылки унификации алгоритмического и прикладного программного обеспечения микропроцессорных систем управления для однородных объектов.

1.3. Технические средства и программное обеспечение микропроцессорных систем управления

Характеристики и эксплуатационные параметры систем управления определяются элементной базой и техническими средствами цифровых САУ, которые могут включать унифицированные комплекты изделия и специализированные устройства.

Унифицированными изделиями являются одноплатные микроЭВМ, допускающие расширение при создании встроенных регуляторов с требуемыми техническими параметрами, серийные микроЭВМ с набором средств сопряжения с объектом, средства ручного управления, хранения данных и управляющих программ. Выбор технических средств и проектирование специализированных устройств осуществляется с учетом требуемых характеристик цифровой системы управления, предполагаемой сложности алгоритмов обработки информации в процессе формирования управляющих воздействий. При этом рассматривается широкая номенклатура элементов, технические параметры которых позволяют прогнозировать технические характеристики цифровых САУ.

Элементная база также включает различные унифицированные схемы в интегральном исполнении, для использования которых в САУ необходимо выполнить схемотехническое проектирование, разработать и изготовить печатные платы, реализовать микропрограммное управление.

Элементная база цифровых систем управления. Элементная база цифровых систем управления опирается на интенсивно развивающуюся технологию производства интегральных схем (ИС). Схемы разделяются на типы [100], каждый из которых характеризуется совокупностью технических параметров (минимальной технологической компонентой ИС является эквивалент логической схемы 2И, называемый *вентилем*); МОП-технология имеет наилучшие показатели динамической рассеивающей мощности (мВт/МГц); n — МОП-технология характеризуется минимальными геометрическими размерами вентиля; ЭСЛ-схемы имеют минимальную задержку на вентиль (τ); p — МОП-технология — минимальную сложность изготовления; КМОП-технология отличается минимальной рассеивающей мощностью в статике (P).

Совместимость по питанию (+5 В) с широко распространенными сериями схем малой (около 10² вентилях в корпусе) и средней (до 10² вентилях в корпусе) степени интеграции является одним из важных требований к элементной базе, выбираемой для серийных цифровых систем управления. Совместимыми являются элементы КМОП-, ТТЛ-, ТТЛШ-, n -МОП- и И²Л-технологий.

Одним из наиболее важных достижений интегральной схемотехники и технологии является создание функционально-законченных схем большой (БИС — 10³ ... 10⁴ вентилях в корпусе) и сверхбольшой (СБИС — 10⁴ ... 10⁶ вентилях) степени интеграции.

Интегральные характеристики системы управления могут быть рассчитаны с учетом параметров ИС. При этом параметры одного вентиля для расчета мощности умножаются на коэффициенты 10⁴ ... 10⁶, что соответствует сложности цифровой системы управления, а для расчета минимального времени одноктного цифрового преобразования задержка вентиля умножается на коэффициент 10¹—10², что соответствует реальным задержкам сигналов в схеме.

С учетом интегральных характеристик наиболее перспективной является КМОП-технология, которая обеспечивает минимальную рассеиваемую статическую мощность при удовлетворительном значении критерия $R\tau$.

Интегральные схемы, используемые в цифровых системах управления, могут быть ориентированы на специализированное применение в некотором узком классе систем и унифицированное использование в широком классе изделий. Специализация обеспечивается заказными и полузаказными БИС.

В *заказных БИС* может быть выполнено любое цифровое устройство, как с уникальными техническими параметрами, так и массового применения. При этом для выбранной технологии обеспечиваются максимальные плотность компоновки и быстродействие, определяемые схемотехническими решениями, топологией соединений и размещением. Цикл проектирования и изготовления БИС занимает 1,5—2 года и включает трудоемкие и дорогостоящие этапы проектирования топологии масок, изготовление и тестирование опытных образцов, подготовку и оснастку производства и контроля. Технология заказных БИС, как правило, недоступна большинству разработчиков систем управления.

В *полузаказных БИС* также может быть реализована практически любая специальная схема с некоторыми ограничениями по параметрам. Ограничения определяются базовым кристаллом — регулярной вентиляющей матрицей, имеющей фиксированные размеры, ограниченное число выводов, заданную задержку на вентиль, ограничение на число вентилях для реализации схемы. *Матричные БИС* (МаБИС) позволяют разработать и изготовить специализированную схему с «программированием» соединений. При этом цикл проектирования и изготовления сокращается до нескольких месяцев. Используются два способа «программирования» соединений БИС — маскирование, т. е. разработка маски для нанесения соединений в МаБИС, и пережигание плавких перемычек (лишних соединений) в полной скоммутированной программируемой логической матрице (ПЛМ). Маскирование соединений в МаБИС дешевле, чем изготовление заказных БИС, однако также требует специальных производственных условий. При проектировании цифровых систем управления массового применения технология полузаказных БИС наиболее перспективна, доступна и позволяет получить высокие технические параметры — минимальные габаритные размеры и стоимость, высокое быстродействие, надежность и другие необходимые качества массовых встраиваемых регуляторов. Программирование ПЛМ практически доступно любому пользователю, так как существует соответствующая технология и стендовое оборудование для записи и контроля.

Очевидное достоинство унификации заключается в сокращении времени и трудоемкости проектирования и изготовления. Специализация к конкретным условиям применения обеспечивается

программированием, комплексированием другими унифицированными или заказными изделиями. Высокий уровень унификации и серийность переводит изделие в разряд массовых унифицированных изделий, которые предпочтительно в дальнейшем использовать в виде массовой заказной БИС. Такими изделиями являются однокристалльные микроЭВМ, микропроцессоры, элементы сопряжения с объектом, запоминающие устройства.

Одними из наиболее распространенных унифицированных элементов и устройств в цифровых системах управления являются *запоминающие элементы* и законченные *запоминающие устройства*, которые разделяются по типам на оперативные и постоянные [68]. *Оперативное запоминающее устройство* (ОЗУ или RAM) обеспечивает произвольный доступ к данным по адресу с возможностью чтения и записи в рабочем режиме. *Постоянное запоминающее устройство* (ПЗУ или ROM) допускает только чтение по адресу в рабочем режиме. Запись информации осуществляется однократно при изготовлении в специальном технологическом процессе на специальном оборудовании. По способу записи информации (по способу программирования) различают четыре класса постоянных запоминающих устройств.

1. *Маскируемые ПЗУ* (ПЗУ или ROM) программируются в процессе изготовления элемента с использованием технологических масок. Этот способ записи отличается высокой плотностью и надежностью, используется в массовых изделиях.

2. *Программируемые пользователем ПЗУ* (ППЗУ или PROM). Значение бита информации в ППЗУ определяется наличием или отсутствием плавкой перемычки, которая может быть удалена в специальном режиме, доступном пользователю.

3. *Перепрограммируемые ПЗУ с возможностью стирания* ультрафиолетовым облучением (СППЗУ или EPROM). Запись информации осуществляется электрически на специальном оборудовании.

4. *Перепрограммируемые ПЗУ с возможностью выборочной электрической перезаписи* (ЭППЗУ или EEPROM).

Технические средства цифровых систем управления. Структура технических средств цифровой системы управления приведена на рис. 1.12. Центральным элементом системы является микроЭВМ, которая осуществляет ввод данных о состоянии объекта, обработку данных в соответствии с алгоритмом управления и формирование управляющего воздействия в цифровой форме. Для выполнения этих функций и согласования с разнообразными требованиями к форме, мощности, виду сигналов на входе и выходе объекта управления цифровая система обеспечивается внешними устройствами и устройствами связи с объектом.

К *специальным внешним устройствам*, характерным для цифрового управления с использованием ЭВМ, относятся: 1) устройства отображения, которые индицируют состояние ЭВМ и объекта управления на индикаторах, дисплеях, печати; 2) пультовые

устройства, используемые для установки начальных параметров, ввода команд оператора в случае необходимости вмешательства или в процессе автономной отладки и контроля; 3) устройства дистанционной связи, обеспечивающие связь и взаимодействие с центральной ЭВМ, с удаленными терминалами и локальными элементами управления; физическая связь осуществляется по линиям связи (телефонным и оптическим линиям, с использованием радио- и телевизионных сигналов); 4) долговременная и энергонезависимая внешняя память цифровых САУ, позволяющая хранить информацию о ходе технологического процесса, а также прикладные программы САУ.

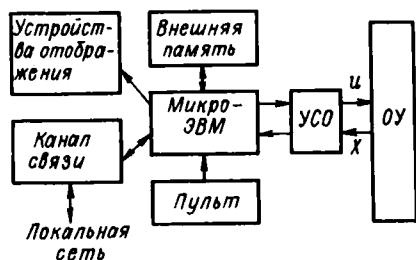


Рис. 1.12. Структура технических средств цифровой САУ

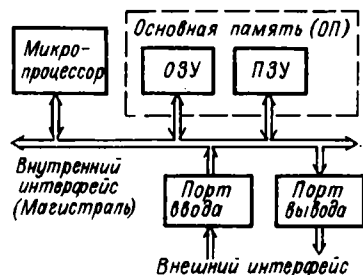


Рис. 1.13. Структура микроЭВМ

Устройства связи с объектом обеспечивают передачу данных в цифровой форме на входы или с выходов микроЭВМ. При этом может потребоваться временное согласование, преобразование из аналоговой формы в цифровую и обратное преобразование, электрическое согласование. Конструктивно и функционально устройства связи с объектом могут быть объединены с оборудованием микроЭВМ, включены в технологическое оборудование объекта управления и выделены в самостоятельное оборудование. При проектировании технических средств цифровых САУ широко используются микропроцессорные наборы унифицированных элементов БИС, из которых может быть скомплексирована схема, максимально удовлетворяющая требованиям, предъявляемым к ресурсам памяти, вводу—выводу, габаритным размерам и быстродействию.

Различают микропрограммируемые и программируемые микропроцессорные наборы. Структура микроЭВМ, выполненной на унифицированных элементах программируемого набора, приведена на рис. 1.13 и включает микропроцессор, основную память, схемы временного хранения данных при вводе (порт ввода) и схемы временного хранения данных при выводе (порт вывода). Связь между функциональными элементами обеспечивается внутренним интерфейсом (магистралью); связь с внешними устройствами осуществляется внешним интерфейсом.

Микропроцессор, выполненный на одной БИС, является основным элементом, осуществляющим арифметические операции над числами и формирующим сигналы управления другими элементами структуры в процессе выполнения программы.

Оперативное запоминающее устройство (ОЗУ) используется для хранения программ системы управления и оперативных данных, обрабатываемых этими программами. Различного типа ПЗУ используются для хранения программ и констант, которые должны сохраняться при отключении питания. Оперативная и постоянная память физически разделены, так как выполнены на различных элементах памяти и логически разделяются адресами. Совокупность ячеек ОЗУ и ПЗУ рассматривается как непрерывное адресное пространство и называется *основной (ОП) или внутренней памятью* микроЭВМ, в отличие от *внешней*, в которой накапливаются данные и хранятся программы.

Исполнение микроЭВМ в одном модуле БИС в виде *однокристалльного микрокомпьютера* является примером унификации массовых комплексных изделий вычислительной техники, переходящих в унифицированные элементы. Вместе с тем при таком исполнении заметно сказываются технологические и конструктивные ограничения, следствием которых являются существенные ограничения в ресурсах: ОЗУ предназначено только для хранения данных и имеет объем 64—512 слов, программа может храниться только в ПЗУ (СПЗУ) и занимать объем 1—2 Кслов. В табл. 1.1 приведены характеристики некоторых однокристалльных микрокомпьютеров [36].

Значительные возможности современной технологии БИС позволяют расширить функции однокристалльных микрокомпьютеров включением в него аналого-цифровых и цифроаналоговых преобразований, что позволяет создать простые цифровые регуляторы на одном кристалле. Одним из первых представителей этого класса элементов является *аналоговый микрокомпьютер K1813*, содержащий быстрый многоразрядный микропроцессор, аналоговый компаратор и цифроаналоговый 8-разрядный преобразователь, который может быть запрограммирован на вывод цифровой информации. Благодаря высокому быстродействию и многоразрядным арифметическим операциям такой однокристалльный цифровой регулятор превосходит по характеристикам аналогичный регулятор, выполненный на основе универсальной микроЭВМ. При разработке САУ широкого назначения, программируемых пользователем в процессе эксплуатации, требуются значительные ресурсы памяти, развитые программируемые средства ввода—вывода. Цифровые регуляторы конструктивно усложняются и выполняются в виде встраиваемой одноплатной или многоплатной системы на основе микропроцессорного набора или в виде унифицированных микроЭВМ. В одноплатной конфигурации [48] объем ПЗУ (СПЗУ) достигает 8—16 Кслов, объем ОЗУ — 32 Кслов, количество цифровых входов — выходов — несколько

Таблица 1.1

Характеристики однокристальных микроЭВМ

Тип микроЭВМ	Разрядность, бит	Статическая мощность рассеивания, Вт	Объем ОЗУ, бит	Объем ПЗУ, бит	Число цифровых входов — выходов	Число последовательных каналов	Тактовая частота, МГц	Внешняя память ОЗУ (ПЗУ), Кбайт	Число входов прерываний	Число таймеров
1820BE1	4	0,35	64×4	1К×8	16	1	1,6	—	1	
1816BE48	8	0,6	64×8	1К×8	24	—	6	0,5(3)	2	
C5-31	16	—	512×16	1К×16	24	1	2	64	2	1
1801BE1	16	—	128×16	1К×16	24	—	8	64	2	
1816BE51	8	—	128×8	4К×8	24	1	4	64	5	

Таблица 1.2

Характеристики одноплатных микроЭВМ

Тип микроЭВМ	Тип микропроцессора	Объем ОЗУ, бит	Объем ПЗУ, бит	Число цифровых входов — выходов	Число последовательных каналов	Габаритные размеры, мм	Статическая мощность, Вт
МС 11100.1	1801ВМ1	28К×16	4К×16	—	1	252×296	15
МС 2702	К580ИК80	1К×8	4÷16К×8	48	1	350×240	35
C5-21	Специальный	256×16	2К×16	32	4	309×244	20
МС 1201	1801ВМ1	64К×8	8÷16К×8	8	1	292×256	14
ПМ В02П	К1810ВА86	64К×8	16÷64К×8	24	4	245×390	40
ПМ В01П	К580ИК80	16К×8	4÷16К×8	48	4	245×390	25

десятков, имеется последовательный канал для связи с терминалом. Характеристики некоторых серийных одноплатных микроЭВМ приведены в табл. 1.2.

В многоплатной конфигурации объем ОЗУ может быть расширен до 128 Кслов и более, использованы сотни цифровых входов—выходов, многоканальные высокоточные ЦАП, АЦП, усилители и преобразователи. Многоплатные микроЭВМ комплектуются обычно одноплатными процессорами или одноплатными микроЭВМ, выполненными на микропроцессорных наборах.

Одноплатные микроЭВМ могут быть объединены в *мультисистему* на основе системного интерфейса — *системной маги*

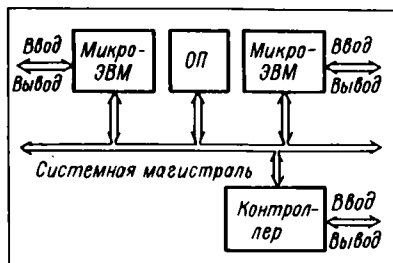


Рис. 1.14. Структура магистральной мультисистемы

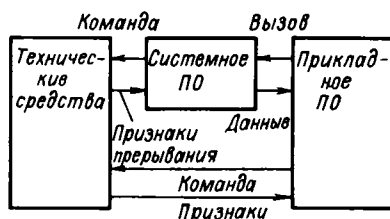


Рис. 1.15. Двухуровневая организация ПО

страли (рис. 1.14). Существуют несколько типов стандартных системных магистралей, к которым в соответствии с едиными требованиями подключаются: одноплатные микроЭВМ или процессоры на микропроцессорных наборах, системная память ОП и контроллеры ввода—вывода для управления системными внешними устройствами.

Внешние устройства цифровых САУ разнообразны по организации и назначению; в большинстве случаев являются стандартными комплектующими изделиями серийных микроЭВМ. С некоторыми из них можно познакомиться в [2].

Средства ручного управления в виде пультов и клавиатур в настоящее время широко применяются для оперативного управления, задания режимов, запуска и контроля. Пульты могут быть автономными и встроенными, иметь специальный и унифицированный набор клавиш, близкий по назначению клавиатуре пишущей машинки (клавиатура является стандартным устройством серийных терминалов всех типов ЭВМ).

Обзор существующих технических средств демонстрирует их разнообразие по номенклатуре и принципу действия, организации и системному использованию. Наиболее широко в цифровых САУ используются микроЭВМ с программным управлением, серийные и встроенные в объект управления и выполненные на основе программируемых микропроцессорных наборов. Программное обеспечение цифровых САУ является важным и самостоятельным объ-

ектом проектирования. При разработке сложных цифровых САУ естественно разделение труда по двум направлениям: 1) проектирование технических средств; 2) разработка программного обеспечения.

Программное обеспечение цифровых систем управления. Программирование является наиболее универсальным и доступным средством специализации цифровых систем. Программное обеспечение легко модифицируется и может быть полностью заменено без изменения аппаратуры и технических средств, что способствует унификации САУ. Различают два уровня в программном обеспечении: *системные* и *прикладные* программы (рис. 1.15).

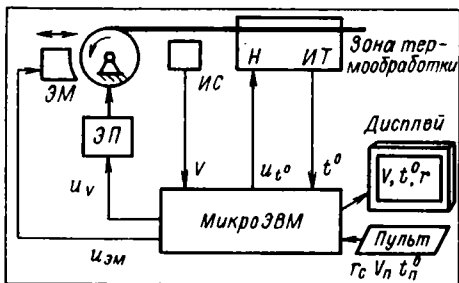


Рис. 1.16. Схема цифровой САУ технологической линии

Наиболее распространенная системная программа *монитор* управляет вводом—выводом и обеспечивает доступ прикладным программам к данным, реализует пультовые режимы управления — инициализацию системы, останов, пошаговый режим работы, контроль состояния памяти микроЭВМ и др. Связь монитора с техническими средствами цифровой

САУ обеспечивается через систему команд процессора, сигналами прерывания и признаками, формируемыми в аппаратуре. В прикладной программе при обращении к функциям монитора используются системные обращения (вызовы) и под управлением монитора из устройств ввода передаются данные для обработки.

Пример 1.1. Рассмотрим цифровую систему управления технологической линией (рис. 1.16). Система включает измерители скорости *ИС* движения ленты и температуры *ИТ*, электропривод *ЭП*, электромагнитную муфту *ЭМ*, нагреватель *Н*. Все эти элементы системы подключены к микроЭВМ через устройство сопряжения. Кроме того, в состав аппаратуры входят стандартные средства ввода—вывода или внешние устройства — дисплей для индикации температуры и скорости, пульт оператора, на котором устанавливаются режимы работы r_c .

Выделим режимы работы САУ:

r_1 — программное регулирование скорости движения и температуры ленты по сигналу с пульта $r_c = r_{c1}$;

r_2 — стабилизацию скорости и температуры по сигналу с пульта $r_c = r_{c2}$ и уставкам $V = V_y$, $t^o = t_y^o$;

r_3 — аварийное торможение по сигналу $V \cong 0$ или $r = r_{c3}$.

Программное обеспечение системы управления содержит алгоритм переключения структуры (*планирующую программу*) и программы локального регулирования $P_1 — P_3$, реализующие соответствующие режимы. Планирующая программа осуществляет анализ сигналов r_c , V и выбирает режим работы САУ, подключая регуляторы P_i , рассчитывающие управляющие воздействия на электропривод u_v , нагреватель u_t и ЭМ $u_{ЭМ}$. Структура прикладного программного обеспечения системы приведена на рис. 1.17.

Каждый режим работы характеризуется собственными диапазонами изменения скоростей и температур, а также интервалом дискретизации управления.

В режиме r_1 при скорости ленты 0 ... 0,1 м/с, выбираем интервал дискретизации 0,01 с для расчета u_v и 1 с для расчета u_t ; время аварийного торможения 0,1 с; перемотка осуществляется со скоростью 0—5 м/с при постоянной температуре, а управление с интервалом квантования 0,01 с.

Программы, время исполнения которых ограничено заданными временами дискретизации, вносят существенную задержку в формирование управляющих воздействий. Эта задержка может оказаться критической, особенно при переключении режимов. Так, переход из режима r_1 в режим r_3 потребует $(m + 1) 0,12$ с, где m — число последовательных интервалов дискретизации, отрабатываемых при исполнении программ в режиме r_1 . В то же время на торможение в режиме r_3 отводится только 0,1 с. Задержка, связанная с переключением режима, существенно сокращается, если программа режима r_3 включается с прерыванием текущего режима работы САУ. Электромагнитная муфта удерживается включенной в течение фиксированного интервала 0,1 с.

Ввод с пульта значений V_y, t_y^0 занимает 200—300 мс, и его следует совместить с формированием управления, так как в течение этого времени может существенно измениться состояние системы. При нажатии клавиши на пульте по сигналу прерывания вызывается программа ввода символа, после завершения ввода возобновляется формирование управления. Ввод одного символа занимает 10—20 мкс, за это время инерционный объект не претерпевает существенных изменений.

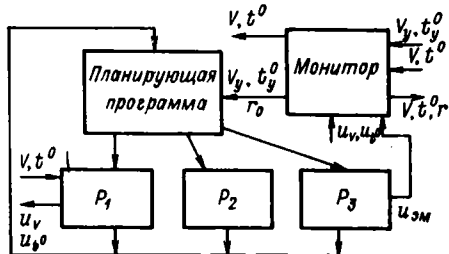


Рис. 1.17. Структура прикладного программного обеспечения

Цифровая система управления работает в реальном времени, реагируя на внешние события, связанные с процессом управления, в течение времени, не превышающего допустимого для каждого события. В рассматриваемой САУ события связаны с изменением режимов, вводом начальных параметров и директив с пульта. События в системе могут быть синхронизированы с другими событиями или происходить в случайные моменты времени. Каждое событие вызывает соответственно синхронизированный или случайный процесс в системе управления. К *синхронизированным процессам* относятся два последовательных процесса, из которых один вызывает другой. Запуск процессов может быть синхронизирован с окончанием других процессов по фиксированному кванту времени. Так, на торможение отводится 0,1 с, после чего может быть включена планирующая программа.

Случайными событиями являются команды r_c с пульта, сбоя питания или неисправности в механической части системы, вызывающие экстремальные условия, которые требуют принятия срочного решения. Случайные события происходят во время выполнения некоторого текущего процесса и вызывают его прерывание. Так, исчезновение сигнала скорости перемещения ленты является случайным событием, вызывающим прерывание процесса управления перемещением.

Отсчет интервалов времени может осуществляться параллельно с текущими процессами специальным таймером. Сигналы об окончании отсчета времени являются случайными событиями, вызывающими прерывание текущего процесса и запуск нового процесса.

События в системе реального времени могут конкурировать между собой. Для разрешения конфликтов в случае возникновения нескольких событий или при появлении события во время некоторого процесса устанавливаются приоритеты между событиями и связанными с ними процессами. Один из них имеет более высокий приоритет и инициирует наиболее важный в данный момент процесс. В рассматриваемой системе, например, сигнал, обозначающий исчезновение скорости при перемещении ленты, является более приоритетным, чем сигнал с пульта управления.

Сложная программа управления разделяется на несколько частных задач, каждая из которых связана с некоторым событием в системе. В данном случае можно выделить следующие задачи (рис. 1.18): 1) ввод начальных параметров с пульта (задача ожидает прерывания с пульта при нажатии клавиши; время ожидания не определено); 2) регулирование скорости в режиме r_1 (задача повторяется с интервалом дискретности $0,01$ с, включает ввод переменной t , формирование управления u_V и его вывод на электропривод); 3) регулирование температуры в режиме r_1 циклически повторяется ($T = 0,1$ с); 4) регулирование скорости в режиме r_2 ($T = 0,01$ с); 5) регулирование температуры в режиме r_2 ($T = 0,1$ с); 6) управление в режиме r_3 (запускается по прерыванию).

Кроме этих программ, используются системные программы инициализации и вывода. Программа инициализации включается по прерыванию сигналом СБРОС и при включении питания автоматически вызывает для исполнения за-

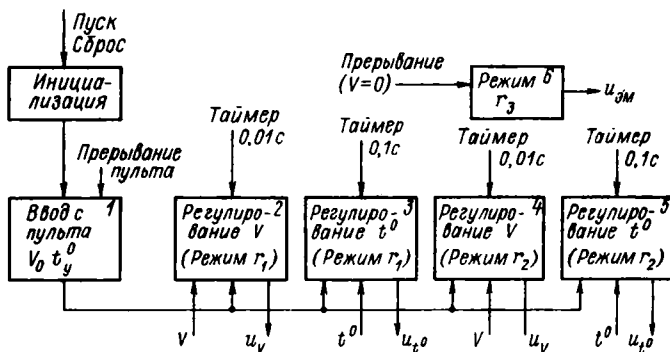


Рис. 1.18. Многозадачная структура ПО цифровой САУ в реальном времени

дачу ввода данных с пульта. Задачи ввода и вывода пассивны и всегда по вызову готовы к выполнению; задачи регулирования включаются по готовности начальных данных и по истечении интервала дискретности, фиксированного для каждой конкретной задачи. Таким образом, в системе определены несколько задач, которые по готовности начальных данных с учетом приоритета передаются на исполнение. Более приоритетные события могут вызывать прерывание менее приоритетных процессов управления. После окончания решения срочных задач возобновляется решение прерванной задачи.

Функции, связанные с организацией вычислительного процесса в системе, в частности, управление таймером, задачами, прерываниями, могут быть унифицированы в *операционной системе реального времени* (ОС РВ). Эта система регистрирует сигналы прерывания и выбирает наиболее приоритетное событие для запуска соответствующего процесса, связывает с каждой задачей интервал времени и контролирует его окончание по системным меткам времени, организует передачу данных между задачами и управление очередями задач.

Различают *собственное* (встроенное или резидентное) программное обеспечение (ПО) и *внешнее*, загружаемое из внешней памяти. Собственное ПО хранится в энергонезависимой памяти, в частности, в ПЗУ и после включения питания сразу запускается на исполнение. В собственное ПО входят наиболее ответственные

программы — мониторы, библиотеки подпрограмм, операционные системы, а также прикладные программы в специализированных цифровых САУ и локальных регуляторах.

Внешнее ПО загружается с перекрытием в ОЗУ так, что следующая исполняемая программа занимает ячейки памяти, которые занимала предыдущая программа. ПО реализует алгоритмы управления и определяет в значительной степени качество и технические параметры цифровой САУ, в частности, такие ее параметры, как задержка формирования управляющего воздействия, объем и структуру памяти.

Сокращение сроков и повышение надежности проектирования цифровых САУ тесно связано с автоматизацией наиболее трудоемких этапов.

Автоматизация программирования и отладка цифровых систем управления поддерживается *инструментальными системами*, которые разделяют на два типа: прототипные и кросс-системы. *Прототипная* система включает ЭВМ, система команд которой совпадает с системой команд микропроцессора, используемого в цифровой САУ. *Кросс-системы* разрабатываются на ЭВМ общего назначения ЕС ЭВМ, СМ ЭВМ (обычно более доступных благодаря распространению вычислительных центров коллективного пользования).

С использованием программных моделей объекта управления на кросс-системах может быть организована разработка и отладка прикладных программ цифровой САУ параллельно с разработкой схем и макета САУ. Создание эквивалентных программных моделей объекта управления является самостоятельной проблемой, связанной с дискретным представлением непрерывных процессов.

Достаточную точность при отладке прикладных программ, реализующих алгоритмы управления, обеспечивают математические модели на аналоговых вычислительных машинах. *Цифроаналоговые комплексы* включают аналоговую вычислительную машину, которая моделирует поведение объекта, и прототипную инструментальную систему со схемным эмулятором и макетом цифровой САУ (рис. 1.19). Цифроаналоговый комплекс позволяет выполнить разработку и отладку прикладных программ в реальном времени с точностью, обеспечиваемой математической моделью объекта управления. Для имитации стартстопных режимов и прерываний могут быть использованы программные имитаторы,

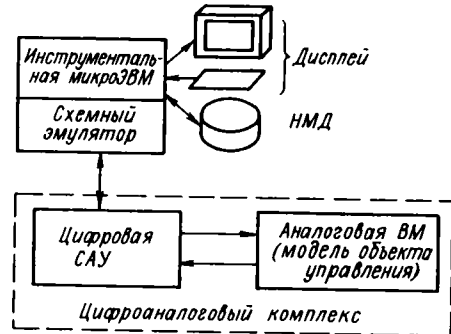


Рис. 1.19. Структура цифроаналогового комплекса проектирования и отладки САУ

которые срабатывают с программируемыми временными задержками в составе многозадачной операционной системы реального времени.

1.4. Методы синтеза цифровых алгоритмов управления

Основное содержание процесса управления в микропроцессорных САУ заключается в пошаговом формировании управляющих воздействий в соответствии с заранее полученными аналитическими выражениями, т. е. алгоритмами. Кроме уже упомянутых алгоритмов управления, описывающих работу локальных регуляторов и переключения структуры, к алгоритмам САУ относятся алгоритмы фильтрации и наблюдения, алгоритмы формирования программных воздействий и т. д. Их совокупность образует алгоритмическое обеспечение системы управления и является основой для построения цифровых САУ и в первую очередь прикладного программного обеспечения.

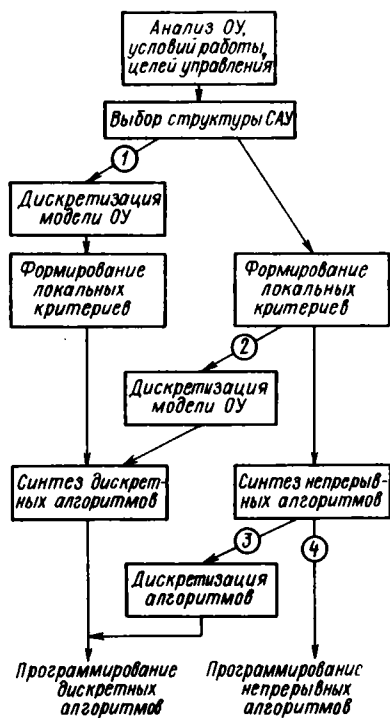


Рис. 1.20. Процедура синтеза цифровой САУ

Синтез алгоритмов цифровых систем. Процедура синтеза приведена на рис. 1.20 и включает этапы анализа задачи управления, выбора структуры САУ и синтеза алгоритмов. На этапе анализа строятся математические модели объекта управления и внешней среды, формулируются цели управления в виде формализованных критериев качества. Выбор структуры предусматривает определение основных функциональных элементов системы (уровней иерархии, алгоритмических модулей) и общего характера их взаимодействия. Структура САУ зависит от класса решаемых системой задач, требований к процессу управления, полноте априорной и текущей информации о переменных состояния, внешних

воздействиях и параметрах объекта. На последнем этапе синтеза осуществляется разработка алгоритмов управления, фильтрации, переключения структуры, решается задача оптимизации системы по локальным критериям качества. В тривиальных случаях оптимальное поведение САУ удастся обеспечить путем соответствующего подбора параметров алгоритмов управления. Для слож-

ных задач управления оптимальное качество поддерживается всей структурой системы, оптимизацией задающих воздействий, а при необходимости и адаптивной настройкой параметров локальных регуляторов. При проектировании САУ этап синтеза алгоритмов управления, наблюдения и формирования задающих воздействий является наиболее ответственным и трудоемким. Его конечным результатом являются аналитические выражения, представленные в удобной для программирования форме. На рис. 1.20 приведен ряд альтернативных процедур (ветви 1—4), отличающихся в первую очередь использованием методов синтеза дискретных, т. е. заданных в рекуррентной форме, или непрерывных алгоритмов.

Методы синтеза непрерывных алгоритмов наиболее отработаны, однако их применение в цифровых системах требует предварительной дискретизации, осуществляемой при синтезе САУ (ветвь 3) или выполняемой с помощью программных средств микроЭВМ (ветвь 4). В последнем случае возможно программирование непрерывных алгоритмов, но требуется развитие программного обеспечения САУ и, в частности, проблемно-ориентированных языков.

Наилучшее приближение к заданным характеристикам цифровой САУ обеспечивается при дискретизации модели объекта управления и применении методов синтеза дискретных систем, позволяющих получить алгоритмы в рекуррентной форме (ветви 1, 2).

Частотные и временные методы синтеза САУ. В теории управления выделяют две группы методов синтеза систем: 1) основанные на частотном описании процессов управления; 2) временные.

В основе частотных методов синтеза лежит операторная форма записи, полученная с использованием преобразований Лапласа, Фурье, Z-преобразования [77]. Связь входных и выходных переменных непрерывного объекта управления устанавливается, например, передаточной функцией $W(s)$:

$$Y(s) = W(s) U(s), \quad (1.21)$$

где $U(s)$, $Y(s)$ — преобразования Лапласа управления u и регулируемой переменной y соответственно; s — комплексная переменная.

Типовой алгоритм управления описывается выражением

$$U(s) = K(s) E(s) + L(s) G(s), \quad (1.22)$$

где $E(s)$ — преобразование Лапласа ошибки e , причем $E(s) = G(s) - Y(s)$; $G(s)$ — преобразование Лапласа задающего воздействия g ; $K(s)$, $L(s)$ — передаточные функции обратной и прямой связи.

Выбор передаточных функций алгоритма управления позволяет обеспечить заданные частотные свойства системы, в результате чего достигается нужная точность обработки задающих воздействий, устойчивость замкнутой системы и динамика переходных

процессов. При реализации САУ на аналоговых элементах передаточные функции заменяются соответствующими интегро-дифференциальными операторами. Цифровая реализация предусматривает преобразование интегро-дифференциальных операторов в рекуррентную форму. Так, для рекуррентного описания алгоритма (1.17) может быть использовано приближенное выражение, определяющее значения u в дискретные моменты времени $t = mT$, $m = 0, 1, 2, \dots$ [28]:

$$u(mT) = k_1 e(mT) + k_2 \sum_{l=1}^m e((l-1)T) + \frac{k_3}{T} [e(mT) - e((m-1)T)],$$

где T — интервал дискретизации.

Рассматривая это же выражение в моменты времени $t = (m-1)T$ и произведя необходимые преобразования, получим

$$u(mT) = u((m-1)T) + k_{1d} e(mT) + k_{2d} e((m-1)T) + k_{3d} e((m-2)T), \quad (1.23)$$

где $k_{1d} = k_1 + k_3/T$; $k_{2d} = -k_1 - 2k_3/T - Tk_2$; $k_{3d} = k_3$.

Алгоритм производит расчет значения u с использованием текущей и прошлой информации об ошибке, а также информации о значении управления на предыдущем шаге. Отметим, что погрешность квантования, вызванная неточностью дискретизации, зависит от величины интервала T и при больших его значениях делает алгоритм (1.23) неработоспособным.

Типовая схема системы, содержащей непрерывный объект управления и ЭВМ, по известным правилам [28, 77] может быть приведена к дискретной операторной форме:

$$Y_d(z) = H(z) U_d(z),$$

где $U_d(z)$, $Y_d(z)$ — Z -преобразования управления u и выходной переменной y ; $H(z)$ — дискретная передаточная функция; $z = \exp(sT)$.

В этом случае синтез регулятора сводится к выбору дискретных передаточных функций $K_d(z)$ и $L_d(z)$, входящих в состав алгоритма управления

$$U_d(z) = K_d(z) E_d(z) + L_d(z) G_d(z), \quad (1.24)$$

где $E_d(z)$ — Z -преобразование ошибки e , причем $E_d(z) = G_d(z) - Y_d(z)$; $G_d(z)$ — Z -преобразование задающего воздействия g .

Полученное выражение может быть использовано непосредственно для построения дискретного алгоритма, если рассмотреть z как оператор сдвига.

Так, алгоритм

$$U_d(z) = \frac{k_{1d} + k_{2d}z^{-1} + k_{3d}z^{-2}}{1 - z^{-1}} E_d(z)$$

соответствует рекуррентному алгоритму (1.23).

Наиболее разработанные частотные методы ориентированы на построение линейных одноконтурных систем, что значительно ограничивает их применение.

С развитием теории оптимального управления и широким внедрением машинных методов проектирования получили распространение временные методы синтеза САУ и, в частности, *метод пространства состояний* [4, 24, 83]. Метод использует дифференциальные векторно-матричные формы описания динамических процессов (1.7) и не накладывает ограничений на число входных и выходных переменных. Введение в рассмотрение вектора состояния позволяет наиболее полно учесть внутренние процессы системы, скрытые при описании (1.21). Типовая форма локального алгоритма управления, обеспечивающего, например, отработку программного воздействия x^* , имеет вид

$$u = V + Ke, \quad (1.25)$$

где $V = V(x^*, \mu)$ — вектор-функция, учитывающая прямые связи по задающему воздействию x^* и возмущению μ ; K — матрица обратных связей по ошибке; $e = x^* - x$ — ошибка, или рассогласование.

В частном случае, функция V вычисляется как

$$V = L_1 x^* + L_2 \mu, \quad (1.26)$$

где L_1 и L_2 — матрицы прямых связей.

Задача синтеза алгоритма управления (1.25), (1.26) сводится к поиску коэффициентов прямых и обратных связей, т. е. матриц K , L_1 , L_2 , удовлетворяющих заданным качественным показателям системы в рассматриваемом режиме: времени переходного процесса, перерегулированию, точности и т. д. В зависимости от выбранного подхода (рис. 1.20) при синтезе используется непрерывная модель (1.7) или ее дискретный аналог:

$$x((m+1)T) = f_d(x(mT), u(mT), \mu(mT)), \quad (1.27)$$

причем для линейного объекта управления может быть найдена дискретная модель, абсолютно точно описывающая поведение системы в моменты $t = mT$.

При использовании алгоритмов, синтезируемых на основе непрерывной модели, поведение цифровой системы будет отличаться от расчетного. Это объясняется методической погрешностью в вычислении управляющего воздействия, обусловленной не учтенными при проектировании факторами: квантованием сигналов, запаздыванием, вносимым ЭВМ в процесс управления. Необходимость уменьшения интервала дискретности, т. е. времени рабочего цикла расчета управления, вынуждает разработчика предъявлять чрезмерные требования к быстродействию используемых вычислительных средств и приводит к увеличению затрат на проектируемое оборудование.

Методы синтеза, основанные на применении дискретных моделей, обеспечивают построение алгоритмов, лишенных методической ошибки. Однако для нелинейных систем характерны трудности получения дискретных моделей (1.27). Для этих целей используются приближенные методы, например метод Эйлера, что естественно приводит к ошибкам дискретизации модели объекта.

Алгоритмы управления (1.25), (1.26) относятся к простейшим статическим алгоритмам, так как не содержат динамических операторов или рекуррентных процедур. Этим объясняется совпадение их структуры при синтезе непрерывными и дискретными методами. В более общем случае непрерывное описание алгоритма содержит дифференциальные уравнения, появление которых связано с астатическим или адаптивным регулированием, необходимостью восстановления неизвестного вектора состояния (фильтрацией или наблюдением) и обуславливает потребность в дискретизации непрерывных алгоритмов на последнем этапе синтеза.

Оптимизация систем управления. Требования повышения точности и качества работы оборудования определяют необходимость оптимизации САУ; методы и приемы оптимизации управления в реальном масштабе времени трудно реализуемы, поэтому системы управления оптимизируются, как правило, на этапе синтеза.

Оптимальное функционирование САУ обеспечивается следующими мерами: выбором оптимальной структуры САУ и алгоритма переключения структуры; подготовкой оптимальных программных воздействий или оптимальной структуры задающего устройства; оптимизацией параметров локальных регуляторов. Отметим, что оптимизация параметров в процессе функционирования реализуется в адаптивных системах [28, 74, 80, 99, 101].

Процедура оптимального синтеза САУ в общем случае направлена на разработку структуры и алгоритмов, наилучшим образом отвечающих требованиям к точности, колебательности и быстродействию переходных процессов; экономичности управления. Эти требования формализуются в виде задачи минимизации интегральных функционалов качества

$$J = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt, \quad (1.28)$$

где t_f — время протекания процесса; $f_0(\cdot)$ — скалярная функция.

При решении линейных задач оптимальной стабилизации и программного управления наибольшее распространение получили квадратичные функционалы

$$J = \int_0^{\infty} (\mathbf{e}^T \mathbf{Q} \mathbf{e} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt, \quad (1.29)$$

где $u = Ke$ — сигнал обратной связи; R, Q — весовые матрицы. Выбор весовых матриц устанавливает разумный компромисс между стремлением наискорейшим образом ликвидировать ошибку системы e , колебательностью переходных процессов и ограничением энергетических потерь, определенных интегралом от $u^T Ru$.

Решение задачи минимизации функционала (1.29) позволяет получить оптимальные значения параметров матрицы обратной связи K локальных алгоритмов управления (1.25). Функционал (1.29) применяется при построении алгоритмов на базе непрерывной модели объекта управления. Для дискретной модели он приобретает вид:

$$J = \sum_{m=0}^{\infty} [e^T(mT) Q e(mT) + u^T(mT) R u(mT)]. \quad (1.30)$$

При использовании более общих типов функционала (1.28) решение оптимальной задачи в замкнутом виде получить обычно не удастся. В этом случае процедура синтеза предусматривает численный расчет оптимальных переходных процессов

$$x^* = x^*(t), \quad (1.31)$$

обеспечивающих минимизацию функционала (1.28).

После этого расчета решается задача программного управления путем минимизации локальных критериев (1.29), где $e = x^* - x$ (рис. 1.20, ветви 3, 4). Для дискретной процедуры синтеза (ветвь 1) можно получить дискретный аналог функционала (1.28) и найти оптимальные последовательности

$$x^* = x^*(mT).$$

В нелинейных задачах, когда процедура дискретизации объекта управления и функционала (1.28) вызывает затруднения, переход к дискретному синтезу осуществляется после нахождения оптимального переходного процесса (1.31) (рис. 1.20, ветвь 2). Для малых значений ошибки программного движения e задача управления легко линеаризуется и приводится к дискретному виду.

Задача терминального управления обычно является нелинейной. Среди возможных критериев оптимальности выделим критерий оптимального быстрогодействия ($J = t_f$), минимальных затрат энергии или топлива (см. п. 4.1), которые применяются для сравнения ансамбля переходных процессов, проходящих через заданные точки x_0 и x_f (рис. 1.21). Задача может быть сведена к задаче программного управления по рассмотренной выше схеме. Для систем с фиксированными значениями x_0 и x_f такое решение наиболее рационально. При значительном разбросе начальных и конечных значений требуется расчет, хранение и воспроизведение в реальном масштабе времени большого числа программных законов, что приводит к неоправданному увеличению рабочего

объема памяти ЭВМ или усложнению задающего устройства. Эффективное решение в этом случае дает многорежимная организация САУ (см. рис. 1.8, 1.10), в основе которой лежит возможность разбиения задачи терминального управления на ряд *локальных задач* регулирования (*режимов*), последовательно решаемых в процессе функционирования САУ [58, 60]. Анализ траекторий типовых терминальных задач показывает, что в пространстве состояния можно выделить *области знакопостоянства управляющих воздействий* и *оптимальные многообразия* S_s (поверхности, линии, рис. 1.21), к которым относятся поверхности предельно допустимых значений переменных состояния, например:

$$\text{тока якоря } i_{я} = i_{я}(\mathbf{x})$$

$$i_{я}(\mathbf{x}) \pm I_{\max} = 0;$$

$$\text{мощности исполнительных механизмов } P = P(\mathbf{x})$$

$$P(\mathbf{x}) - P_{\max} = 0;$$

многообразия оптимального переключения и т. д.

Для описания оптимальных многообразий в общем случае используются неявные уравнения

$$S_s: \Phi_s(\mathbf{x}) = 0, \quad s = \overline{1, N}, \quad (1.32)$$

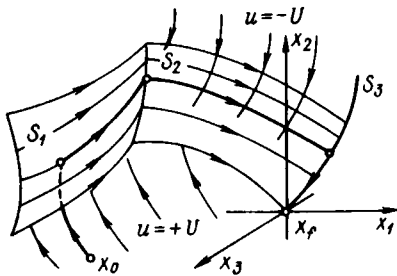


Рис. 1.21. Фазовые траектории и многообразия терминальной задачи управления

задаваемые ограничениями на переменные состояния (1.9)—(1.14) или полученные в результате объединения гладких участков фазовых траекторий, соответствующих ансамблю переходных процессов (1.31).

Прохождение в определенной последовательности областей знакопостоянства и оптимальных многообразий (рис. 1.21) обеспечивает оптимальное качество системы в целом. Локальные задачи управления движением в указанных множествах решаются группой регуляторов P_s , подключаемых алгоритмом переключения структуры. Эти регуляторы осуществляют стабилизацию движения относительно гладких участков оптимальных многообразий (1.32), отклонение от которых характеризуется вектором ошибки $\mathbf{e} = \Phi(\mathbf{x})$.

Таким образом, задача управления в каждом режиме работы САУ сводится к задаче оптимальной стабилизации по критерию (1.29), который в данном случае является локальным критерием общей терминальной задачи. В задачах стабилизации широко применяются модели объекта, линеаризованные в области малых ошибок \mathbf{e} и, следовательно, легко преобразуемые в дискретной форме, поэтому синтез локального регулятора рационально производить методами дискретной теории (см. рис. 1.20, ветвь 2) с использованием функционала (1.30).

Вопросы синтеза цифровых систем управления на основе методов пространства состояния и теории оптимизации подробнее рассматриваются в последующих разделах. При этом выбор одной из альтернативных процедур синтеза производится в зависимости от многих трудноформализуемых факторов, к которым относятся динамические особенности управляемого процесса и характер решаемой задачи управления, уровень развития программных средств отладки программ реального времени, а также квалификация и опыт специалистов, участвующих в разработке системы.

Автоматизация проектирования САУ. Проектирование цифровой системы управления представляет собой многошаговый процесс выполнения предписаний технического задания [75, 77]. Необходимость повторения отдельных этапов и пересмотра вариантов решения обусловлена как отсутствием достаточно полной информации об объекте управления, так и естественным для любого уровня развития несовершенством используемых методов проектирования и технических средств реализации.

В настоящее время трудно говорить о полной автоматизации всей процедуры проектирование, что объясняется большим числом творческих и, следовательно, плохо алгоритмизируемых этапов, а также необходимостью привлечения на разных стадиях разработки специалистов различного профиля. Поэтому обычно под *системой автоматизированного проектирования САУ* подразумевается наиболее рациональное перераспределение функций между проектировщиками и техническими средствами — аппаратурой и программным обеспечением используемых ЭВМ [77].

Проектирование цифровой системы включает следующие этапы (рис. 1.22): разработку алгоритмического обеспечения САУ; выбор технических средств реализации; разработку программного обеспечения, а также моделирование; настройку и анализ промежуточных результатов. Современные методы синтеза алгоритмов в значительной степени формализованы (см. гл. 3, 4), что создает необходимые предпосылки для полной автоматизации этого этапа. После разработки алгоритмов становится возможным математическое моделирование САУ, позволяющее оценить качество про-

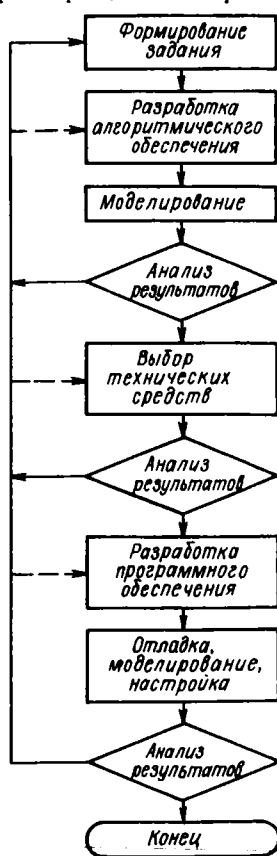


Рис. 1.22. Процедура проектирования системы управления с микроЭВМ

цессов управления без учета погрешностей реализации и осуществить настройку некоторых заранее обусловленных параметров. Автоматизация программирования и отладка прикладного программного обеспечения поддерживаются инструментальными системами, которые рассмотрены в гл. 6. Моделирование и настройка САУ с участием разработанных технических средств осуществляются на последних этапах проектирования.

На основании анализа результатов моделирования и оценок технико-экономических показателей необходимого оборудования принимаются решения о продолжении процедуры синтеза, уточнении модели объекта или возвращении на один из предыдущих этапов проектирования, а в крайнем случае — об изменении требований технически нереализуемого задания на разработку.

Автоматизация проектирования алгоритмических, программных и аппаратных средств цифровых систем требует привлечения модульных принципов проектирования. Эти принципы предусматривают использование единого ряда *унифицированных элементов*, соответствующая настройка и компоновка которых обеспечивает потребности широкого класса задач и объектов управления. Унификация цифровых САУ позволяет: упростить процедуру проектирования, повысить эффективность использования средств вычислительной техники, упростить настройку и отладку цифровых систем, облегчить стыковку локальных САУ между собой и взаимодействие с верхним иерархическим уровнем, обеспечить возможность переналадки и расширения систем при модернизации обслуживаемого оборудования и внедрении более совершенных аппаратных средств.

Глава 2

МОДЕЛИ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

2.1. Модели объектов управления

Основными исходными данными при проектировании автоматических систем служат математические модели объектов управления. Адекватность таких моделей реальным физическим процессам в значительной степени определяет общее качество функционирования разрабатываемой системы.

Линейные стационарные модели. Для одноканальных объектов (или ОУ с одним входом — одним выходом) наиболее распространенными типами моделей являются:

передаточная функция

$$W(s) = \frac{Y(s)}{U(s)} = \frac{b_1 s^{n-1} + \dots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} \quad (2.1)$$

(a_i, b_i — постоянные коэффициенты; $i = \overline{1, n}$; n — порядок модели; s — комплексное число);

дифференциальное уравнение вход—выход, которое для объекта с передаточной функцией (2.1) имеет вид

$$\begin{aligned} \frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_{n-1} \frac{dy}{dt} + a_n y &= \\ &= b_1 \frac{d^{n-1} u}{dt^{n-1}} + \dots + b_{n-1} \frac{du}{dt} + b_n u; \end{aligned} \quad (2.2)$$

векторное дифференциальное уравнение состояния

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u; \\ \mathbf{y} = \mathbf{C}\mathbf{x}, \end{cases} \quad (2.3)$$

где матрицы \mathbf{A} , \mathbf{B} , \mathbf{C} для объекта с передаточной функцией (2.1) равны [4]

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & \dots & -a_1 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{bmatrix};$$
$$\mathbf{C} = [b_n, b_{n-1}, \dots, b_1].$$

В выражениях (2.1)—(2.3) $y = y(t)$ — выходная переменная; $u = u(t)$ — управляющее воздействие; Y_s и $U(s)$ — изображения Лапласа этих величин; $x = x(t)$ — вектор состояния размерности n ,

$$x = \left[y \frac{dy}{dt} \dots \frac{d^{n-1}y}{dt^{n-1}} \right]^T,$$

т. е. вектор пространства состояния R^n .

В более общем случае многоканального объекта управления также можно воспользоваться моделью (2.3), где y — вектор выхода ОУ размерности r ($y \in R^r$); u — вектор управления размерности h ($u \in R^h$). Тогда модель вход—выход может быть задана передаточной матрицей размерности $r \times h$

$$W(s) = C(sI - A)^{-1}B,$$

где I — единичная матрица $n \times n$.

Характеристическое уравнение модели линейного объекта записывается так:

$$\det(sI - A) = 0.$$

Значения $s = s_i$, при которых левая часть уравнения обращается в нуль, называются корнями характеристического уравнения, полюсами объекта управления или собственными значениями (числами) матрицы A : $\lambda_i [A]$, $i = \overline{1, n}$.

Пример 2.1. Уравнения движения гироскопа направления имеют вид [78]

$$\begin{cases} I\ddot{\alpha} + H\dot{\beta} = M_\eta - h\dot{\alpha}; \\ I\ddot{\beta} - H\dot{\alpha} = -M_\zeta - h\dot{\beta}. \end{cases} \quad (2.4)$$

где I — экваториальный момент инерции гироскопа; H — кинетический момент; h — коэффициент демпфирования; α — угол азимута; β — угол места (склонения); M_η и M_ζ — внешние моменты относительно осей, перпендикулярных главной оси гироскопа.

Введем переменные состояния

$$x_1 = \alpha; \quad x_2 = \beta; \quad x_3 = \dot{\alpha}; \quad x_4 = \dot{\beta}.$$

тогда система уравнений (2.4) запишется в виде (2.3), причем

$$\left\{ \begin{array}{l} A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -h/I & -H/I \\ 0 & 0 & H/I & -h/I \end{bmatrix}; \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}; \\ y = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \quad u = \begin{bmatrix} M_\eta/I \\ -M_\zeta/I \end{bmatrix}; \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \end{array} \right. \quad (2.5)$$

Повышение степени адекватности модели объекта управления протекающим в нем процессам увеличивает точность результатов

проектирования систем управления. Однако с увеличением сложности используемых моделей резко возрастает трудоемкость проектирования системы, в частности, трудоемкость синтеза алгоритмов цифровых регуляторов. Один из возможных путей разрешения этого противоречия заключается в использовании нескольких типов моделей объекта в процессе проектирования.

На этапе анализа спроектированной автоматической системы, а также на этапе параметрического синтеза регулятора (по существу, на этапе настройки регулятора) целесообразно использовать наиболее точные модели объекта управления и внешних воздействий; на этапе синтеза алгоритмов цифровых регуляторов — наиболее простые модели, которые отражают только существенные стороны процессов, протекающих в объекте управления.

Редуцированные модели. Для линейного объекта упрощение модели эквивалентно понижению порядка исходной системы (2.3). Свойства модели пониженного порядка (редуцированной модели) определяются ее назначением; другими словами, для одной и той же исходной модели (2.3) можно получить ряд редуцированных моделей различного применения. Так как модель пониженного порядка предназначена для синтеза цифрового регулятора, то при редуцировании (2.3) исходят из желаемых динамических свойств проектируемой системы.

Пусть необходимо синтезировать регулятор, обеспечивающий в замкнутой системе время затухания процессов не более t_n и колебательность процессов μ_n . Переходный процесс в системе будет обладать заданными t_n и μ_n в том случае, когда корни характеристического уравнения замкнутой системы лежат внутри усеченного сектора P на комплексной плоскости, т. е. в области, ограниченной линиями L_1 и L_2 (рис. 2.1), где $L_1: s = -3t_n^{-1}$; $L_2: s = -\theta(1 \pm j\mu_n)$, $\theta \geq 0$ — параметр. Поэтому в процессе синтеза регулятора необходимо таким образом изменить свойства объекта управления, чтобы все собственные значения $\lambda_i [A]$, не принадлежащие области P , переместились в эту область. Собственные значения матрицы A , принадлежащие области P , в процессе синтеза можно оставить неизменными.

Обозначим через P_1 подобласть P , лежащую левее линии L_3 , уравнение которой

$$L_3: s = -3at_n^{-1}, \quad (2.6)$$

где $a > 1$ — параметр.

Множество собственных значений $\lambda_i [A]$, принадлежащих P_1 , может быть игнорировано при синтезе регулятора. Отметим, что область P_1 аналогична высокочастотной области (области малого параметра) в теории частотных методов синтеза регуляторов.

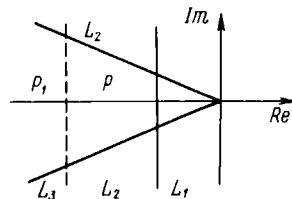


Рис. 2.1. Области расположения корней непрерывной системы

В соответствии с опытом, накопленным в теории частотных методов синтеза, будем выбирать параметр a в (2.6) равным 10. Понизим теперь порядок модели (2.3) таким образом, чтобы исключить собственные числа матрицы A , принадлежащие области P_1 . Для этого в уравнении (2.3) выполним замену переменных

$$\mathbf{x}_s = M\mathbf{x}. \quad (2.7)$$

Здесь M — матрица неособого линейного преобразования; в результате получим

$$\dot{\mathbf{x}}_s = \Lambda \mathbf{x}_s + \mathbf{B}_s \mathbf{u}, \quad (2.8)$$

где

$$\Lambda = \begin{bmatrix} \Lambda_1 & \vdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \vdots & \Lambda_2 \end{bmatrix}; \quad \mathbf{B}_s = \begin{bmatrix} \mathbf{B}_{s1} \\ \vdots \\ \mathbf{B}_{s2} \end{bmatrix}, \quad (2.9)$$

причем Λ имеет каноническую форму Жордана, а \mathbf{B}_s выбирается таким образом, чтобы пара (Λ, \mathbf{B}_s) была полностью управляема.

Подматрица Λ_2 в (2.9) имеет k собственных значений в области P_1 , а $n-k$ собственных значений подматрицы Λ_1 совпадают с оставшимися собственными значениями матрицы A . Преобразование M , обеспечивающее выбранное изменение формы представления матриц модели ОУ, согласно [4], имеет вид

$$M = [\mathbf{B}_s; \Lambda \mathbf{B}_s; \dots; \Lambda^{n-1} \mathbf{B}_s] [\mathbf{B}; \Lambda \mathbf{B}; \dots; \Lambda^{n-1} \mathbf{B}]^{-1}.$$

Преобразованные матрицы модели вычисляются по формулам:

$$\Lambda = MAM^{-1}; \quad \mathbf{B}_s = M\mathbf{B}.$$

Перепишем (2.8) в виде системы двух уравнений

$$\begin{cases} \dot{\mathbf{x}}_{s1} = \Lambda_1 \mathbf{x}_{s1} + \mathbf{B}_{s1} \mathbf{u}; \\ \dot{\mathbf{x}}_{s2} = \Lambda_2 \mathbf{x}_{s2} + \mathbf{B}_{s2} \mathbf{u}. \end{cases} \quad (2.10)$$

Будем считать, что a в (2.6) выбрано таким образом, что $\Lambda_2 \parallel \gg \Lambda_1 \parallel$, тогда система (2.10) относится к классу сингулярно возмущенных [66] и для ее исследования допускается переход к вырожденной системе

$$\begin{cases} \dot{\mathbf{x}}_{s1} = \Lambda_1 \mathbf{x}_{s1} + \mathbf{B}_{s1} \mathbf{u}; \\ \mathbf{0} = \mathbf{x}_{s2} + \Lambda_2^{-1} \mathbf{B}_{s2} \mathbf{u}. \end{cases} \quad (2.11)$$

Из второго уравнения системы (2.11) находим

$$\mathbf{x}_{s2} = -\Lambda_2^{-1} \mathbf{B}_{s2} \mathbf{u}. \quad (2.12)$$

Представим теперь (2.7) в виде

$$\begin{bmatrix} \mathbf{x}_{s1} \\ \mathbf{x}_{s2} \end{bmatrix} = \begin{bmatrix} M_{11} & \vdots & M_{12} \\ \vdots & \ddots & \vdots \\ M_{21} & \vdots & M_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_2 \end{bmatrix}, \quad (2.13)$$

где размерности \mathbf{x}_1 и \mathbf{x}_{s1} совпадают.

Из (2.13) находим

$$\mathbf{x}_2 = \mathbf{M}_{22}^{-1} \mathbf{x}_{s2} - \mathbf{M}_{22}^{-1} \mathbf{M}_{21} \mathbf{x}_1;$$

учитывая (2.12), имеем

$$\dot{\mathbf{x}}_2 = -\mathbf{M}_{22}^{-1} \mathbf{M}_{21} \dot{\mathbf{x}}_1 - \mathbf{M}_{22}^{-1} \Lambda_2^{-1} \mathbf{B}_{s2} \mathbf{u}. \quad (2.14)$$

Используя блочное представление для (2.3), найдем

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{11} \\ \mathbf{B}_{22} \end{bmatrix} \mathbf{u},$$

откуда с учетом (2.14) получим уравнение вырожденной системы в исходном базисе, описывающее «медленные» движения объекта,

$$\dot{\mathbf{x}}_1 = (\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{M}_{22}^{-1} \mathbf{M}_{21}) \mathbf{x}_1 + (\mathbf{B}_{11} - \mathbf{A}_{12} \mathbf{M}_{22}^{-1} \Lambda_2^{-1} \mathbf{B}_{s2}) \mathbf{u},$$

при этом координаты «быстрого» движения связаны с медленным движением объекта алгебраическим уравнением (2.14).

Перепишем уравнение «медленных» движений, представляющее модель объекта пониженной размерности, в виде

$$\dot{\mathbf{x}}_1 = \mathbf{A}_1 \mathbf{x}_1 + \mathbf{B}_1 \mathbf{u}, \quad (2.15)$$

где $\mathbf{A}_1 = \mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{M}_{22}^{-1} \mathbf{M}_{21}$; $\mathbf{B}_1 = \mathbf{B}_{11} - \mathbf{A}_{12} \mathbf{M}_{22}^{-1} \Lambda_2^{-1} \mathbf{B}_{s2}$.

Матрица \mathbf{A}_1 в (2.15) подобна матрице Λ_1 в (2.10); $\mathbf{T} \mathbf{A}_1 = \Lambda_1 \mathbf{T}$, причем матрица подобия $\mathbf{T} = \mathbf{M}_{11} - \mathbf{M}_{12} \mathbf{M}_{22}^{-1} \mathbf{M}_{21}$, а \mathbf{M}_{11} , \mathbf{M}_{12} , \mathbf{M}_{21} , \mathbf{M}_{22} определены в (2.13).

Редуцированную модель (2.15) целесообразно использовать вместо исходной модели (2.3) для синтеза регулятора, обеспечивающего динамические свойства системы, определяемые из рис. 2.1.

Пример 2.2. Рассмотрим понижение порядка модели (2.3) гироскопа направления (2.4), для которого коэффициент демпфирования h подобран таким образом, что нутационными движениями гироскопа на этапе синтеза можно пренебречь. Собственные значения матрицы \mathbf{A} в (2.5) будут $\lambda_1[\mathbf{A}] = 0$, $\lambda_2[\mathbf{A}] = 0$, $\lambda_{3,4}[\mathbf{A}] = -h/I \pm jH/I$. Согласно предположению собственные значения $\lambda_{3,4}[\mathbf{A}]$ попадают в область P_1 и их можно не учитывать при синтезе регулятора. —

Следя изложенной процедуре понижения порядка модели, зададим

$$\Lambda = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -a & -b \\ 0 & 0 & b & -a \end{bmatrix}; \quad \mathbf{B}_s = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ \hline 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad (2.16)$$

где $a = h/I$, $b = H/I$, тогда получим

$$\mathbf{M} = \begin{bmatrix} a-b & a+b & \vdots & 1 & 1 \\ a+b & -a+b & \vdots & 1 & -1 \\ \hline 0 & 0 & \vdots & 1 & 1 \\ 0 & 0 & \vdots & -1 & 1 \end{bmatrix}. \quad (2.17)$$

Из (2.5), (2.16) и (2.17) находим

$$A_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; \quad A_{12} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad B_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix};$$

$$M_{21} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; \quad M_{22} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix},$$

поэтому

$$A_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; \quad B_1 = \frac{1}{a^2 + b^2} \begin{bmatrix} a & -b \\ b & a \end{bmatrix}.$$

Используя выражения для a и b , запишем редуцированное уравнение гироскопа в виде

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \frac{1}{h^2 + H^2} \begin{bmatrix} h & -H \\ H & h \end{bmatrix} \begin{bmatrix} M_\eta \\ -M_\zeta \end{bmatrix}.$$

Модели нелинейных объектов. Модель состояния для нелинейного объекта в общем случае имеет вид

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (2.18)$$

где \mathbf{x} — вектор состояния объекта; \mathbf{u} — вектор управления; $\mathbf{f}(\cdot)$ — гладкая по \mathbf{x} и \mathbf{u} нелинейная вектор-функция.

Получение более простых моделей нелинейных объектов для целей синтеза регуляторов осуществляется обычно в два этапа.

На первом этапе осуществляется аппроксимация функции, стоящей в правой части уравнения (2.18), более простой функцией:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{f}_1(\mathbf{x}, \mathbf{u}) + \mathbf{O}(\mathbf{x}, \mathbf{u}),$$

где $\mathbf{f}_1(\cdot)$ — аппроксимирующая вектор-функция, а функция $\mathbf{O}(\mathbf{x}, \mathbf{u})$ определяет точность аппроксимации.

При малых значениях компонент вектора $\mathbf{O}(\mathbf{x}, \mathbf{u})$ $\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{f}_1(\mathbf{x}, \mathbf{u})$.

Широко распространенным и наиболее простым способом аппроксимации гладких функциональных зависимостей является линеаризация, когда

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{O}(\mathbf{x}, \mathbf{u}),$$

где $\mathbf{A} = \partial \mathbf{f} / \partial \mathbf{x}$, $\mathbf{B} = \partial \mathbf{f} / \partial \mathbf{u}$.

Частные производные вычисляются на решениях $\mathbf{x} = \mathbf{x}^*(t)$ и $\mathbf{u} = \mathbf{u}^*(t)$, определяющих программное движение объекта, отнositельно которого производится линеаризация.

Пример 2.3. Динамика электропривода (1.1) — (1.5) в задачах стабилизации угловой скорости гребной электрической установки, для которой $T_3 \cong \cong 0$ и $M = M_0 \omega |\omega|$, описывается уравнением (2.18), где

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}; \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}; \quad \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} a_{11}x_2x_3 + a_{12}x_1x_3^2 + a_{13}x_1|x_1| \\ a_2x_2 + b_1u_1 \\ a_3x_3 + b_2u_2 \end{bmatrix};$$

x_1 — угловая скорость двигателя; x_2 — выходное напряжение усилителя; x_3 — ток возбуждения; u_1 и u_2 — входные сигналы цепей тока якоря и возбуждения

(все величины даны в относительных единицах). Линеаризируя $f(x, u)$ в окрестности рабочей точки $x = x^* = [x_1^* x_2^* x_3^*]^T$, найдем

$$A = \frac{\partial f}{\partial x} \Big|_{x=x^*} = \begin{bmatrix} a_{12}(x_3^*)^2 + 2a_{13}x_1^* & a_{11}x_2^* & a_{11}x_2^* + 2a_{12}x_1^*x_3^* \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{bmatrix};$$

$$B = \frac{\partial f}{\partial u} = \begin{bmatrix} 0 & 0 \\ b_1 & 0 \\ 0 & b_2 \end{bmatrix}.$$

В случае, когда линеаризация приводит к недопустимым ошибкам, аппроксимирующая функция выбирается в виде линейной комбинации задаваемых проектировщиком базисных вектор-функций $f_k(x, u)$, т. е.

$$f(x, u) = \sum_{k=1}^{k_1} c_k f_k(x, u) + O(x, u),$$

причем матричные коэффициенты c_k обычно находятся методом наименьших квадратов (см. п. 4.4).

В широко распространенном случае, когда зависимость компонент $f(x, u)$ от координат u_j вектора управления u близка к линейной, т. е. $\partial f / \partial u_1 = \text{const}$, используется следующая форма аппроксимации:

$$f(x, u) = f_1(x) + G(x)u,$$

где $f_1(x) = f(x, 0)$; $G(x) = \partial f / \partial u |_{u=0}$ (модели этого типа рассматриваются в гл. 4).

На втором этапе получения упрощенной модели нелинейного объекта осуществляется понижение порядка аппроксимированной модели

$$\dot{x}_1 = f_1(x, u). \quad (2.19)$$

Для линейной функции $f_1(x, u)$ процедура понижения порядка модели описана ранее. В случае нелинейной функции $f_1(x, u)$ осуществляется выделение «медленных» x_1 и «быстрых» x_2 координат вектора x , причем скорости координат этих векторов сопоставляются с требуемой скоростью процессов проектируемой замкнутой системы. Если выделение таких составляющих возможно, то уравнение (2.19) представляется в виде:

$$\dot{x}_1 = f_3(x_1, x_2, u);$$

$$\varepsilon_2 \dot{x}_2 = f_4(x_1, x_2, u),$$

где ε_2 — малый параметр, характеризующий разницу скоростей «медленных» и «быстрых» движений объекта.

В случае большой разницы этих скоростей ε_2 мало, тогда

$$0 = f_4(x_1, x_2, u).$$

Разрешая это уравнение относительно x_2 , имеем $x_2 = f_5(x_1, u)$. После подстановки полученного выражения в первое уравнение системы получим уравнение модели пониженной размерности:

$$\dot{x}_1 = f_3(x_1, f_5(x_1, u), u).$$

2.2. Линейные модели внешних воздействий

Значительное влияние на поведение реальных объектов управления оказывает окружающая среда, т. е. неуправляемые внешние факторы, к которым относятся возмущающие воздействия $\mu(t)$ и задающие сигналы $g(t)$ или $x^*(t)$ (см. п. 1.1). Так как величина внешних воздействий, а иногда и характер их изменения, остаются неизвестными вплоть до начала работы САУ, то традиционные методы теории управления предусматривают изучение поведения системы для типовых законов изменения переменных μ и g : кусочно-постоянного, линейно возрастающего и гармонического [77].

Улучшение качества проектируемых систем управления достигается за счет более полного учета характера внешних воздействий на этапе синтеза систем управления. Одним из эффективных методов расширения класса допустимых внешних воздействий при исследовании систем управления является метод пространства состояний. С этой целью в рассмотрение вводятся динамические модели воздействий, называемые командными генераторами (КГ) [10, 102]. Такие модели с определенной степенью точности отражают динамику реальных физических процессов, действующих на объект управления извне. Командные генераторы могут входить в состав систем программного управления в качестве динамических задающих устройств, предназначенных для воспроизведения эталонных сигналов $g(t)$ и $x^*(t)$ (см. гл. 4). В общем случае модели внешних воздействий в явном виде в состав САУ не входят, с чем связаны в частности дополнительные трудности измерения состояния командного генератора.

Детерминированные модели. Детерминированные внешние воздействия μ и g представляются как выходные векторы размерности q и r ($\mu \in R^q$, $g \in R^r$) автономного командного генератора вида

$$\dot{\xi} = \Gamma \xi, \quad (2.20)$$

т. е.

$$\mu = \mathbf{H}_\mu \xi; \quad (2.21)$$

$$g = \mathbf{H}_g \xi, \quad (2.22)$$

где ξ — вектор состояния КГ размерности l ($\xi \in R^l$); Γ , \mathbf{H}_μ , \mathbf{H}_g — матрицы постоянных коэффициентов.

Изменение начальных условий $\xi(0)$ модели (2.20) обеспечивает генерирование различных реализаций внешних воздействий μ и g .

Пример 2.4. Для судового электродвигателя (см. п. 1.1) момент сопротивления при движении судна на волнении с постоянной скоростью имеет две составляющие — постоянную, зависящую от скорости судна, и переменную, которая для многих практических приложений может считаться синусоидальной,

$$\mu_c = \mu_1 + \mu_2 = \mu_1 + \mu_m \sin \omega_k t, \quad (2.23)$$

где ω_k — средняя частота качки судна.

Составляющая μ_1 представляется как решение дифференциального уравнения

$$\dot{\mu}_1 = 0, \quad (2.24)$$

а составляющая μ_2 — как решение уравнения

$$\ddot{\mu}_2 + \omega_k^2 \mu_2 = 0. \quad (2.25)$$

Введем промежуточные переменные $\xi_1 = \mu_1$, $\xi_2 = \mu_2$, $\xi_3 = \dot{\mu}_2$ и представим уравнения (2.24), (2.25) в нормальной форме Коши

$$\begin{aligned} \dot{\xi}_1 &= 0, \\ \dot{\xi}_2 &= \xi_3, \\ \dot{\xi}_3 &= -\omega_k^2 \xi_2 \end{aligned}$$

или в векторной форме (2.20), причем

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix}; \quad \Gamma = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega_k^2 & 0 \end{bmatrix};$$

$$\mu_c = \xi_1 + \xi_2 = H\xi; \quad H = [110].$$

Решение уравнения командного генератора при различных начальных условиях $\xi_1(0)$, $\xi_2(0)$, $\xi_3(0)$ дает различные реализации момента сопротивления μ_c судового электродвигателя.

Приведенный пример иллюстрирует общий подход к построению командного генератора в том случае, когда функция, описывающая внешнее воздействие, может быть легко представлена в виде решения линейного дифференциального уравнения. Во многих случаях внешние воздействия задаются аналитическими выражениями, не являющимися решениями линейных уравнений, или реализациями (графиками, таблицами), для которых значения матриц Γ , H_g , H_μ сразу установить не удастся. Тогда для описания динамики командного генератора воспользуемся аппроксимирующим разложением искомой функции, например,

$$\mu(t) = \sum_{k=1}^{k_1} c_k \mu_k(t). \quad (2.26)$$

Здесь c_k — матричные коэффициенты аппроксимации; $\mu_k(t)$ — базисные вектор-функции, являющиеся решениями известных линейных дифференциальных уравнений, т. е. выходными переменными локальных командных генераторов

$$\dot{\xi}_k = \Gamma_k \xi_k; \quad (2.27)$$

$$\mu_k = H_k \xi_k, \quad (2.28)$$

Здесь ξ_h — вектор состояния локального КГ; Γ_h, H_h — матрицы постоянных коэффициентов. Функции $\mu_h(t)$ находятся таким образом, чтобы при заданном числе членов разложения (2.26) получить минимальную норму вектора ошибки

$$\Delta(t) = \mu(t) - \sum_{k=1}^{k_1} c_k \mu_k(t) \quad (2.29)$$

или при заданной норме ошибки получить минимальное число членов разложения. В качестве нормы вектора принимается квадратичная норма, для которой матричные коэффициенты в разложении (2.24) находятся методом наименьших квадратов.

Из (2.26)—(2.28) нетрудно получить описание командного генератора в форме (2.20), (2.21). Для этого достаточно ввести обозначения:

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dots \\ \xi_{k_1} \end{bmatrix}; \quad \Gamma = \begin{bmatrix} \Gamma_1 & 0 & \dots & 0 \\ 0 & \Gamma_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Gamma_{k_1} \end{bmatrix}; \quad H_\mu = [c_1 H_1 \dots c_{k_1} H_{k_1}]. \quad (2.29a)$$

Заметим, что знание коэффициентов c_k для составления уравнений КГ не всегда обязательно. Так, в случаях, когда $\mu(t)$ — скалярная функция, а c_k — скалярные коэффициенты, выражения (2.26)—(2.28) можно записать в виде:

$$\begin{aligned} \mu(t) &= \sum_{k=1}^{k_1} \mu'_k(t); \\ \xi'_k &= \Gamma_k \xi'_k; \\ \mu'_k &= H_k \xi'_k, \end{aligned}$$

где $\mu'_k = c_k \mu_k$; $\xi'_k = c_k \xi_k$.

Переходя к описанию (2.20), (2.21), обозначим

$$\xi = \begin{bmatrix} \xi'_1 \\ \xi'_2 \\ \dots \\ \xi'_{k_1} \end{bmatrix}; \quad H_\mu = [H_1 \ H_2 \ \dots \ H_{k_1}], \quad (2.30)$$

оставляя матрицу Γ в форме, описанной в (2.29a).

Стохастические модели. Случайные внешние воздействия, в отличие от детерминированных, представляются как выход динамической системы, возбуждаемой белым шумом [47]. Уравнения командного генератора возмущающих воздействий $\mu(t)$ имеет вид:

$$\dot{\xi}(t) = \Gamma \xi(t) + G \zeta(t); \quad (2.31)$$

$$\mu(t) = H_\mu \xi(t), \quad (2.32)$$

где $\xi(t)$ — вектор состояния динамической системы; $\zeta(t)$ — белый шум (обычно нормально распределенный); Γ , \mathbf{G} и \mathbf{H} — матрицы, определяемые свойствами процесса $\mu(t)$.

Матрицы дифференциальных уравнений (2.31), (2.32) легко могут быть получены, если известна, например, спектральная плотность $S_\mu(\omega)$ процесса $\mu(t)$, которая определяется выражением

$$S_\mu(\omega) = |W_\mu(j\omega)|^2 S_\zeta(\omega), \quad (2.33)$$

где $W_\mu(j\omega)$ — амплитудно-фазовая частотная характеристика системы; $S_\zeta(\omega)$ — спектральная плотность входного сигнала.

Пусть $S_\mu(\omega)$ аппроксимирована дробно-рациональной функцией, допускающей факторизацию

$$S_\mu(\omega) = S(j\omega) S(-j\omega),$$

причем $S(s)$ — дробно-рациональная функция с нулями и полюсами в левой полуплоскости комплексной плоскости.

Учитывая также, что для белого шума интенсивности N спектральная плотность $S_\zeta(\omega) = N$, из (2.33) получим

$$W_\mu(j\omega) = S(j\omega)/\sqrt{N}$$

и, следовательно, искомая передаточная функция

$$W_\mu(s) = S(s)/\sqrt{N}. \quad (2.34)$$

Представим $W_\mu(s)$ в виде

$$W_\mu(s) = \frac{b_1 s^{n-1} + \dots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n},$$

тогда компонентами командного генератора (2.31), (2.32) будут [4]:

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dots \\ \xi_n \end{bmatrix}; \quad \Gamma = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & \dots & -a_1 \end{bmatrix}; \quad \mathbf{G} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{bmatrix};$$

$$\mathbf{H}_\mu = [b_n \dots b_1].$$

Пример 2.5. Пусть спектральная плотность внешнего воздействия $\mu(t)$ аппроксимирована выражением

$$S_\mu(\omega) = k \frac{\omega^2 + b^2}{\omega^4 + (a_1^2 + a_2^2)\omega^2 + a_1^2 a_2^2},$$

где $k = 1 \text{ с}^{-1}$; $b = 2 \text{ с}^{-1}$; $a_1 = 1 \text{ с}^{-1}$; $a_2 = 3 \text{ с}^{-1}$.

Разложив числитель и знаменатель этого выражения на произведения комплексно-сопряженных сомножителей, найдем

$$S(j\omega) = \sqrt{k} \frac{j\omega + b}{(j\omega + a_1)(j\omega + a_2)}.$$

По формуле (2.34) определим

$$W_{\mu}(s) = \sqrt{\frac{k}{N}} \frac{s+b}{(s+a_1)(s+a_2)} = \frac{\sqrt{k/N} s + \sqrt{k/N} b}{s^2 + (a_1 + a_2)s + a_1 a_2}.$$

Для заданных параметров $S_{\mu}(\omega)$ и шума единичной интенсивности $N = 1$ с, найдем значения матриц уравнений командного генератора (2.31), (2.32):

$$\Gamma = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix}; \quad G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad H = [2 \ 1].$$

Представления внешних воздействий в виде решения уравнения (2.20)—(2.22) или (2.31), (2.32) охватывают практически все случаи, встречающиеся в инженерной практике при проектировании и исследовании систем управления с вычислительными машинами в контуре управления.

2.3. Выбор интервала квантования

В рамках дискретных моделей, которые описывают поведение цифровых систем управления с непрерывными объектами только в дискретные моменты времени, можно выбирать любой интервал квантования и в результате синтеза управления обеспечить наперед заданные свойства линейной системы в дискретные моменты времени. Однако внутри интервала квантования управляющее воздействие остается постоянным и характер процессов внутри этого интервала (см. рис. 2.2) во многом определяется свойствами объекта управления и внешних воздействий. При слишком больших интервалах квантования могут проявляться не демпфируемые колебательные составляющие собственного движения ОУ и высокочастотные возмущающие воздействия. Поэтому с точки зрения обеспечения плавности протекания управляемых процессов внутри интервала целесообразно уменьшать значение T . Однако, чем меньше интервал квантования при управлении одним и тем же

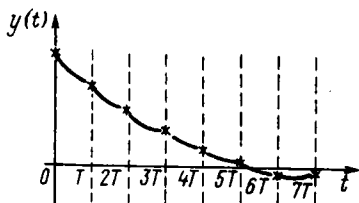


Рис. 2.2. Переходный процесс в дискретной системе

объектом, тем выше нагрузка управляющей ЭВМ, т. е. уменьшение интервала квантования приводит к снижению эффективности использования ЭВМ. Поэтому с точки зрения повышения эффективности использования ЭВМ надо стремиться к увеличению интервала квантования, имея в виду, что требования к качеству процессов управления являются главенствующими, т. е. ни одно из требований к свойствам проектируемой системы не должно быть нарушено при выборе максимально возможного интервала квантования.

В линейной дискретной системе с одномерным входом можно синтезировать такой закон управления, что переходный процесс закончится в течение минимального числа интервалов дискрет-

ности [4, 39]. Это минимально возможное число интервалов равно порядку системы n (при подсчете порядка системы учитывается порядок динамического регулятора). На основании этого свойства для обеспечения заданного времени переходного процесса t_n необходимо выбирать интервал квантования T из условия

$$T < t_n/n. \quad (2.35)$$

В многомерных системах интервал квантования может выбираться большим, чем это следует из (2.35).

Условие (2.35) ограничивает сверху значение T в силу принципиальных свойств дискретных систем и никак не связано с поведением процесса $y(t)$ дискретной системы внутри интервала квантования. Значительные колебания внутри интервала, как уже отмечалось, возможны в том случае, когда свободное движение объекта управления имеет колебательные составляющие, т. е. когда среди корней характеристического уравнения объекта управления имеются комплексно-сопряженные. Эти корни, лежащие вне области P (см. рис. 2.1), определяют сильно колебательные составляющие свободного движения объекта, не удовлетворяющие заданному качеству процессов в проектируемой системе, поэтому свойства этих составляющих должны быть скорректированы цифровым регулятором. Для управления гармонической составляющей процесса в соответствии с теоремой Котельникова требуются хотя бы два дискретных управляющих воздействия на период T_r этой гармонической составляющей, поэтому интервал квантования управления должен удовлетворять неравенству

$$T < 0,5T_r. \quad (2.36)$$

Выберем среди возможных корней характеристического уравнения объекта, не принадлежащих области P , такую пару, которая имеет максимальный коэффициент мнимой части ω_0 , тогда для демпфирования этой составляющей в соответствии с (2.36) требуется, чтобы выполнялось условие

$$T < \pi/\omega_0; \quad (2.37)$$

при этом в дискретной системе будут демпфироваться все колебательные составляющие свободного движения с более низкой частотой.

Выбор интервала квантования должен учитывать также требования, предъявляемые к системе в установившемся режиме, когда вынужденное движение воспроизводит входное воздействие g с ошибкой, не превышающей заданную. Внешние воздействия обычно описываются разложением по базисным функциям, как это показано в п. 2.2. Для гармонических базисных функций интервал квантования в соответствии с (2.36) выбирается согласно условию

$$T < \pi/\omega_v, \quad (2.38)$$

где ω_b — частота верхней гармоники, при учете которой обеспечивается требуемая точность описания g .

В том случае, когда в качестве базисных функций для g взяты полиномы времени, рекомендуется выбирать интервал квантования из условия [79]

$$T < \sqrt{8\varepsilon/g_2}, \quad (2.39)$$

где ε — заданная ошибка слежения; g_2 — максимальное ускорение входного воздействия g .

При компенсации возмущающего воздействия μ интервал квантования также должен удовлетворять условию (2.38). Частота ω_b в этом случае определяется на основании вычисления вынужденного движения для неуправляемого объекта по дифференциальным уравнениям объекта и командного генератора возмущения. В качестве ω_b выбирается частота той наиболее высокочастотной составляющей, которая возбуждает составляющую регулируемой величины объекта с амплитудой, превышающей заданную ошибку.

Объединяя (2.35), (2.37)—(2.39), получим условие для выбора интервала квантования

$$T < \min(t_{\text{п}}/n, \pi/\omega_0, \pi/\omega_b, \sqrt{8\varepsilon/g_2}). \quad (2.40)$$

Рекомендации по выбору интервала квантования, отличные от изложенных, можно найти в [41, 83].

2.4. Разностные модели линеаризованных объектов управления

В гл. 1 отмечалось, что наиболее эффективное использование управляющих микроЭВМ обеспечивается в том случае, когда уже на этапе синтеза алгоритмов управления учитывается квантование процессов в цифровой системе во времени. Учет этого квантования, а также учет запаздывания, вносимого в контур управления микроЭВМ, осуществляется переходом от непрерывной модели ОУ к дискретной в соответствии с временной диаграммой обмена информацией между объектом и управляющей ЭВМ. Временная диаграмма изображена на рис. 2.3, где отмечены моменты ... mT , $mT + T$, ... начала такта работы ЭВМ в системе управления. В качестве этих моментов удобно выбрать начало опроса измерительных устройств, установленных на ОУ. В течение времени τ_1 происходит опрос измерителей, преобразование аналоговых сигналов в двоичный код и запись этих кодов в соответствующие ячейки оперативной памяти. В дальнейшем везде будем считать, что в течение времени τ_1 показания измерителей существенно не меняются, т. е. отсчеты всех сигналов, выполненные в течение времени τ_1 , можно отнести к моменту времени mT .

В течение времени τ_2 выполняется алгоритм расчета управляющего воздействия, а в течение времени τ_3 рассчитанное управляющее воздействие выводится на ЦАП, преобразуется в аналого-

вую форму и поступает на ОУ. Это рассчитанное управляющее воздействие запоминается на выходе ЦАП до выдачи из ЭВМ нового рассчитанного управляющего воздействия.

Как следует из диаграммы на рис. 2.3, в управляющей ЭВМ имеет место запаздывание $\tau = \tau_1 + \tau_2 + \tau_3$ между началом опроса измерителей и выдачей управляющего воздействия. До выдачи этого воздействия к ОУ приложено управление, рассчитанное на предыдущем такте. Запаздывание ЭВМ τ может быть равно интервалу квантования T и может быть меньше этого интервала. Если $\tau < T$, то в промежуток времени $(mT + \tau, mT + T)$ внутри каждого такта ЭВМ решает задачи, которые являются фоновыми по отношению к задаче управления.

Модели автономных объектов управления. На основании временной диаграммы рис. 2.3 составим дискретную модель линейного стационарного объекта, дифференциальное уравнение которого имеет вид (2.3). Переход к дискретной модели осуществляется решением дифференциального уравнения на интервале $[mT, mT + T)$. Решение стационарного уравнения (2.3) имеет вид

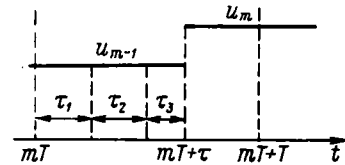


Рис. 2.3. Временная диаграмма обмена информацией между ЭВМ и ОУ

$$\mathbf{x}(mT + T) = \exp(\mathbf{A}T) \mathbf{x}(mT) + \int_{mT}^{mT+T} \exp[\mathbf{A}(mT + T - \theta)] \mathbf{B} \mathbf{u}(\theta) d\theta, \quad (2.41)$$

где m — номер интервала квантования (дискретное время), а T — интервал квантования.

Учитывая временную диаграмму рис. 2.3, выражение (2.41) перепишем в виде

$$\begin{aligned} \mathbf{x}(mT + T) = & \exp(\mathbf{A}T) \mathbf{x}(mT) + \int_{mT}^{mT+\tau} \exp[\mathbf{A}(mT + T - \theta)] \times \\ & \times \mathbf{B} d\theta \mathbf{u}_{m-1} + \int_{mT+\tau}^{mT+T} \exp[\mathbf{A}(mT + T - \theta)] \mathbf{B} d\theta \mathbf{u}_m. \end{aligned}$$

Заменяя в последнем выражении переменную интегрирования по формуле $mT - \theta = -\zeta$, а также введя обозначение $\mathbf{x}(mT) = \mathbf{x}_m$, получим

$$\begin{aligned} \mathbf{x}_{m+1} = & \exp(\mathbf{A}T) \mathbf{x}_m + \int_0^{\tau} \exp[\mathbf{A}(T - \zeta)] \mathbf{B} d\zeta \mathbf{u}_{m-1} + \\ & + \int_{\tau}^T \exp[\mathbf{A}(T - \zeta)] \mathbf{B} d\zeta \mathbf{u}_m. \end{aligned} \quad (2.42)$$

Выражению (2.42) соответствует схема моделирования дискретной системы, изображенная на рис. 2.4. На этом рисунке

ЭЭ — векторные элементы задержки, размерности которых равны размерностям задерживаемых на один такт векторов, а остальные элементы являются матричными усилителями. Полный вектор состояния дискретной системы равен объединению выходов элементов задержки, поэтому для дискретной модели ОУ в соответствии с рис. 2.4 вектор состояния будет следующий:

$$\mathbf{x}_m^p = \begin{bmatrix} \mathbf{x}_m \\ \vdots \\ \mathbf{u}_{m-1} \end{bmatrix}.$$

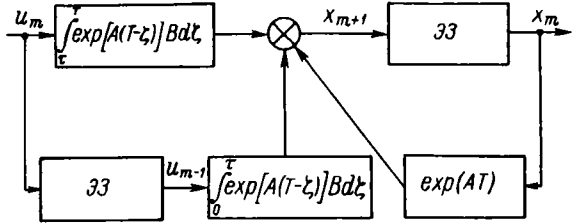


Рис. 2.4. Структурная дискретная модель объекта управления

С учетом введенного вектора, уравнение (2.42) переписывается в виде

$$\begin{bmatrix} \mathbf{x}_{m+1} \\ \vdots \\ \mathbf{u}_m \end{bmatrix} = \begin{bmatrix} \exp(AT) & \int_0^T \exp[A(T-\zeta)] \mathbf{B} d\zeta \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_m \\ \vdots \\ \mathbf{u}_{m-1} \end{bmatrix} + \begin{bmatrix} \int_0^T \exp[A(T-\zeta)] \mathbf{B} d\zeta \\ \vdots \\ \mathbf{I} \end{bmatrix} \mathbf{u}_m,$$

где \mathbf{I} — единичная матрица $m_u \times m_u$.
Вводя обозначения

$$\begin{cases} \mathbf{A}^p = \begin{bmatrix} \exp(AT) & \int_0^T \exp[A(T-\zeta)] \mathbf{B} d\zeta \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \end{bmatrix}; \\ \mathbf{B}^p = \begin{bmatrix} \int_0^T \exp[A(T-\zeta)] \mathbf{B} d\zeta \\ \vdots \\ \mathbf{I} \end{bmatrix}; \\ \mathbf{C}^p = [\mathbf{C} \mid \mathbf{0}], \end{cases} \quad (2.43)$$

получим разностную модель ОУ

$$\begin{cases} \mathbf{x}_{m+1}^p = \mathbf{A}^p \mathbf{x}_m^p + \mathbf{B}^p \mathbf{u}_m; \\ y_m = \mathbf{C}^p \mathbf{x}_m^p. \end{cases} \quad (2.44)$$

Дискретные модели возмущенных объектов. Рассмотрим линейный стационарный ОУ, на который кроме управления \mathbf{u} воздействует возмущение μ . Уравнение такого объекта имеет вид

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{B}_\mu\mu; \quad (2.45)$$

при этом будем считать, что возмущающее воздействие μ есть векторный выход командного генератора, который описывается уравнениями (см. п. 2.2):

$$\begin{cases} \dot{\xi} = \mathbf{\Gamma}\xi; \\ \mu = \mathbf{H}_\mu\xi. \end{cases} \quad (2.46)$$

Для получения разностной модели в этом случае объединим динамические системы (2.45), (2.46) в одну систему

$$\dot{\mathbf{x}}^1 = \mathbf{A}^1 \mathbf{x}^1 + \mathbf{B}^1 \mathbf{u}, \quad (2.47)$$

где

$$\mathbf{x}^1 = \begin{bmatrix} \mathbf{x} \\ \xi \end{bmatrix}; \quad \mathbf{A}^1 = \begin{bmatrix} \mathbf{A} & \mathbf{B}_\mu \mathbf{H}_\mu \\ \mathbf{0} & \mathbf{\Gamma} \end{bmatrix}; \quad \mathbf{B}^1 = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}. \quad (2.48)$$

Тогда (2.47) полностью совпадает с (2.3) и поведение этой динамической системы в дискретные моменты времени будет описываться уравнением (2.42), т. е.

$$\begin{aligned} \mathbf{x}_{m+1}^1 &= \exp(\mathbf{A}^1 T) \mathbf{x}_m^1 + \int_0^T \exp[\mathbf{A}^1 (T - \zeta)] \mathbf{B}^1 d\zeta \mathbf{u}_{m-1} + \\ &+ \int_{\tau}^T \exp[\mathbf{A}^1 (T - \zeta)] \mathbf{B}^1 d\zeta \mathbf{u}_m. \end{aligned} \quad (2.49)$$

Вычислим матричную экспоненту, используя ряд Тейлора,

$$\exp(\mathbf{A}^1 T) = \mathbf{I} + T\mathbf{A}^1 + \frac{T^2}{2!} (\mathbf{A}^1)^2 + \frac{T^3}{3!} (\mathbf{A}^1)^3 + \dots$$

Подставив в этот ряд выражение для \mathbf{A}^1 из (2.48), после соответствующих преобразований получим

$$\exp(\mathbf{A}^1 T) = \begin{bmatrix} \exp(\mathbf{A}T) & \mathbf{D}^1 \\ \mathbf{0} & \exp(\mathbf{\Gamma}T) \end{bmatrix},$$

где

$$\mathbf{D}^1 = \sum_{l=1}^{\infty} \frac{T^l}{l!} \sum_{j=1}^l \mathbf{A}^{l-j} \mathbf{B}_\mu \mathbf{H}_\mu \mathbf{\Gamma}^{j-1}.$$

Подставив x^1 , $\exp(A^1 T)$ и B^1 в (2.49), получим

$$\begin{bmatrix} x_{m+1} \\ \xi_{m+1} \end{bmatrix} = \begin{bmatrix} \exp(AT) & D^1 \\ 0 & \exp(\Gamma T) \end{bmatrix} \begin{bmatrix} x_m \\ \xi_m \end{bmatrix} + \\ + \begin{bmatrix} \int_0^T \exp[A(T-\zeta)] d\zeta B \\ 0 \end{bmatrix} u_{m-1} + \begin{bmatrix} \int_0^T \exp[A(T-\zeta)] d\zeta B \\ 0 \end{bmatrix} u_m.$$

От этого уравнения можно перейти к двум матричным разностным уравнениям, первое из которых имеет вид

$$x_{m+1} = \exp(AT) x_m + \int_0^T \exp[A(T-\zeta)] d\zeta B u_{m-1} + \\ + \int_0^T \exp[A(T-\zeta)] d\zeta B u_m + D^1 \xi_m \quad (2.50)$$

и описывает поведение ОУ, возбуждаемого возмущением, в дискретные моменты времени, а второе —

$$\xi_{m+1} = \exp(\Gamma T) \xi_m \quad (2.51)$$

является разностной моделью возмущающего воздействия.

Введем, как и ранее, вектор состояния

$$x_m^p = \begin{bmatrix} x_m \\ \dots \\ u_{m-1} \end{bmatrix},$$

тогда получим разностную модель ОУ, на который действует возмущение, в виде

$$x_{m+1}^p = A^p x_m^p + B^p u_m + D^p \xi_m, \quad (2.52)$$

где A^p , B^p имеют вид (2.43), а матрица

$$D^p = \begin{bmatrix} D^1 \\ \dots \\ 0 \end{bmatrix},$$

причем способ вычисления D^1 указан ранее.

Вектор возмущающего воздействия ξ в (2.52) изменяется в соответствии с уравнением (2.51).

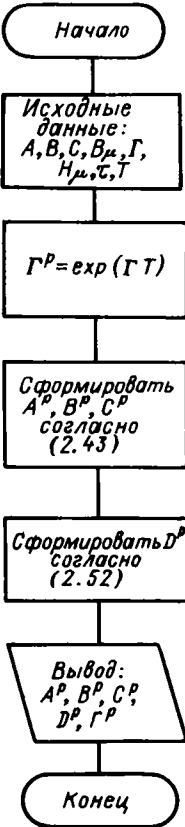


Рис. 2.5. Процедура формирования разностной модели

Алгоритм автоматизированного формирования разностной модели линейного стационарного объекта представлен на рис. 2.5.

Глава 3

СИНТЕЗ ЛИНЕЙНЫХ АЛГОРИТМОВ УПРАВЛЕНИЯ

3.1. Собственные и вынужденные движения в дискретной системе

Полное движение динамической системы содержит собственную и вынужденную составляющие. Собственное движение характеризует динамические свойства системы, такие как время затухания, колебательность переходных процессов. Вынужденное движение определяет поведение системы под влиянием внешних воздействий и характеризует точность воспроизведения входного воздействия g в системах слежения, а также степень подавления возмущающих воздействий μ . Алгоритмы управления должны обеспечивать желаемые свойства как свободных, так и вынужденных движений цифровых систем.

Рассмотрим процессы в линейной дискретной системе, возмущаемой внешним воздействием μ . Пусть уравнения замкнутой системы имеют вид:

$$\begin{cases} x_{m+1} = A_c x_m + B_\mu \mu_m; \\ y_m = C x_m, \end{cases} \quad (3.1)$$

где $x_m = x(mT) \in R^n$ — вектор состояния; $\mu_m = \mu(mT) \in R^q$ — вектор возмущающих воздействий; $y_m = y(mT) \in R^r$ — вектор выхода; A_c , B_μ , C — матрицы соответствующих размерностей. Как показано в п. 2.1, внешние воздействия можно представить как выход командного генератора:

$$\begin{cases} \xi_{m+1} = \Gamma \xi_m; \\ \mu_m = H_\mu \xi_m, \end{cases} \quad (3.2)$$

где $\xi_m = \xi(mT) \in R^l$ — вектор состояния КГ; Γ , H_μ — матрицы соответствующих размерностей.

Будем считать, что матрица Γ не имеет нулевых собственных чисел и поэтому обратима, что практически не сужает класса внешних воздействий, генерируемых системой (3.2). Движение линейной системы (3.1) представляется суммой собственного x_{cm} и вынужденного x_{vm} движений, т. е. $x_m = x_{cm} + x_{vm}$, что служит основанием для их раздельного рассмотрения.

Значения α , r и θ нетрудно найти из условий касания R_3 и R_1 в точке A (см. рис. 3.1) и касания R_3 и R_2 в точке B :

$$\begin{aligned}\exp(-3T/t_n) &= \alpha + r; \\ \exp(-\theta T) \cos \mu_n \theta T &= \alpha + r \cos \theta; \\ \exp(-\theta T) \sin \mu_n \theta T &= r \sin \theta.\end{aligned}$$

Решение этой системы относительно α , r и θ может быть получено одним из численных методов.

Круговая область желаемого расположения собственных значений $z_i = \lambda_i [A_c]$ матрицы замкнутой системы (3.3) в п. 3.2 используется для формирования весовых матриц суммарного показателя качества.

Регулятор, изменяющий матрицу A_c , будем называть регулятором собственного движения. Этот регулятор можно синтезировать, игнорируя внешнее воздействие.

Система (3.1) при нулевом векторе начального состояния $x(0) = 0$ имеет решение [83]

$$x_{вм} = \sum_{i=0}^{m-1} A_c^i B_\mu \mu_{m-i-1}$$

или, с учетом решения однородного уравнения (3.2),

$$x_{вм} = \sum_{i=0}^{m-1} A_c^i B_\mu H_\mu \Gamma^{m-i-1} \xi(0) = \sum_{i=0}^{m-1} A_c^i B_\mu H_\mu \Gamma^{-i-1} \xi_m. \quad (3.4)$$

Выражение (3.4) отражает фундаментальное свойство линейных динамических систем, заключающееся в том, что линейные системы осуществляют линейное, в общем случае нестационарное преобразование сигналов

$$x_{вм} = M_m \xi_m, \quad (3.5)$$

где, согласно (3.4),

$$M_m = \sum_{i=0}^{m-1} A_c^i B_\mu H_\mu \Gamma^{-i-1}. \quad (3.6)$$

Подставив (3.5) в (3.1), получим

$$M_{m+1} \xi_{m+1} = A_c M_m \xi_m + B_\mu H_\mu \xi_m$$

или, с учетом (3.2),

$$M_{m+1} \Gamma \xi_m = A_c M_m \xi_m + B_\mu H_\mu \xi_m,$$

причем $M_0 \xi_0 = 0$ для любых ξ_0 , т. е. $M_0 = 0$, так как из (3.6) видно, что $M_1 \Gamma = B_\mu H_\mu$.

Следовательно, справедливо матричное разностное уравнение

$$M_{m+1} = A_c M_m \Gamma^{-1} + B_\mu H_\mu \Gamma^{-1}, \quad M_0 = 0. \quad (3.7)$$

Решением этого уравнения является (3.6), что проверяется непосредственной подстановкой (3.6) в (3.7). Решение (3.7), на-

пример, в виде (3.6), с ростом m может стремиться к стационарной матрице M , так что $M_{m+1} = M_m = M$, следовательно, матрица M должна удовлетворять алгебраическому уравнению Сильвестра:

$$M\Gamma - A_c M = B_\mu H_\mu, \quad (3.8)$$

которое получается из (3.7). Стационарная матрица M , если она существует, определяет установившееся вынужденное движение

$$x_{\nu m}^y = M \xi_m. \quad (3.9)$$

Решение матричного разностного уравнения (3.7) связано с решением алгебраического уравнения (3.8) соотношением

$$M_m = A_c^m M \Gamma^{-m} + M, \quad (3.10)$$

которое обращает (3.7) в тождество при любых значениях m в силу выполнения (3.8).

Таким образом, вычисление вынужденного движения в любой момент времени сводится к решению матричного разностного уравнения (3.7), а вычисление вынужденного установившегося значения — к решению матричного алгебраического уравнения (3.8). Методом решения алгебраического уравнения Сильвестра (3.8) посвящена обширная литература [15, 21, 29, 42].

Решение алгебраического уравнения (3.8) может быть получено также в результате решения разностного уравнения (3.7) при $m \rightarrow \infty$, если решение (3.7) сходится к решению (3.8). Условие сходимости, как следует из (3.10), определяется свойствами пары матриц A_c , Γ и формулируется следующим образом [18].

Теорема 3.1. Пусть $\lambda_j[\Gamma] \neq 0$, $j = \overline{1, l}$. Решение разностного матричного уравнения (3.7) при $m \rightarrow \infty$ сходится к решению алгебраического матричного уравнения (3.8) из произвольного начального значения M_0 тогда и только тогда, когда для любых $i = \overline{1, n}$, $j = \overline{1, l}$

$$|\lambda_i[A_c] \lambda_j^{-1}[\Gamma]| < 1$$

($\lambda_i[\cdot]$ — собственные числа матрицы $[\cdot]$).

Как следует из теоремы 3.1, условия сходимости выполняются, если система (3.2) порождает более «медленные» движения, чем собственное движение системы (3.1).

Пример 3.1. Пусть в уравнении системы (3.1)

$$A_c = \begin{bmatrix} 0 & 1 \\ -0,3 & 1,1 \end{bmatrix}; \quad B_\mu = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

а в уравнении модели внешнего воздействия (3.2)

$$\Gamma = \begin{bmatrix} 1 & 0,1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad H_\mu = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Вектор внешнего воздействия ξ_m на основании (3.2) изменяется согласно выражению $\xi_m = \Gamma^m \xi_0$ и при заданной матрице Γ получим

$$\begin{bmatrix} \xi_{1m} \\ \xi_{2m} \\ \xi_{3m} \end{bmatrix} = \begin{bmatrix} 1 & 0,1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^m \begin{bmatrix} \xi_{10} \\ \xi_{20} \\ \xi_{30} \end{bmatrix}.$$

Для вычисления установившегося вынужденного движения решим алгебраическое матричное уравнение (3.8) с заданными матрицами A_c , B_μ , Γ и H_μ , в результате получим

$$M = \begin{bmatrix} -0,5 & 0,7475 & 5 \\ -1,5 & 0,7425 & 5 \end{bmatrix},$$

тогда координаты вектора $x_{\text{в}m}^y$ установившегося вынужденного движения системы будут:

$$x_{\text{в}1m}^y = -0,5\xi_{1m} + 0,7475\xi_{2m} + 5\xi_{3m};$$

$$x_{\text{в}2m}^y = -1,5\xi_{1m} + 0,7425\xi_{2m} + 5\xi_{3m}.$$

Учитывая закон изменения координат вектора ξ_m при заданной матрице Γ , окончательно получим

$$x_{\text{в}1m}^y = -0,5\xi_{10} + (0,7475 - 0,05m) \xi_{20} + 5\xi_{30};$$

$$x_{\text{в}2m}^y = -1,5\xi_{10} + (0,7425 - 0,15m) \xi_{20} + 5\xi_{30}.$$

Вынужденное движение, как это видно из выражений (3.5) и (3.9), определяется матрицей M_m . Изменяя эту матрицу, можем влиять на характер вынужденного движения. Матрица M_m в свою очередь при заданном классе входных воздействий зависит от матриц A_c и B_μ управляемой системы (3.1), как это видно из выражения (3.6). Свойства вынужденного движения, таким образом, зависят как от вида матрицы A_c , так и от вида матрицы B_μ , в то время как свойства собственного движения зависят только от вида матрицы A_c . Поэтому можно рекомендовать выбор необходимых параметров матрицы A_c на основании требований к переходному процессу системы. После синтеза регулятора собственного движения проверяются свойства вынужденного движения с использованием уравнения (3.8). Если эти свойства оказываются неудовлетворительными, то их коррекция осуществляется изменением матрицы B_μ . Регулятор, изменяющий свойства вынужденного движения, будем называть регулятором вынужденного движения.

3.2. Алгоритмы управления собственным движением

Синтез алгоритмов для ЭВМ, управляющей непрерывным объектом (см. рис. 1.10), предполагает использование дискретной модели непрерывного объекта, полученной в п. 2.5. В дальнейшем,

опуская верхние индексы в (2.44), представим уравнение разностной модели непрерывного объекта в виде:

$$\begin{cases} \mathbf{x}_{m+1} = \mathbf{A}\mathbf{x}_m + \mathbf{B}\mathbf{u}_m; \\ \mathbf{y}_m = \mathbf{C}\mathbf{x}_m. \end{cases} \quad (3.11)$$

Задача управляющей ЭВМ заключается в расчете на каждом такте координат вектора управления \mathbf{u}_m , как функций от переменных системы (3.11), доступных непосредственному измерению. Прежде всего рассмотрим систему с обратной связью по вектору состояния \mathbf{x}_m , полагая, что его координаты могут быть измерены (см. п. 1.1) или оцениваются с помощью специальных алгоритмов. Алгоритмы оценивания (см. пп. 3.7—3.8) в сочетании с обратной связью по состоянию образуют динамический алгоритм управления, т. е. корректирующий фильтр, широко распространенный в практических САУ.

Алгоритм управления. Исследуем возможности линейного закона управления

$$\mathbf{u}_m = -\mathbf{K}\mathbf{x}_m. \quad (3.12)$$

Алгоритм функционирования управляющей ЭВМ, реализующей закон управления (3.12), изображен на рис. 3.2. Блок 1 этого алгоритма соответствует пуску управляющей ЭВМ. Сразу после пуска выполняется специальная программа установки начального состояния ЭВМ (в счетчик команд заносится первый адрес области ПЗУ, в которой размещена эта про-

Рис. 3.2. Алгоритм управления собственным движением

грамма). В процессе выполнения блока 2 устанавливаются определенные регистры процессора и ячейки двустороннего ОЗУ в заданное состояние, тем самым обеспечивается нормальное начало функционирования ЭВМ.

Реализация закона управления (3.12) начинается с блока 3, в котором опрашиваются измерительные устройства, установленные на объекте для получения координат вектора состояния \mathbf{x} , осуществляется преобразование аналоговых сигналов в двоичный код, ввод их в процессор и запись в определенные ячейки оперативной памяти. В блоке 4 рассчитывается управляющее воздействие согласно (3.12). При выполнении блока 5 рассчитанные координаты вектора управляющего воздействия выводятся на цифроаналоговые преобразователи, где преобразуются в аналоговые сигналы, запоминаемые ЦАП до следующего момента вывода \mathbf{u} ; аналоговые выходные сигналы ЦАП воздействуют на исполнительные механизмы объекта управления.

Работа ЭВМ должна быть организована таким образом, чтобы от момента начала выполнения блока 3 до момента окончания

выполнения блока 5 прошло время τ , которое принималось в качестве времени запаздывания при построении разностной модели (3.11) объекта управления. В том случае, когда запаздывание τ меньше интервала квантования T , после блока 5 должен включаться таймер, дополняющий время выполнения блоков 3—5 до значения T ; вместо таймера могут включаться блоки фоновых алгоритмов, время выполнения которых должно быть подобрано таким образом, чтобы суммарное время выполнения всей программы ЭВМ равнялось интервалу T , принятому при составлении разностной модели объекта. Невыполнение перечисленных требований обычно приводит к тому, что свойства цифровой системы отличаются от тех желаемых свойств, которые задаются при синтезе закона управления (3.12).

В алгоритме на рис. 3.2 отсутствует блок «КОНЕЦ», это объясняется тем, что прекращение выполнения программы управления определяется причинами, внешними по отношению к выполняемой программе, например, остановкой ЭВМ, выключением объекта управления и т. д.

Поведение динамической системы, состоящей из объекта управления и управляющей ЭВМ, реализующей алгоритм рис. 3.1, описывается системой из двух разностных уравнений (3.11) и (3.12). Исключив промежуточную переменную u_m , получим разностное уравнение

$$x_{m+1} = A_c x_m, \quad (3.13)$$

где

$$A_c = A - BK. \quad (3.14)$$

Уравнение (3.13) совпадает с (3.3) и, следовательно, описывает собственное движение исследуемой динамической системы. Надлежащим выбором матрицы K можно получить матрицу A_c с заданными свойствами [40, 44, 83], т. е. можно обеспечить наперед заданные свойства собственного движения (3.13) рассматриваемой системы. В современной инженерной практике распространены два основных метода расчета матрицы K обратных связей регулятора собственного движения.

Метод модального управления. В соответствии с методом матрица обратной связи K находится из условия равенства собственных чисел (мод) матрицы A_c наперед заданным значениям [4, 24, 32, 38, 83]. Процедура синтеза начинается с выбора корней характеристического уравнения (мод) замкнутой проектируемой системы, при этом должно обеспечиваться требуемое качество переходных процессов, например, заданное время затухания t_n и заданная колебательность μ_n процессов. Такие корни должны лежать в области P_d (см. рис. 3.1).

Желаемые корни характеристического уравнения могут быть выбраны также при помощи стандартных полиномов [38, 75], когда на основании заданного времени затухания и перерегули-

рования непрерывного процесса на выходе системы выбирается стандартный полином и значения его коэффициентов. Затем находятся корни этого полинома s_1, s_2, \dots, s_n и рассчитываются корни дискретного полинома, определяющего дискретный процесс, получающийся квантованием желаемого непрерывного процесса,

$$z_i = \exp(s_i T), \quad i = \overline{1, n}.$$

Выбранные корни определяют характеристический полином проектируемой дискретной системы

$$D(z) = (z - z_1) \dots (z - z_n) = z^n + \alpha_1 z^{n-1} + \dots + \alpha_n. \quad (3.15)$$

Построим эталонную динамическую систему $\zeta_{m+1} = F\zeta_m$, характеристический полином которой совпадает с (3.15). Для того чтобы эталонная система имела в качестве характеристического полинома систему (3.15), достаточно, чтобы матрица F была сопровождающей матрицей полинома (3.15) [15, 42], т. е.

$$F = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -\alpha_n & -\alpha_{n-1} & -\alpha_{n-2} & \dots & -\alpha_1 \end{bmatrix}.$$

Известно [24], что различные линейные системы имеют одинаковые характеристические полиномы, если вектора состояния этих систем для любого момента времени связаны линейным преобразованием, поэтому будем синтезировать регулятор собственного движения таким образом, чтобы возможная ошибка

$$e_m = x_m - M\zeta_m \quad (3.16)$$

стремилась к нулю с течением времени (ζ_m, x_m — вектора состояния эталонной и проектируемой систем, а M — неизвестная матрица линейного преобразования).

Найдем разностное уравнение ошибки (3.16)

$$e_{m+1} = x_{m+1} - M\zeta_{m+1}$$

или после подстановки (3.13)—(3.16)

$$e_{m+1} = (A - BK)e_m + (A - BK)M\zeta_m - MF\zeta_m. \quad (3.17)$$

Матрица состояния $(A - BK)$ проектируемой системы должна обеспечивать устойчивые процессы, поэтому ошибка, определяемая разностным уравнением (3.17), будет стремиться к нулю при выполнении условия

$$(A - BK)M - MF = 0$$

или

$$AM - MF = BKM. \quad (3.18)$$

Таким образом, система (3.13) будет обладать желаемым характеристическим уравнением, если матрицы K и M удовлетворяют уравнению (3.18). Этому уравнению могут удовлетворять сколь угодно много пар матриц K и M . Для придания определенности задаче синтеза примем

$$KM = H \quad (3.19)$$

и выберем матрицу H наиболее простой структуры (например, с максимально возможным числом нулевых элементов) таким образом, чтобы пара матриц (F, H) была полностью наблюдаема. Уравнение (3.18) при условии полной наблюдаемости пары (F, H) и полной управляемости пары (A, B) имеет решение M (в этом случае матрица M обратима). Тогда из (3.19) вычисляем $K = HM^{-1}$.

Алгоритм автоматизированного синтеза матрицы K обратных связей по изложенной методике изображен на рис. 3.3.

Пример 3.2. Пусть в разностной модели (3.11) получены следующие параметры:

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad C = [1 \quad 0]$$

при $\tau = T = 0,2$ с.

Необходимо синтезировать матрицу обратных связей таким образом, чтобы время затухания процесса было $t_{\text{п}} = 0,6$ с. Степень устойчивости для непрерывного процесса будет $\eta = 3/t_{\text{п}} = 5 \text{ с}^{-1}$, для дискретной системы $\eta_d = \exp(-\eta T) = 0,37$. Таким образом, корни характеристического уравнения проектируемой дискретной системы должны лежать внутри окружности радиуса 0,37. Выберем $z_1 = 0,3$; $z_2 = 0,1$, тогда характеристический полином проектируемой системы будет

$$D(z) = (z - 0,3)(z - 0,1) = z^2 - 0,4z + 0,03,$$

а матрица состояния эталонной системы —

$$F = \begin{bmatrix} 0 & 1 \\ -0,03 & 0,4 \end{bmatrix}.$$

Выберем $H = [0 \quad 1]$, при этом пара матриц (F, H) наблюдаема. Отметим, что пара (A, B) в рассматриваемом примере управляема. Подставив A, B, F и H в (3.18), найдем матрицу M в виде

$$M = \begin{bmatrix} 0,95 & 1,59 \\ -1 & 0 \end{bmatrix},$$

тогда матрица обратной связи

$$K = HM^{-1} = [0 \quad 1] \begin{bmatrix} 0 & -1 \\ 0,63 & 0,6 \end{bmatrix} = [0,63 \quad 0,6].$$

Матрица состояния A_c уравнения (3.13) свободного движения

$$A_c = A - BK = \begin{bmatrix} 1 & 1 \\ -0,63 & -0,6 \end{bmatrix}.$$

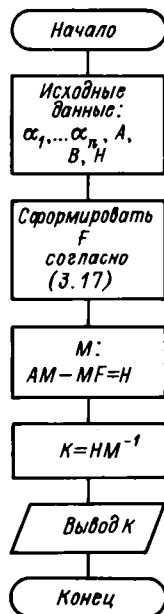


Рис. 3.3. Процедура синтеза модального управления

Свободное движение в системе (3.13) для полученной матрицы A_0 и начального состояния $x(0) = [1 \ 0]^T$ изображено на рис. 3.4. Переходный процесс заканчивается за 3 интервала дискретности, т. е. за 0,6 с.

Метод оптимального управления. Широкое распространение на практике получило оптимальное управление собственным движением в дискретных системах [20, 40, 41, 60, 75, 83]. Для объекта (3.11) ищется такой закон управления, который минимизирует функционал

$$J = \sum_{m=0}^{N-1} (x_m^T Q x_m + u_m^T R u_m) + x_N^T Q_N x_N, \quad (3.20)$$

где Q , R и Q_N — симметрические положительно определенные матрицы штрафа на текущее состояние, текущее управление и конечное состояние соответственно; N — число интервалов, в течение которых осуществляется процесс управления. В частном случае N может стремиться к бесконечности, тогда $Q_N = 0$.

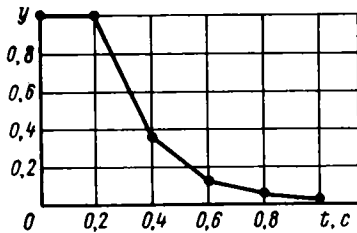


Рис. 3.4. График переходного процесса

Решение этой задачи хорошо известно. Оптимальное управление имеет вид

$$u_m = -K_{N-m} x_m, \quad (3.21)$$

где матрица нестационарных обратных связей вычисляется по формуле

$$K_{m+1} = [R + V^T P_m V]^{-1} V^T P_m A, \quad (3.22)$$

а матрица P_m находится в результате решения нелинейного разностного уравнения

$$P_{m+1} = Q + K_{m+1}^T R K_{m+1} + [A - B K_{m+1}]^T P_m [A - B K_{m+1}];$$

$$P_0 = Q_N. \quad (3.23)$$

Обычно решение уравнения (3.23) быстро стабилизируется и матрица обратных связей также становится постоянной, поэтому на практике вместо оптимального закона управления (3.21) используется стационарный квазиоптимальный закон (3.12), в котором матрица K вычисляется из системы алгебраических уравнений:

$$K = (R + V^T P V)^{-1} V^T P A; \quad (3.24)$$

$$P = Q + K^T R K + (A - B K)^T P (A - B K), \quad (3.25)$$

получающейся из (3.22) и (3.23) при условии, что

$$K_{m+1} = K_m = K, \quad P_{m+1} = P_m = P.$$

Изложенный способ синтеза управления, минимизирующего функционал (3.20), обладает существенным недостатком — в спо-

собе отсутствуют рекомендации по выбору матриц штрафа этого функционала. В результате любая устойчивая линейная система оказывается оптимальной в смысле некоторого квадратичного функционала [44]. Поэтому в инженерной практике при синтезе оптимального управления приведенным методом приходится выполнять перебор матриц штрафа функционала (3.20) до тех пор, пока не будет получено приемлемое качество собственного движения в проектируемой системе.

Изменим задачу синтеза оптимального управления таким образом [20], чтобы исключался перебор матриц штрафа функционала (3.20). Будем считать, что проектируемая система обладает требуемыми свойствами, если собственные числа матрицы A_c проектируемой системы лежат внутри окружности с параметрами r и α (см. рис. 3.1).

Рассмотрим вспомогательную динамическую систему, уравнение которой имеет вид

$$\zeta_{m+1} = A_1 \zeta_m + B_1 v_m; \quad (3.26)$$

матрицы этого уравнения связаны с матрицами объекта (3.11) следующими соотношениями:

$$A_1 = r^{-1}(A - \alpha I); \quad B_1 = r^{-1}B, \quad (3.27)$$

где r и α определяют окружность (см. п. 3.1 и рис. 3.1), а I — единичная матрица.

Для объекта (3.26) рассчитаем управление

$$v_m = -K \zeta_m \quad (3.28)$$

таким образом, чтобы минимизировался функционал

$$J_1 = \sum_{m=0}^{\infty} (\zeta_m^T Q_1 \zeta_m + v_m^T R_1 v_m),$$

где Q_1 и R_1 — произвольные положительно определенные матрицы, например единичные.

Матрица обратных связей K в управлении (3.28), минимизирующем функционал J_1 , вычисляется на основании (3.24), (3.25)

$$K = (R_1 + B_1^T P B_1)^{-1} B_1^T P A_1; \quad (3.29)$$

$$P = Q_1 + K^T R_1 K + (A_1 - B_1 K)^T P (A_1 - B_1 K). \quad (3.30)$$

Матрица

$$A_{c1} = A_1 - B_1 K \quad (3.31)$$

является матрицей замкнутой системы, состоящей из объекта (3.26) и регулятора, вырабатывающего управляющее воздействие в соответствии с (3.28). С учетом (3.31) уравнение (3.30) можно переписать в виде

$$A_{c1}^T P A_{c1} - P = -Q_1 - K^T R_1 K,$$

т. е. в виде уравнения Ляпунова, причем матрицы P и $Q_1 + K^T R_1 K$ являются положительно определенными.

Таким образом, матрица A_{c1} является устойчивой [40] и собственные числа этой матрицы лежат внутри окружности единичного радиуса.

Выберем теперь для исходного ОУ (3.11) регулятор, реализующий закон управления (3.12), где матрица обратных связей K та же, что и в (3.28). Уравнение замкнутой системы имеет вид (3.13), а матрица A_c определяется выражением (3.14). Сопоставим матрицы A_{c1} и A_c , для этого в (3.31) подставим (3.27), в результате получим

$$A_{c1} = r^{-1}(A - \alpha I) - r^{-1}BK = r^{-1}(A - BK - \alpha I),$$

т. е.

$$A_{c1} = r^{-1}(A_c - \alpha I). \quad (3.32)$$

Известно [15], что собственные числа матрицы, являющейся функцией некоторой матрицы-аргумента, равны той же функции от собственных чисел матрицы-аргумента, тогда на основании (3.32) имеем

$$\lambda_i[A_{c1}] = r^{-1}\lambda_i[A_c - \alpha I], \quad i = \overline{1, n}$$

или

$$\lambda_i[A_c] = r\lambda_i[A_{c1}] + \alpha, \quad i = \overline{1, n}.$$

Из последнего выражения следует, что если все собственные числа матрицы A_{c1} лежат внутри окружности единичного радиуса с началом в центре координат, то все собственные числа матрицы A_c лежат внутри окружности радиуса r с центром, смещенным по вещественной оси на α (см. рис. 3.1). Таким образом, в рассмотренной постановке оптимальной задачи одновременно с минимизацией квадратичного функционала обеспечивается наперед заданное расположение корней характеристического уравнения замкнутой системы.

Алгоритм синтеза матрицы обратных связей в рассмотренной постановке изображен на рис. 3.5. В качестве решения уравнения Риккати в этом алгоритме используется установившееся решение матричного разностного уравнения (3.22), (3.23) с нулевым начальным условием. При расчете матрицы K по предлагаемой методике имеется неопределенность выбора матриц Q_1 , R_1 , которая может быть использована для достижения каких-либо дополнительных целей. Эти дополнительные свойства проектируемой системы обеспечиваются решением задачи нелинейного программирования относительно элементов матриц Q_1 и R_1 .

Алгоритм синтеза матрицы обратных связей в рассмотренной постановке изображен на рис. 3.5. В качестве решения уравнения Риккати в этом алгоритме используется установившееся решение матричного разностного уравнения (3.22), (3.23) с нулевым начальным условием. При расчете матрицы K по предлагаемой методике имеется неопределенность выбора матриц Q_1 , R_1 , которая может быть использована для достижения каких-либо дополнительных целей. Эти дополнительные свойства проектируемой системы обеспечиваются решением задачи нелинейного программирования относительно элементов матриц Q_1 и R_1 .

Можно показать, что синтезированное по изложенной методике управление обеспечивает минимизацию квадратичного функционала

$$J = \sum_{m=0}^{\infty} (\mathbf{x}_m^T \mathbf{Q} \mathbf{x}_m + \mathbf{x}_m^T \mathbf{S} \mathbf{u}_m + \mathbf{u}_m^T \mathbf{R} \mathbf{u}_m),$$

где матрицы \mathbf{Q} , \mathbf{S} и \mathbf{R} выражаются через матрицы функционала J_1 :

$$\mathbf{Q} = (1 - r^2 + \alpha^2) \mathbf{P} + r^2 \mathbf{Q}_1 - \alpha (\mathbf{P} \mathbf{A} + \mathbf{A}^T \mathbf{P});$$

$$\mathbf{R} = r^2 \mathbf{R}_1; \quad \mathbf{S} = -\alpha \mathbf{P} \mathbf{B},$$

а \mathbf{P} есть решение матричного уравнения (3.30).

Важным достоинством рассмотренных методов синтеза алгоритмов управления собственным движением (см. рис. 3.3 и 3.5) является их инвариантность к размерности вектора управления.

3.3. Алгоритмы комбинированного управления

Рассмотрим ОУ, возмущенный внешним воздействием. Опустив верхние индексы в (2.52), запишем разностную модель возмущенного объекта в виде

$$\begin{cases} \mathbf{x}_{m+1} = \mathbf{A} \mathbf{x}_m + \mathbf{B} \mathbf{u}_m + \mathbf{D} \xi_m; \\ \mathbf{y}_m = \mathbf{C} \mathbf{x}_m. \end{cases} \quad (3.33)$$

Положим, что разностная модель внешнего воздействия (2.20) — (2.22) имеет вид

$$\begin{cases} \xi_{m+1} = \Gamma \xi_m; \\ \mu_m = \mathbf{H}_\mu \xi_m; \\ \mathbf{g}_m = \mathbf{H}_g \xi_m, \end{cases} \quad (3.34)$$

где μ_m — возмущающее внешнее воздействие; \mathbf{g}_m — задающее воздействие, которое должно отслеживаться выходной величиной объекта \mathbf{y}_m .

Используем для управления объектом (3.33) регулятор собственного движения (3.12), синтезированный для автономного объекта. Уравнение динамической системы, состоящей из объекта (3.33) и регулятора (3.12), реализуемого управляющей ЭВМ, будет

$$\begin{cases} \mathbf{x}_{m+1} = \mathbf{A}_c \mathbf{x}_m + \mathbf{D} \xi_m; \\ \mathbf{y}_m = \mathbf{C} \mathbf{x}_m, \end{cases} \quad (3.35)$$

где матрица состояния замкнутой системы равна $\mathbf{A}_c = \mathbf{A} - \mathbf{B} \mathbf{K}$.

Внешнее воздействие вызовет вынужденное движение, которое после окончания переходного процесса будет определяться выражением

$$\mathbf{x}_{\mathbf{y}m}^{\mathbf{y}} = \mathbf{M} \xi_m, \quad (3.36)$$

как это показано в п. 3.1.

Матрица M в (3.36) определяется в результате решения уравнения (3.8), которое для данного случая записывается в виде

$$M\Gamma - A_c M = D. \quad (3.37)$$

Установившаяся ошибка слежения за входным воздействием в системе (3.35) будет

$$e_m^y = g_m - y_{zm}^y$$

или

$$e_m^y = H_g \xi_m - C x_{zm}^y.$$

Учитывая (3.37), получим

$$e_m^y = (H_g - CM) \xi_m. \quad (3.38)$$

Полученная формула остается справедливой и при отсутствии входного воздействия, в этом случае в (3.38) необходимо положить

$H_g = 0$. Если установившаяся ошибка (3.38) не превосходит заданную в технических условиях на проектирование, то в системе управления используются только регулятор свободной движения (только обратные связи по состояниям объекта управления), и управляющая ЭВМ реализует алгоритм рис. 3.2; если же установившаяся ошибка не удовлетворяет требованиям к точности системы управления, то необходимо скорректировать алгоритм расчета управляющего воздействия на управляющей ЭВМ.

Изменение ошибки, как это видно из (3.36), обеспечивается изменением матрицы M , являющейся решением уравнения (3.37). В этом уравнении матрица Γ задана, поскольку задан класс внешних воздействий на объект, а матрица A_c получена в результате синтеза регулятора собственного движения, поэтому единственная возможность воз-

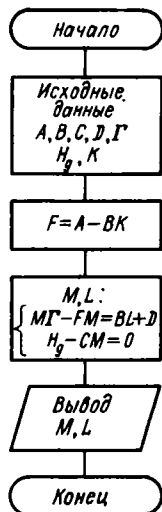


Рис. 3.6. Процедура синтеза связей по внешним воздействиям

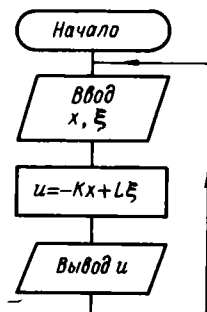


Рис. 3.7. Алгоритм комбинированного управления вынужденным движением

действия на матрицу M в данной постановке заключается в изменении правой части уравнения (3.37). Для достижения этой цели используем алгоритм комбинированного управления вида

$$u_m = -Kx_m + L\xi_m. \quad (3.39)$$

где матрица K имеет то же значение, что и в (3.12), а матрица связей по внешнему воздействию L пока не определена.

Принимая закон управления (3.39), считаем, что имеется техническая возможность измерения не только координат вектора состояния x_m объекта управления, но и координат вектора состояния ξ_m динамической системы, генерирующей внешнее воздействие.

Уравнение замкнутой системы, состоящей из объекта (3.33) и регулятора (3.39), будет

$$\begin{cases} x_{m+1} = A_c x_m + (BL + D) \xi_m; \\ y_m = Cx_m, \end{cases} \quad (3.40)$$

где матрица A_c имеет то же значение, что и в (3.35), т. е. введение связей по внешнему воздействию в закон управления не изменяет свойств собственного движения.

Уравнение (3.37) для системы (3.40) преобразуется к виду

$$MG - A_c M = BL + D, \quad (3.41)$$

откуда видно, что изменением матрицы L можно менять матрицу M .

Воспользуемся этой возможностью таким образом, чтобы обеспечить выполнение условия

$$H_g - CM = 0; \quad (3.42)$$

при этом установившаяся ошибка в системе будет равна нулю, т. е. система будет инвариантна к внешнему воздействию, генерируемому системой (3.34).

Таким образом, в результате решения системы матричных уравнений (3.41), (3.42) находим матрицу связей по внешним воздействиям L , обеспечивающих инвариантность системы к определенному типу внешнего воздействия. Способы решения системы матричных уравнений (3.41), (3.42) можно найти в [15, 29, 42, 75]. Алгоритм автоматизированного синтеза комбинированного регулятора вынужденного движения изображен на рис. 3.6. Синтез этого регулятора сводится к решению матричных уравнений. Для реализации комбинированного управления требуется измерение всех координат векторов состояния объекта управления и командного генератора внешнего воздействия. Алгоритм функционирования управляющей ЭВМ для этого случая изображен на рис. 3.7.

Пример 3.3. Необходимо синтезировать алгоритм управляющей ЭВМ, обеспечивающий стабилизацию угла поворота исполнительного электродвигателя постоянного тока. Уравнение двигателя имеет вид

$$T_0 \frac{d\omega}{dt} + \omega = k_d v + k_M M; \quad \frac{d\alpha}{dt} = \omega,$$

где T_0 — электромеханическая постоянная времени двигателя; k_d — коэффициент передачи двигателя с редуктором по управляющему воздействию; k_M — коэффициент передачи двигателя с редуктором по возмущающему воздействию; ω — угловая скорость; v — входное напряжение; α — угол поворота; M — момент сопротивления.

Рассмотрим случай, когда электромеханической постоянной времени двигателя можно пренебречь ($T_g \approx 0$), тогда уравнение объекта управления будет

$$\frac{d\alpha}{dt} = k_d v + k_M M.$$

Введем в рассмотрение безразмерные переменные:

$$x = \alpha/\alpha_B; \quad u = v/v_B; \quad \mu = M/M_B,$$

где α_B , v_B , M_B — базовые значения переменных.

С использованием безразмерных переменных уравнение объекта будет

$$\dot{x} = au + b\mu,$$

где $a = k_d v_B/\alpha_B$; $b = k_M M_B/\alpha_B$ (в рассматриваемом примере примем $a = 10 \text{ с}^{-1}$; $b = 5 \text{ с}^{-1}$).

По условиям эксплуатации уставка g меняется редко и в расчетах ее можно считать постоянной. Тогда уравнение, описывающее изменение g , имеет вид $\dot{g} = 0$. Отметим, что g физически представляет собой угол; будем его измерять в относительных единицах при условии, что $g_B = \alpha_B$. Момент сопротивления изменяется по гармоническому закону с круговой частотой 5 с^{-1} , закон изменения момента в относительных единицах имеет вид $\mu = \cos 5t$. Введем в рассмотрение переменные состояния КГ внешних воздействий $\xi_1 = \mu$, $\xi_2 = \dot{\mu}$, $\xi_3 = g$, тогда в уравнении непрерывного командного генератора (2.20) на основании уравнений для g и μ получим:

$$\Gamma = \begin{bmatrix} 0 & 1 & 0 \\ -25 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad H_\mu = [1 \ 0 \ 0]; \quad H_g = [0 \ 0 \ 1].$$

Будем считать, что в проектируемой цифровой системе выбран интервал квантования $T = 0,1 \text{ с}$ и только половина этого интервала отводится на реализацию закона управления, т. е. $\tau = 0,5T = 0,05 \text{ с}$.

Разностные модели объекта управления и внешнего воздействия, полученные при этих условиях по методике п. 2.4, имеют вид:

$$x_{m+1}^p = A^p x_m^p + B^p u_m + D^p \xi_m;$$

$$y_m = C^p x_m^p;$$

$$\xi_{m+1} = \Gamma^p \xi_m;$$

$$\mu_m = H_\mu \xi_m; \quad g_m = H_g \xi_m.$$

где

$$x^p = \begin{bmatrix} x_m \\ u_{m-1} \end{bmatrix}; \quad \xi_m = \begin{bmatrix} \xi_{1m} \\ \xi_{2m} \\ \xi_{3m} \end{bmatrix}; \quad A^p = \begin{bmatrix} 1 & \int_0^\tau a d\zeta \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0,5 \\ 0 & 0 \end{bmatrix};$$

$$B^p = \begin{bmatrix} \int_0^\tau a d\zeta \\ 1 \end{bmatrix} = \begin{bmatrix} 0,5 \\ 1 \end{bmatrix}; \quad C^p = [1 \ 0];$$

$$D^p = \begin{bmatrix} D^1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0,504 & 0,025 & 0 \\ 0 & 0 & 0 \end{bmatrix};$$

$$\Gamma^p = \exp(\Gamma T) = \begin{bmatrix} 0,878 & 0,096 & 0 \\ -2,397 & 0,878 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Матрицу обратных связей K синтезируем методом модального управления в виде $K = [0,2 \quad -0,2]$; при этом условии собственные числа матрицы A_c совпадают и равны 0,55. Кривая 1 (рис. 3.8) — собственное движение автономной системы (3.3) для данного случая; кривая 2 — возмущающее воздействие μ . Если использовать только регулятор свободного движения в системе стабилизации скорости, то установившаяся ошибка, вычисленная согласно (3.37), (3.38), при $H_d = 0$ имеет вид, соответствующий кривой 3 (рис. 3.8). Характер кривой 3 показывает, что возмущающее воздействие в этом случае не только не подавляется, но даже амплитуда ошибки оказывается больше амплитуды возмущения. В целях ликвидации установившейся ошибки введем прямую связь по внешнему воздействию. Матрица L , найденная в результате решения системы уравнений (3.41), (3.42), имеет вид $L = [-0,38 \quad -0,05 \quad 0,2]$. Подставляя найденные значения K и L в (3.39), получим выражение для расчета управляющего воздействия в ЭВМ

$$u_m = 0,2(g_m - y_m) + 0,2u_{m-1} + 0,38\xi_{1m} - 0,05\xi_{2m}$$

(коэффициенты безразмерны и заносятся в ЗУ при программировании в том случае, когда одна относительная единица принята равной одной машинной единице, если же это условие не выполняется, то осуществляется перемасштабирование алгоритма управления по методике, описанной в гл. 6).

График ошибки в системе при указанном законе управления представлен кривой 4 (рис. 3.8), откуда видно, что комбинированное управление является эффективным средством увеличения точности цифровых систем в установившемся режиме.

Основным недостатком комбинированного управления является его чувствительность к изменению параметров объекта управления. Действительно, пусть изменились параметры объекта (3.33) таким образом, что соответствующие матрицы получили приращения ΔA , ΔB , ΔC , ΔD . Если для объекта с новыми значениями параметров использовать управление (3.39), то уравнение замкнутой системы вместо (3.40) будет иметь вид:

$$\begin{aligned} x_{m+1} &= (A_c + \Delta A_c) x_m + [(B + \Delta B) L + (D + \Delta D)] \xi_m; \\ y_m &= (C + \Delta C) x_m, \end{aligned}$$

где матрица L вычисляется в результате совместного решения уравнений (3.41), (3.42), $A_c = A - BK$ и $\Delta A_c = \Delta A - \Delta BK$.

В этой замкнутой системе появляется установившаяся ошибка, равная в соответствии с (3.38)

$$e_m^y = [H_g - (C + \Delta C) M_1] \xi_m,$$

причем матрица M_1 удовлетворяет уравнению (3.37), которое в рассматриваемом случае принимает вид

$$M_1 \Gamma - (A_c + \Delta A_c) M_1 = D + \Delta D + (B + \Delta B) L. \quad (3.43)$$

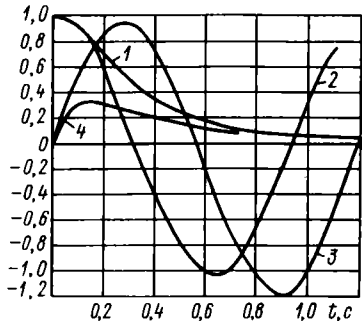


Рис. 3.8. Переходные процессы в комбинированной системе: 1 — собственное движение; 2 — возмущающее воздействие; 3 — установившаяся ошибка в системе без комбинированной связи; 4 — установившаяся ошибка в комбинированной системе

Представим M_1 в виде $M_1 = M + \Delta M$, где матрица M удовлетворяет системе матричных уравнений (3.41), (3.42), тогда уравнение (3.43) переписывается в виде

$$\Delta M \Gamma - (A_c + \Delta A_c) \Delta M = \Delta A_c M + \Delta B L + \Delta D. \quad (3.44)$$

Выражение для вынужденной ошибки, которая появляется в комбинированной системе за счет изменения параметров объекта управления, примет вид

$$e_m^y = [-(C + \Delta C) \Delta M - \Delta C M] \xi_m. \quad (3.45)$$

Пример 3.4. Пусть коэффициент передачи электродвигателя, рассматриваемого в примере 3.3, увеличился на 10 %, тогда коэффициент уравнения двигателя в относительных единицах $a = 11 \text{ с}^{-1}$ и матрицы A^P и B^P получают приращения

$$\Delta A = \begin{bmatrix} 0 & 0,05 \\ 0 & 0 \end{bmatrix}; \quad \Delta B = \begin{bmatrix} 0,05 \\ 0 \end{bmatrix}.$$

Используя матрицу обратных связей, рассчитанную в примере 3.3, $K = [0,2 \quad -0,2]$, получим

$$\Delta A_c = \begin{bmatrix} -0,01 & 0,06 \\ 0 & 0 \end{bmatrix}.$$

Подставив найденные приращения параметров системы в (3.44) и используя значения L и M , найденные в предыдущем примере, получим

$$\Delta M = \begin{bmatrix} 0,03 & 0,18 & 0 \\ 0,01 & 0,07 & 0 \end{bmatrix},$$

тогда в соответствии с (3.45) при возмущении (рис. 3.8, кривая 2) получим

$$e_m^y = 0,1 \sin(0,5t - 0,44).$$

Эта ошибка измеряется в долях α_0 (см. пример 3.3).

Таким образом, параметрические возмущения объекта управления приводят к нарушению условий инвариантности, однако ошибка в системе с комбинированным управлением остается значительно меньшей по сравнению с системой, в которой связи по внешнему воздействию разомкнуты.

3.4. Грубый алгоритм управления вынужденным движением

К *грубым*, или *робастным*, системам управления относится широкий класс САУ, обеспечивающих приемлемое качество работы при изменении характеристик внешних воздействий и параметров объекта управления. В частности, к этому классу принадлежат системы, в которых уменьшение установившейся ошибки достигается за счет повышения астатизма алгоритма управления или придания ему более общих динамических свойств, позволяющих компенсировать влияние внешних воздействий [4, 39, 40, 84]. Возможность получения нулевой ошибки для некоторого диапазона изменения параметров объекта позволяет отнести рассматриваемый тип алгоритма к *грубым алгоритмам управления*.

Алгоритм строится следующим образом. Пусть разностная модель объекта управления имеет вид (3.33), причем выход объ-

екта y_m скалярный, т. е. матрица C является строкой. Будем считать также, что разностная модель внешнего воздействия имеет вид (3.34), причем g_m является скалярной функцией времени. Алгоритм дискретного регулятора будем проектировать таким образом, чтобы регулятор реализовывал модель внешнего воздействия, возбуждаемую ошибкой системы, т. е. потребуем, чтобы в состав алгоритма регулятора входило уравнение

$$z_{m+1} = \Gamma z_m + E e_m. \quad (3.46)$$

Здесь матрица Γ та же, что и в (3.34); E — матрица-столбец, образующая управляемую пару (Γ, E) ; z_m — вектор состояния динамической системы (3.46), а ошибка e_m вычисляется по формуле

$$e_m = g_m - y_m = H_g \xi_m - C x_m. \quad (3.47)$$

При этих условиях структурная схема разомкнутой системы представлена на рис. 3.9.

Примем в качестве нового объекта управления динамическую систему, составленную из подсистем (3.33) и (3.46). Уравнение такого ОУ с учетом (3.47) будет следующее:

$$\begin{bmatrix} x_{m+1} \\ z_{m+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -EC & \Gamma \end{bmatrix} \begin{bmatrix} x_m \\ z_m \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_m + \begin{bmatrix} D \\ EH_g \end{bmatrix} \xi_m. \quad (3.48)$$

Если пара матриц (A, B) полностью управляема, пара (A, C) хотя бы частично наблюдаема, а пара матриц (Γ, E) полностью управляема, то пара

$$\left(\begin{bmatrix} A & 0 \\ -EC & \Gamma \end{bmatrix}, \begin{bmatrix} B \\ 0 \end{bmatrix} \right)$$

полностью управляема.

Тогда для объекта (3.48) можно построить регулятор собственного движения

$$u_m = -K \begin{bmatrix} x_m \\ z_m \end{bmatrix} = -[K_1; K_2] \begin{bmatrix} x_m \\ z_m \end{bmatrix}, \quad (3.49)$$

обеспечивающий любые наперед заданные свойства собственного движения замкнутой системы, состоящей из объекта (3.48) и регулятора (3.49).

Этот регулятор синтезируется любым из рассмотренных в п. 3.2 способов. Как видно из (3.49) и рис. 3.9, регулятор собственного

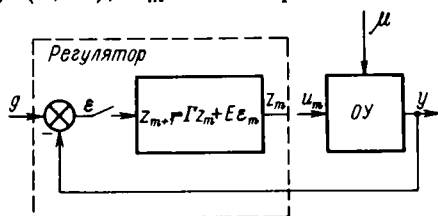


Рис. 3.9. Структурная схема разомкнутой системы с грубым регулятором

движения обеспечивает введение обратных связей по состояниям объекта управления и модели внешнего воздействия.

Уравнение замкнутой системы, состоящей из объекта (3.48) и регулятора (3.49), имеет вид

$$\begin{bmatrix} \dot{x}_{m+1} \\ \dot{z}_{m+1} \end{bmatrix} = \begin{bmatrix} A - BK_1 & -BK_2 \\ -EC & \Gamma \end{bmatrix} \begin{bmatrix} x_m \\ z_m \end{bmatrix} + \begin{bmatrix} D \\ EH_g \end{bmatrix} \xi_m. \quad (3.50)$$

Вычислим вынужденное движение в системе (3.50). Согласно (3.9) установившееся вынужденное движение

$$\begin{bmatrix} x_{вм} \\ z_{вм} \end{bmatrix} = M \xi_m = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \xi_m, \quad (3.51)$$

где матрица M находится как решение уравнения (3.8), которое для системы (3.50) имеет вид

$$M\Gamma - \begin{bmatrix} A - BK_1 & -BK_2 \\ -EC & \Gamma \end{bmatrix} M = \begin{bmatrix} D \\ EH_g \end{bmatrix}.$$

Учитывая введенную в (3.51) блочную структуру матрицы M , перепишем последнее уравнение в виде

$$\begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \Gamma - \begin{bmatrix} A - BK_1 & -BK_2 \\ -EC & \Gamma \end{bmatrix} \times \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} = \begin{bmatrix} D \\ EH_g \end{bmatrix},$$

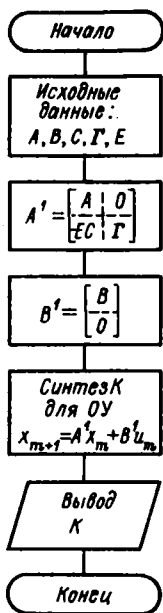


Рис. 3.10. Процедура синтеза грубого регулятора

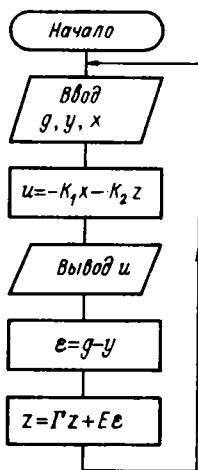


Рис. 3.11. Грубый алгоритм управления вынужденным движением

которое можно представить как систему двух матричных уравнений:

$$\begin{aligned} M_1\Gamma - (A - BK_1)M_1 + BK_2M_2 &= D; \\ M_2\Gamma + ECM_1 - \Gamma M_2 &= EH_g, \end{aligned}$$

откуда имеем

$$M_2\Gamma - \Gamma M_2 = E(H_g - CM_1). \quad (3.52)$$

Решение M_2 уравнения (3.52) существует, так как в системе существуют вынужденные движения как для вектора x_m , так и для вектора z_m . В этом случае имеет место условие

$$E(H_g - CM_1) = 0. \quad (3.53)$$

Заметим, что матрица \mathbf{E} выбирается неоднозначно, единственное ограничение на выбор \mathbf{E} заключается в обеспечении управляемости пары (\mathbf{G}, \mathbf{E}) . Условие (3.53) должно выполняться для различных матриц \mathbf{E} , а это может быть только тогда, когда имеет место уравнение

$$\mathbf{H}_g - \mathbf{C}\mathbf{M}_1 = 0. \quad (3.54)$$

Таким образом, вычисление вынужденного движения в системе (3.50) сводится к решению системы матричных уравнений:

$$\begin{cases} \mathbf{M}_1\mathbf{\Gamma} - (\mathbf{A} - \mathbf{B}\mathbf{K}_1)\mathbf{M}_1 = -\mathbf{B}\mathbf{K}_2\mathbf{M}_2 + \mathbf{D}; \\ \mathbf{M}_2\mathbf{\Gamma} - \mathbf{\Gamma}\mathbf{M}_2 = 0; \\ \mathbf{H}_g - \mathbf{C}\mathbf{M}_1 = 0. \end{cases}$$

В соответствии с (3.51) вынужденное движение ОУ будет вычисляться по формуле $\mathbf{x}_{вм} = \mathbf{M}_1\boldsymbol{\xi}_m$; подставляя это выражение в (3.47), получим установившуюся ошибку системы $\mathbf{e}_m = (\mathbf{H}_g - \mathbf{C}\mathbf{M}_1)\boldsymbol{\xi}_m$, которая, согласно (3.54), равна нулю. Отметим, что условие (3.54) не зависит от матриц \mathbf{A} и \mathbf{B} объекта управления и устанавливается автоматически при любой матрице \mathbf{C} , т. е. регулятор компенсирует соответствующее внешнее воздействие при таких изменениях ОУ, которые не приводят к потере устойчивости системы (3.50). По этой причине подобные регуляторы вынужденного движения называют грубыми.

Процедура синтеза грубого регулятора приведена на рис. 3.10. Исходными данными для алгоритма синтеза являются: матрицы разностных моделей ОУ и внешних воздействий; матрица входа модели воздействия \mathbf{E} , реализуемой регулятором, а также показатели качества собственного движения системы с грубым регулятором (например, в виде времени переходного процесса и перерегулирования). В алгоритме синтеза грубого регулятора используются рассмотренные в п. 3.2 способы синтеза регулятора собственного движения.

Алгоритм управляющей ЭВМ, которая реализует грубый регулятор, изображен на рис. 3.11. Для его реализации не требуется измерение или оценивание вектора состояния КГ.

Если ошибка слежения \mathbf{e} является вектором, то грубые регуляторы необходимо включать в цепь сигнала каждой координаты e_j этого вектора.

3.5. Совместное использование комбинированного и грубого управления

Комбинированное и грубое управления вынужденным движением могут сочетаться различным образом. Пусть к объекту управления (3.33) приложено внешнее воздействие $\boldsymbol{\xi}_m$, которое описывается составным вектором

$$\boldsymbol{\xi}_m = \begin{bmatrix} \xi_{1m} \\ \dots \\ \xi_{2m} \end{bmatrix},$$

а уравнение соответствующего командного генератора имеет вид:

$$\left\{ \begin{array}{l} \begin{bmatrix} \xi_{1m+1} \\ \xi_{2m+1} \end{bmatrix} = \begin{bmatrix} \Gamma_1 & 0 \\ 0 & \Gamma_2 \end{bmatrix} \begin{bmatrix} \xi_{1m} \\ \xi_{2m} \end{bmatrix}; \\ \mu_m = [H_{\mu 1} \mid H_{\mu 2}] \begin{bmatrix} \xi_{1m} \\ \xi_{2m} \end{bmatrix}; \\ g_m = [H_{g1} \mid H_{g2}] \begin{bmatrix} \xi_{1m} \\ \xi_{2m} \end{bmatrix}. \end{array} \right. \quad (3.55)$$

Уравнения (3.33) удобно переписать в виде:

$$\begin{aligned} x_{m+1} &= Ax_m + Bu_m + [D_1 \mid D_2] \begin{bmatrix} \xi_{1m} \\ \xi_{2m} \end{bmatrix}; \\ y_m &= Cx_m. \end{aligned}$$

Спроектируем алгоритм управляющей ЭВМ таким образом, чтобы влияние внешнего воздействия ξ_{1m} на поведение ОУ компенсировалось грубым регулятором, а влияние ξ_{2m} компенсировалось при помощи связей по координатам вектора ξ_{2m} . Для этого, как и в п. 3.4, рассмотрим расширенный объект

$$\begin{bmatrix} x_{m+1} \\ z_{m+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -EC & \Gamma_1 \end{bmatrix} \begin{bmatrix} x_m \\ z_m \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_m + \begin{bmatrix} D_1 & D_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_{1m} \\ \xi_{2m} \end{bmatrix}, \quad (3.56)$$

и для него используем закон управления вида

$$u_m = -[K_1 \mid K_2] \begin{bmatrix} x_m \\ z_m \end{bmatrix} + L\xi_{2m}. \quad (3.57)$$

Матрицу обратных связей $[K_1 \mid K_2]$ в (3.57) рассчитаем таким образом, чтобы обеспечить заданные свойства собственного движения в замкнутой системе, состоящей из объекта (3.56) и регулятора (3.57). Уравнение этой замкнутой системы с учетом (3.55) и (3.47) имеет вид

$$\begin{aligned} \begin{bmatrix} x_{m+1} \\ z_{m+1} \end{bmatrix} &= \begin{bmatrix} A - BK_1 & -BK_2 \\ -EC & \Gamma_1 \end{bmatrix} \begin{bmatrix} x_m \\ z_m \end{bmatrix} + \\ &+ \begin{bmatrix} D_1 & D_2 + BL \\ EH_{g1} & EH_{g2} \end{bmatrix} \begin{bmatrix} \xi_{1m} \\ \xi_{2m} \end{bmatrix}. \end{aligned} \quad (3.58)$$

Вычислим вынужденное движение в системе (3.58). Согласно (3.9) установившееся вынужденное движение

$$\begin{bmatrix} x_{\text{вм}} \\ z_{\text{вм}} \end{bmatrix} = M\xi_{\text{вм}} = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix} \begin{bmatrix} \xi_{1m} \\ \xi_{2m} \end{bmatrix}, \quad (3.59)$$

т. е.

$$x_{вм} = M_1 \xi_{1m} + M_2 \xi_{2m}, \quad z_{вм} = M_3 \xi_{1m} + M_4 \xi_{2m}. \quad (3.60)$$

Матрица M в (3.59) находится в результате решения уравнения (3.8), которое для системы (3.58) с учетом блочной структуры матрицы M имеет вид

$$\begin{aligned} \left[\begin{array}{c|c} M_1 & M_2 \\ \hline M_3 & M_4 \end{array} \right] \left[\begin{array}{c|c} \Gamma_1 & 0 \\ \hline 0 & \Gamma_2 \end{array} \right] - \left[\begin{array}{c|c} A - BK_1 & -BK_2 \\ \hline -EC & \Gamma_1 \end{array} \right] \left[\begin{array}{c|c} M_1 & M_2 \\ \hline M_3 & M_4 \end{array} \right] = \\ = \left[\begin{array}{c|c} D_1 & D_2 + BL \\ \hline EH_{g1} & EH_{g2} \end{array} \right]. \end{aligned}$$

Это уравнение можно представить в виде четырех матричных уравнений:

$$M_1 \Gamma_1 - (A - BK_1) M_1 + BK_2 M_3 = D_1; \quad (3.61)$$

$$M_2 \Gamma_2 - (A - BK_1) M_2 + BK_2 M_4 = D_2 + BL; \quad (3.62)$$

$$M_3 \Gamma_1 + E C M_1 - \Gamma_1 M_3 = E H_{g1}; \quad (3.63)$$

$$M_4 \Gamma_2 + E C M_2 - \Gamma_1 M_4 = E H_{g2}. \quad (3.64)$$

Перепишем уравнение (3.63) в виде

$$M_3 \Gamma_1 - \Gamma_1 M_3 = E (H_{g1} - C M_1),$$

тогда получим (см. п. 3.4):

$$H_{g1} - C M_1 = 0. \quad (3.65)$$

Условие (3.65) выполняется не только независимо от свойств объекта управления, но также независимо и от матрицы L прямых связей по второй составляющей внешнего воздействия, т. е. грубый регулятор, компенсирующий часть внешнего воздействия, можно проектировать, игнорируя вторую составляющую внешнего воздействия.

Преобразуем уравнения (3.62) и (3.64) к стандартному виду уравнения Сильвестра:

$$M_2 \Gamma_2 - (A - BK_1) M_2 = D_2 + BL - BK_2 M_4; \quad (3.66)$$

$$M_4 \Gamma_2 - \Gamma_1 M_4 = E (H_{g2} - C M_2). \quad (3.67)$$

Здесь матрица L неизвестна, при ее изменении меняется матрица M_2 . Будем выбирать L таким образом, чтобы обеспечить выполнения условия

$$H_{g2} - C M_2 = 0, \quad (3.68)$$

тогда уравнение (3.67) примет вид

$$M_4 \Gamma_2 - \Gamma_1 M_4 = 0. \quad (3.69)$$

Положим, что разбиение вектора ξ_m на составляющие ξ_{1m} и ξ_{2m} выполнено таким образом, что спектры матриц Γ_1 и Γ_2 не пересекаются (такое разбиение всегда можно сделать). В этом

случае уравнение (3.69) имеет только нулевое решение $M_4 = 0$ [15]. Тогда (3.66) принимает вид

$$M_3 \Gamma - (A - BK_1) M_3 = D_3 + BL \quad (3.70)$$

и матрица L находится в результате решения системы уравнений (3.68), (3.70).

Эта система не отличается от системы (3.41), (3.42). Таким образом, матрица L не зависит от той части расширенного объекта управления, которая включена последовательно в цепь ошибки до точки приложения управляющего воздействия (см. рис. 3.9). Отметим, что эта часть ОУ не обязательно вводится искусственно; она может существовать и объективно. Такая ситуация имеет место в различных системах сопровождения и слежения, когда при помощи, например, оптических преобразователей информации можно измерять только отклонение регулируемой величины y от входной величины g . И в этом случае, как видно из (3.68), (3.70), матрица прямых связей L определяется только матрицами A , B , D_3 , C , описывающими часть ОУ от точки приложения управляющего воздействия до регулируемой величины. При совместном использовании грубого и комбинированного регуляторов каждый из этих регуляторов можно синтезировать независимо друг от друга. Алгоритм синтеза такого регулятора изображен на рис. 3.12, а алгоритм управляющей ЭВМ, реализующей подобный регулятор, — на рис. 3.13.

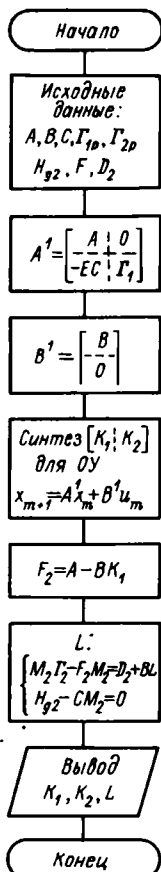


Рис. 3.12. Процедура синтеза регулятора грубого и комбинированного управления

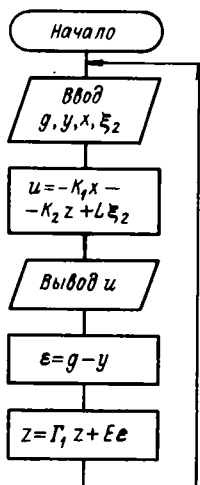


Рис. 3.13. Алгоритм грубого и комбинированного управления

3.6. Синтез комбинированной системы с ненулевой установившейся ошибкой

На практике при проектировании систем управления часто бывает достаточно обеспечить установившуюся ошибку, не превосходящую заданного значения ϵy . Ненулевая установившаяся ошибка получается в том случае, когда вводятся прямые связи

не по всем координатам вектора состояния ξ_m командного генератора внешнего воздействия. Рассмотрим способ, позволяющий определить те составляющие вектора ξ_m , по которым необходимо вводить прямые связи, для обеспечения значения установившейся ошибки не более ε^y .

Будем считать, что в уравнении (3.34)

$$\xi_m = \begin{bmatrix} \xi_{1m} \\ \dots \\ \xi_{2m} \\ \dots \\ \dots \\ \dots \\ \xi_{pm} \end{bmatrix}; \quad \Gamma = \begin{bmatrix} \Gamma_1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \Gamma_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Gamma_p \end{bmatrix};$$

$$H_g = [H_1 \vdots \dots \vdots H_p]. \quad (3.71)$$

Такое представление ξ_m согласуется с его разложением по базисным функциям в линейном функциональном пространстве внешних воздействий. Некоторые из матриц Γ_i в (3.71) или вся матрица Γ могут быть жордановыми клетками. Матрицы H_i , которые соответствуют возмущающим компонентам ξ_{im} , равны нулю.

Матрица D в уравнении (3.33) в соответствии с (3.71) представляется следующим образом:

$$D = [D_1 \vdots \dots \vdots D_p]. \quad (3.72)$$

Матрицу прямых связей L в алгоритме управления (3.39) в соответствии с (3.71) представим в виде

$$D = [\alpha_1 L_1 \vdots \dots \vdots \alpha_p L_p], \quad (3.73)$$

где α_i — булевские переменные (если $\alpha_i = 0$, то прямая связь по компоненте ξ_{im} вектора ξ_m не вводится, если же $\alpha_i = 1$, то соответствующая связь вводится посредством матрицы L_i).

Матрицу M , являющуюся решением (3.41) и определяющую вынужденное движение, а следовательно, и установившуюся ошибку, согласно (3.71) представим в виде

$$M = [M_1 \vdots \dots \vdots M_p]. \quad (3.74)$$

Система матричных уравнений (3.41), (3.42), определяющая матрицы M и L , на основании (3.71)—(3.74) принимает вид:

$$[M_1 \vdots \dots \vdots M_p] \begin{bmatrix} \Gamma_1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Gamma_p \end{bmatrix} - A_c [M_1 \vdots \dots \vdots M_p] =$$

$$= B [\alpha_1 L_1 \vdots \dots \vdots \alpha_p L_p] + [D_1 \vdots \dots \vdots D_p];$$

$$[\alpha_1 H_1 \vdots \dots \vdots \alpha_p H_p] - C [\alpha_1 M_1 \vdots \dots \vdots \alpha_p M_p] = 0;$$

откуда получаем систему следующих матричных уравнений:

$$\begin{cases} \mathbf{M}_i \Gamma_i - \mathbf{A}_c \mathbf{M}_i = \alpha_i \mathbf{B} \mathbf{L}_i + \mathbf{D}_i; \\ \alpha_i (\mathbf{H}_i - \mathbf{C} \mathbf{M}_i) = 0, \quad i = \overline{1, p}. \end{cases} \quad (3.75)$$

При этом i -я составляющая ошибки, обуславливаемая i -й компонентой ξ_{im} вектора внешнего воздействия, имеет вид

$$e_{im} = \bar{\alpha}_i (\mathbf{H}_i - \mathbf{C} \mathbf{M}_i) \xi_{im}, \quad (3.76)$$

а полная ошибка

$$e_m = \sum_{i=1}^p \bar{\alpha}_i (\mathbf{H}_i - \mathbf{C} \mathbf{M}_i) \xi_{im}. \quad (3.77)$$

Уравнения системы (3.75) независимы, поэтому каждую пару уравнений можно решать отдельно. Матрицы \mathbf{M}_i принимают два значения в зависимости от значения булевой переменной α_i . Если положить $\alpha_i = 0$ для $\forall i \in \overline{1, p}$, то составляющие ошибки находятся согласно (3.76), а матрица \mathbf{M}_i получается в результате решения первого из уравнений (3.75). Полная ошибка в установившемся режиме вычисляется в соответствии с (3.77), а ее норма оценивается неравенством

$$\|e_m\| \leq \sum_{i=1}^p \|(\mathbf{H}_i - \mathbf{C} \mathbf{M}_i) \xi_{im}\|.$$

Если окажется, что $\|e_m\| \geq \epsilon$, то без комбинированных связей требуемую точность ϵ в установленном режиме обеспечить нельзя. Выбор компонентов ξ_{im} вектора ξ_m , по которым целесообразно

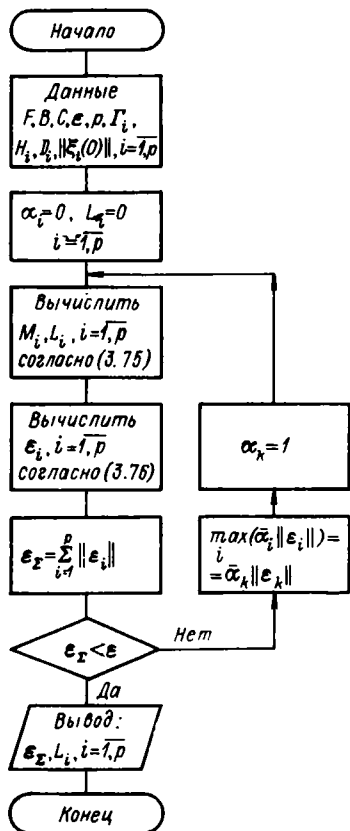


Рис. 3.14. Процедура синтеза инвариантной до ϵ комбинированной системы

вводить комбинированные связи, осуществляется в соответствии с алгоритмом, изображенным на рис. 3.14. Алгоритм основан на том, что в системе (3.75) последовательно придается значение 1 той булевой переменной, которая соответствует максимальной норме $\|e_{im}\|$, а затем проверяется условие $\|e_m\| < \epsilon$. Если при выбранных $\alpha_i = 1$ условие не выполняется, то среди оставшихся составляющих ошибки выбирается максимальная по норме, и соответствующей булевой переменной придается значение 1. Таким образом поступаем до тех пор, пока не получим заданную точность в установленном режиме.

Отметим, что если Γ_i является верхнетреугольной матрицей, например жордановой, то для вычисления M_i можно использовать итеративную процедуру. При этом вычисляются компоненты матрицы M_i малой размерности, что сокращает трудоемкость алгоритма вычислений. Действительно, пусть

$$\Gamma_i = \begin{bmatrix} \Gamma_{11}^i & \Gamma_{12}^i & \dots & \Gamma_{1q}^i \\ \dots & \dots & \dots & \dots \\ 0 & \Gamma_{22}^i & \dots & \Gamma_{2q}^i \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Gamma_{qq}^i \end{bmatrix},$$

в соответствии с этим представим

$$\xi_{im} = \begin{bmatrix} \xi_{1m}^i \\ \dots \\ \xi_{qm}^i \end{bmatrix}; \quad M_i = [M_1^i \mid \dots \mid M_q^i];$$

$$N_i = [N_1^i \mid \dots \mid N_q^i]; \quad D_i = [D_1^i \mid \dots \mid D_q^i].$$

При этом пара матричных уравнений (3.75) имеет вид:

$$[M_1^i \mid \dots \mid M_q^i] \begin{bmatrix} \Gamma_{11}^i & \Gamma_{12}^i & \dots & \Gamma_{1q}^i \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Gamma_{qq}^i \end{bmatrix} - A_c [M_1^i \mid \dots \mid M_q^i] =$$

$$= B [\alpha_1^i L_1^i \mid \dots \mid \alpha_q^i L_q^i] + [D_1^i \mid \dots \mid D_q^i];$$

$$[\alpha_1^i N_1^i \mid \dots \mid \alpha_q^i N_q^i] - C [\alpha_1^i M_1^i \mid \dots \mid \alpha_q^i M_q^i] = 0.$$

Этой паре уравнений соответствует система матричных уравнений:

$$M_1^i \Gamma_{11}^i - A_c M_1^i = \alpha_1^i B L_1^i + D_1^i;$$

$$\alpha_1^i (N_1^i - C M_1^i) = 0;$$

$$M_1^i \Gamma_{12}^i + M_2^i \Gamma_{22}^i - A_c M_2^i = \alpha_2^i B L_2^i + D_2^i;$$

$$\alpha_2^i (N_2^i - C M_2^i) = 0;$$

$$\dots \dots \dots$$

$$M_1^i \Gamma_{1q}^i + M_2^i \Gamma_{2q}^i + \dots + M_q^i \Gamma_{qq}^i - A_c M_q^i = \alpha_q^i B L_q^i + D_q^i;$$

$$\alpha_q^i (N_q^i - C M_q^i) = 0,$$

причем

$$e_{jm}^i = \bar{\alpha}_j^i (N_j^i - C M_j^i) \xi_{jm}^i; \quad j = \overline{1, q}.$$

В этом случае матрицы связей L_j^i также можно вычислять последовательно, как и в основном алгоритме на рис. 3.14.

Принцип компенсации установившейся ошибки только до заданного уровня, а не до нулевого значения, распространяется также и на грубые регуляторы вынужденного движения. Алгоритм синтеза грубых регуляторов вынужденного движения, обеспечивающих инвариантность системы до ϵ , похож по своей идее на рассмотренный алгоритм.

3.7. Алгоритмы оценивания состояний

Управление объектом (3.33), который возбуждается внешним воздействием (3.34), осуществляется регулятором, работающим в соответствии с алгоритмом (3.39). При этом обеспечиваются требуемые свойства собственного движения и нулевая установившаяся ошибка. Для реализации алгоритма (3.39) необходимо измерять все координаты вектора состояния x_m ОУ и вектора состояния ξ_m КГ внешнего воздействия. Эти измерения осуществляются датчиками, установленными на объекте управления. Часто на практике их число в силу различных технических и экономических причин недостаточно для измерения всех сигналов, необходимых для реализации алгоритма (3.39). Широко распространены также случаи, когда выходные сигналы измерительных устройств содержат высокий уровень шумов. В обеих названных ситуациях используются специальные алгоритмы оценивания неизмеряемых состояний, которые строятся на основе понятия наблюдающих устройств или наблюдателей [4, 24, 32, 38, 83]. Для построения алгоритмов оценивания состояний объединим динамические системы (3.33), (3.34) и будем рассматривать расширенную модель:

$$\begin{cases} x_{m+1}^o = A^o x_m^o + B^o u_m; \\ y_m^x = C^o x_m^o, \end{cases} \quad (3.78)$$

где

$$x_m^o = \begin{bmatrix} x_m \\ \dots \\ \xi_m \end{bmatrix}; \quad A^o = \begin{bmatrix} A & D \\ \dots & \dots \\ 0 & \Gamma \end{bmatrix}; \quad B^o = \begin{bmatrix} B \\ \dots \\ 0 \end{bmatrix}$$

(вектор измерений y_m^x образуется выходными сигналами измерительных устройств, установленных на объекте, а матрица C^o определяется перечнем этих устройств).

Для динамической системы (3.78) могут быть построены наблюдающие устройства полной и пониженной размерности. Наблюдатели полной размерности используются в том случае, когда сигналы измерителей имеют высокий уровень шумов. Наблюдатели пониженной размерности строятся в случае использования датчиков с малым уровнем шумов, при этом для целей управления применяются как выходные сигналы датчиков, так и сигналы наблюдающего устройства.

Наблюдатели полной размерности. Алгоритм оценивания полного вектора \mathbf{x}_m^o системы (3.78) строится как модель этой системы, на вход которой дополнительно подается вектор невязки, представляющей собой разность векторов выходных сигналов системы \mathbf{y}_m^n и ее модели $\widehat{\mathbf{y}}_m^n$, т. е.

$$\widehat{\mathbf{x}}_{m+1}^o = \mathbf{A}^o \widehat{\mathbf{x}}_m^o + \mathbf{B}^o \mathbf{u}_m + \mathbf{K}_n (\mathbf{y}_m^n - \widehat{\mathbf{y}}_m^n). \quad (3.79)$$

Здесь $\widehat{\mathbf{x}}_m^o$ — вектор оценки вектора \mathbf{x}_m^o ; \mathbf{K}_n — матрица коэффициентов усиления по невязке; вектор $\widehat{\mathbf{y}}_m^n$ вычисляется по формуле

$$\widehat{\mathbf{y}}_m^n = \mathbf{C}^o \widehat{\mathbf{x}}_m^o. \quad (3.80)$$

Алгоритм (3.79) должен быть построен таким образом, чтобы ошибка оценивания

$$\widetilde{\mathbf{x}}_m = \widehat{\mathbf{x}}_m^o - \mathbf{x}_m^o \quad (3.81)$$

с течением времени стремилась к нулю.

Для выяснения условий, при которых выполняется это требование, составим разностное уравнение ошибки. Из (3.81) следует

$$\widetilde{\mathbf{x}}_{m+1} = \widehat{\mathbf{x}}_{m+1}^o - \mathbf{x}_{m+1}^o;$$

используя (3.78)—(3.80), получим

$$\widetilde{\mathbf{x}}_{m+1} = \mathbf{A}^o (\widehat{\mathbf{x}}_m^o - \mathbf{x}_m^o) + \mathbf{K}_n \mathbf{C}^o (\mathbf{x}_m^o - \widehat{\mathbf{x}}_m^o).$$

Подставив в это выражение (3.81), окончательно получим

$$\widetilde{\mathbf{x}}_{m+1} = (\mathbf{A}^o - \mathbf{K}_n \mathbf{C}^o) \widetilde{\mathbf{x}}_m,$$

т. е. ошибка $\widetilde{\mathbf{x}}_m$ стремится к нулю, если все собственные числа матрицы $(\mathbf{A}^o - \mathbf{K}_n \mathbf{C}^o)$ по модулю меньше единицы.

Расположение собственных чисел этой матрицы внутри единичного круга определяет динамику стремления ошибки $\widetilde{\mathbf{x}}_m$ к нулю. Синтез алгоритма оценки вектора \mathbf{x}^o сводится к такому выбору матрицы невязки \mathbf{K}_n , который обеспечивает наперед заданную динамику изменения $\widetilde{\mathbf{x}}_m$. Поэтому вместо матрицы $(\mathbf{A}^o - \mathbf{K}_n \mathbf{C}^o)$ будем рассматривать матрицу $(\mathbf{A}^{oT} - \mathbf{C}^{oT} \mathbf{K}_n^T)$; известно, что собственные числа этих матриц совпадают [15]. Для матрицы $(\mathbf{A}^{oT} - \mathbf{C}^{oT} \mathbf{K}_n^T)$ синтез \mathbf{K}_n^T можно рассматривать как синтез регулятора собственного движения для ОУ, имеющего матрицы состояния \mathbf{A}^{oT} и управления \mathbf{C}^{oT} .

Вектор оценки $\widehat{\mathbf{x}}_m^o$ используется для реализации закона управления (3.39), который преобразуется к виду (рис. 3.15):

$$\begin{cases} \mathbf{u}_m = -\mathbf{K} \widehat{\mathbf{x}}_m + \mathbf{L} \xi_m; \\ \widehat{\mathbf{x}}_{m+1}^o = \mathbf{A}^o \widehat{\mathbf{x}}_m^o + \mathbf{B}^o \mathbf{u}_m + \mathbf{K}_n (\mathbf{y}_m^n - \widehat{\mathbf{y}}_m^n). \end{cases} \quad (3.82)$$

Алгоритм наблюдающего устройства полной размерности (3.79) совпадает с алгоритмом фильтра Калмана [28, 45]. Различие между ними заключается в том, что для фильтра Калмана матрица коэффициентов невязки K_n рассчитывается по специальной процедуре, обеспечивающей минимум нормы ошибки оценивания при известных статистических характеристиках сигналов и шумов, а для алгоритма (3.79) эта матрица постоянна.

Уравнения объекта (3.33), управляемого регулятором (3.82), имеют вид:

$$\begin{cases} \mathbf{x}_{m+1} = \mathbf{A}\mathbf{x}_m + \mathbf{B}\mathbf{u}_m + \mathbf{D}\xi_m; \\ \mathbf{u}_m = [-K \mid L] \hat{\mathbf{x}}_m^o; \\ \hat{\mathbf{x}}_{m+1}^o = \mathbf{A}^o \hat{\mathbf{x}}_m^o + \mathbf{B}^o \mathbf{u}_m + K_n (\mathbf{y}_m^n - \mathbf{C}^o \hat{\mathbf{x}}_m^o). \end{cases}$$

Для выявления влияния наблюдающего устройства на свойства системы перейдем к новому базису в пространстве состояния замкнутой системы, введя замену переменных по формуле (3.81), в результате получим:

$$\begin{aligned} \mathbf{x}_{m+1} &= \mathbf{A}\mathbf{x}_m + \mathbf{B}[-K \mid L](\mathbf{x}_m^o + \tilde{\mathbf{x}}_m) + \mathbf{D}\xi_m; \\ \tilde{\mathbf{x}}_{m+1} &= \mathbf{A}^o(\mathbf{x}_m^o + \tilde{\mathbf{x}}_m) + \mathbf{B}^o[-K \mid L](\mathbf{x}_m^o + \tilde{\mathbf{x}}_m) - \\ &\quad - K_n \mathbf{C}^o \tilde{\mathbf{x}}_m - \mathbf{x}_{m+1}^o. \end{aligned}$$

Воспользовавшись (3.78), получим после алгебраических преобразований

$$\begin{cases} \mathbf{x}_{m+1} = \mathbf{A}\mathbf{x}_m + \mathbf{B}[-K \mid L]\mathbf{x}_m^o + \mathbf{D}\xi_m + \mathbf{B}[-K \mid L]\tilde{\mathbf{x}}_m; \\ \tilde{\mathbf{x}}_{m+1} = (\mathbf{A}^o - K_n \mathbf{C}^o)\tilde{\mathbf{x}}_m \end{cases}$$

или, учитывая структуру вектора $\tilde{\mathbf{x}}_m^o$,

$$\begin{cases} \mathbf{x}_{m+1} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}_m + (\mathbf{B}\mathbf{L} + \mathbf{D})\xi_m + \mathbf{B}[-K \mid L]\tilde{\mathbf{x}}_m; \\ \tilde{\mathbf{x}}_{m+1} = (\mathbf{A}^o - K_n \mathbf{C}^o)\tilde{\mathbf{x}}_m. \end{cases} \quad (3.83)$$

Сравнение (3.40) и (3.83) показывает, что наблюдатель полной размерности влияет на переходные процессы этих систем. После окончания переходных процессов, когда ошибка оценивания $\tilde{\mathbf{x}}_m$ стремится к нулю, поведение систем (3.40) и (3.83) различаться не будет.

Собственные числа системы (3.83), определяющие переходный процесс, являются объединением собственных чисел системы (3.40) без наблюдателя и чисел матрицы $(\mathbf{A}^o - K_n \mathbf{C}^o)$ наблюдателя полной размерности. Таким образом, изменяя собственные

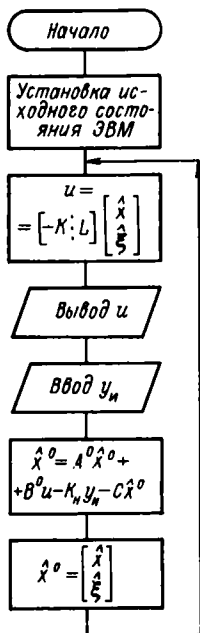


Рис. 3.15. Алгоритм комбинированного управления с наблюдателем

числа матрицы состояния наблюдателя, можно изменять степень влияния наблюдателя на переходные процессы в системе.

Использование наблюдателей полной размерности в значительной степени увеличивает загрузку управляющей ЭВМ из-за необходимости вычисления компонент вектора \hat{x}_m^o высокой размерности. Поэтому предпочтительнее оценивать только часть компонент вектора x_m^o . Такая возможность представляется в том случае, когда выходные сигналы измерителей имеют малый уровень шумов и эти сигналы можно использовать без предварительной обработки для реализации алгоритмов управления. При этом недостающие (неизмеряемые) координаты векторов x_m , ξ_m оцениваются при помощи редуцированных наблюдателей.

Наблюдатели пониженной размерности. Один из возможных методов синтеза дискретных наблюдателей заключается в следующем. Как и ранее, полагаем, что для объекта управления (3.33), возбуждаемого внешним воздействием (3.34), необходимо использовать алгоритм управления (3.39), но при помощи преобразователей информации измеряются и могут быть введены в ЭВМ только сигналы, образующие вектор измерения

$$y_m^n = C^o x_m^o, \quad (3.84)$$

где C^o и x_m^o имеют тот же смысл, что и в (3.78), т. е. измеряется линейная комбинация векторов x_m и ξ_m .

Линейная комбинация

$$w_m = Mx_m^o \quad (3.85)$$

этих векторов не доступна для измерения, причем $\dim w_m = \dim x_m^o - \dim y_m^n$.

Преобразуем (3.39) таким образом, чтобы в явном виде выделить измеряемые и неизмеряемые координаты векторов x_m и ξ_m ,

$$u_m = -Kx_m + L\xi_m = N_1 y_m^n + N_2 w_m. \quad (3.86)$$

Подставляя (3.84) и (3.85) в (3.86), получим

$$\{ -K \mid L \} \begin{bmatrix} x_m \\ \xi_m \end{bmatrix} = [N_1 C^o + N_2 M] x_m^o,$$

но

$$x_m^o = \begin{bmatrix} x_m \\ \xi_m \end{bmatrix},$$

поэтому должно иметь место равенство

$$\{ -K \mid L \} = N_1 C^o + N_2 M,$$

которое на основании правила перемножения блочных матриц записывается в виде

$$\{ -K \mid L \} = [N_1 \mid N_2] \begin{bmatrix} C^o \\ M \end{bmatrix},$$

откуда

$$[N_1 \ ; \ N_2] = [-K \ ; \ L] \left[\frac{C^0}{M} \right]^{-1}. \quad (3.87)$$

Полученное выражение определяет правило выбора матриц N_1 , N_2 , при которых обеспечивается равенство (3.86).

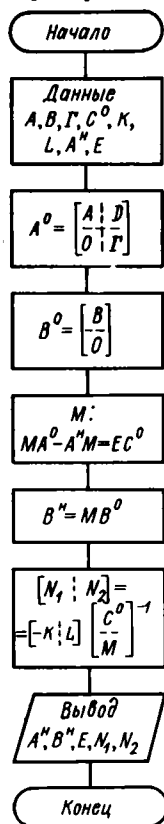


Рис. 3.16. Процедура синтеза регулятора с наблюдателем пониженной размерности

В алгоритме (3.86) u_m^n есть физически существующий выход преобразователей информации, в то время как w_m физически не измеряется. По аналогии с использованием наблюдателей полной размерности вместо алгоритма (3.86) будем реализовывать алгоритм

$$u_m = N_1 y_m^n + N_2 \hat{w}_m, \quad (3.88)$$

где \hat{w}_m — выход наблюдающего устройства.

Примем алгоритм функционирования этого устройства в виде

$$\hat{w}_{m+1} = A_n \hat{w}_m + B_n u_m + E y_m^n \quad (3.89)$$

и выберем матрицы A_n , B_n и E таким образом, чтобы ошибка оценивания

$$\tilde{w}_m = \hat{w}_m - w_m \quad (3.90)$$

с течением времени стремилась к нулю.

Для этого составим разностное уравнение ошибки. Из (3.90) получим выражение $\tilde{w}_{m+1} = \hat{w}_{1+m} - w_{m+1}$, которое с учетом (3.78), (3.84), (3.85) и (3.89) преобразуется к виду

$$\begin{aligned} \tilde{w}_{m+1} = & A_n \tilde{w}_m + B_n u_m + \\ & + E C^0 x_m^0 - M A^0 x_m^0 - M B^0 u_m. \end{aligned} \quad (3.91)$$

Потребуем выполнения равенства

$$B_n = M B^0 \quad (3.92)$$

и исключим из (3.91) \hat{w}_m в соответствии с (3.90), тогда получим

$$\tilde{w}_{m+1} = A_n \tilde{w}_m + A_n M x_m^0 + E C^0 x_m^0 - M A^0 x_m^0.$$

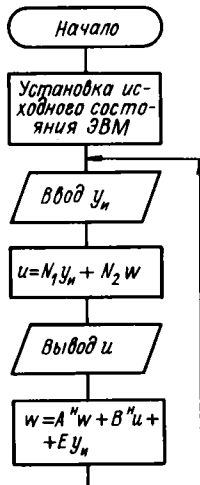


Рис. 3.17. Алгоритм комбинированного управления с наблюдателем пониженной размерности

Потребуем теперь выполнения равенства

$$MA^{\circ} - A_n M = EC^{\circ}, \quad (3.93)$$

тогда разностное уравнение ошибки примет вид $\tilde{w}_{m+1} = A_n \tilde{w}_m$; эта ошибка с течением времени стремится к нулю, если все собственные числа матрицы A_n по модулю меньше единицы.

Выбором матрицы A_n обеспечивается желаемая динамика стремления ошибки к нулю. Матрица E в (3.89) выбирается такой, чтобы пара матриц (A_n, E) была управляемой. Алгоритм синтеза цифрового регулятора с наблюдателем пониженной размерности имеет вид, изображенный на рис. 3.16. Алгоритм управляющей ЭВМ, реализующей цифровой комбинированный регулятор с наблюдателем пониженной размерности, изображен на рис. 3.17.

Уравнение замкнутой системы, состоящей из ОУ (3.33) и регулятора, алгоритм которого изображен на рис. 3.16, имеет вид:

$$\begin{cases} x_{m+1} = Ax_m + Bu_m + D\xi_m; \\ u_m = N_1 y_m^n + N_2 \hat{w}_m; \\ \hat{w}_{m+1} = A_n \hat{w}_m + B_n u_m + E y_m^n. \end{cases} \quad (3.94)$$

Внешнее воздействие, приложенное к системе (3.94), описывается уравнением (3.34).

Рассмотрим динамическую систему, состоящую из замкнутой системы (3.94) и командного генератора (3.34):

$$\begin{cases} x_{m+1}^{\circ} = A^{\circ} x_m^{\circ} + B^{\circ} u_m; \\ u_m = N_1 y_m^n + N_2 \hat{w}_m; \\ \hat{w}_{m+1} = A_n \hat{w}_m + B_n u_m + E y_m^n, \end{cases}$$

где x_m° , A° , B° и y_m^n имеют тот же смысл, что и в (3.76).

Исключив управляющее воздействие u_m из этих уравнений, получим

$$\begin{bmatrix} x_{m+1}^{\circ} \\ \hat{w}_{m+1} \end{bmatrix} = \begin{bmatrix} A^{\circ} + B^{\circ} N_1 C^{\circ} & B^{\circ} N_2 \\ (B_n N_1 + E) C^{\circ} & A_n + B_n N_2 \end{bmatrix} \begin{bmatrix} x_m^{\circ} \\ \hat{w}_m \end{bmatrix}. \quad (3.95)$$

Перейдем в (3.95) к новому базису, введя ошибку наблюдающего устройства (3.90), при этом матрица перехода от вектора $\begin{bmatrix} x_m^{\circ} \\ \hat{w}_m \end{bmatrix}$ к вектору $\begin{bmatrix} x_m^{\circ} \\ \tilde{w}_m \end{bmatrix}$ имеет вид $\begin{bmatrix} I & 0 \\ -M & I \end{bmatrix}$, где M — решение уравнения (3.93), а I — единичная матрица соответствующей размерности. Используя известные правила перехода к новому

базису [4], после алгебраических преобразований с учетом (3.92) и (3.93) получим

$$\begin{bmatrix} \mathbf{x}_{m+1}^{\circ} \\ \tilde{\mathbf{w}}_{m+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{\circ} + \mathbf{B}^{\circ} \mathbf{N}_1 \mathbf{C}^{\circ} + \mathbf{B}^{\circ} \mathbf{N}_2 \mathbf{M} & \mathbf{B}^{\circ} \mathbf{N}_2 \\ 0 & \mathbf{A}_n \end{bmatrix} \begin{bmatrix} \mathbf{x}_m^{\circ} \\ \tilde{\mathbf{w}}_m \end{bmatrix}.$$

Принимая во внимание (3.78) и (3.86), получим:

$$\begin{cases} \mathbf{x}_{m+1} = (\mathbf{A} - \mathbf{BK}) \mathbf{x}_m + (\mathbf{BL} + \mathbf{D}) \xi_m + \mathbf{BN}_2 \tilde{\mathbf{w}}_m; \\ \xi_{m+1} = \Gamma \xi_m; \\ \tilde{\mathbf{w}}_{m+1} = \mathbf{A}_n \tilde{\mathbf{w}}_m. \end{cases} \quad (3.96)$$

Сравнивая (3.96) с (3.40), видим, что за счет наблюдателя пониженной размерности появляется дополнительная ошибка $\tilde{\mathbf{w}}_m$, искажающая изменение вектора \mathbf{x}_m . Однако величина $\tilde{\mathbf{w}}_m$ с течением времени стремится к нулю, откуда следует, что установившаяся ошибка системы (3.96) будет совпадать с установившейся ошибкой системы (3.40), т. е. будет равна нулю. Таким образом, введение наблюдателя пониженной размерности, как и наблюдателя полной размерности, не нарушает свойства инвариантности системы, обеспечиваемого комбинированным управлением.

Алгоритм ускоренного оценивания состояний. Оценка вектора неизмеряемых координат в алгоритме на рис. 3.17 осуществляется в конце каждого такта работы управляющей ЭВМ и используется для расчета управляющего воздействия только в следующем такте. Рассмотрим алгоритм синтеза наблюдателя пониженной размерности, обеспечивающего более оперативное использование оценки вектора неизмеряемых координат. Будем считать, что в уравнении объекта (3.78) матрица выхода $\mathbf{C}^{\circ} = [\mathbf{I} \ ; \ \mathbf{0}]$. Уравнение (3.78) всегда можно привести к такому виду преобразованием подобия, если пара матриц $(\mathbf{A}^{\circ}, \mathbf{C}^{\circ})$ полностью наблюдаема. Тогда вектор состояния представляется в виде

$$\mathbf{x}_m^{\circ} = \begin{bmatrix} \mathbf{y}_m^n \\ \mathbf{w}_m \end{bmatrix} = \begin{bmatrix} \mathbf{C}^{\circ} \\ \tilde{\mathbf{T}} \end{bmatrix} \mathbf{x}_m, \quad (3.97)$$

где \mathbf{y}_m^n — вектор измеряемых, а \mathbf{w}_m — неизмеряемых координат вектора \mathbf{x}_m° (при таком представлении вектора \mathbf{x}_m° матрица $\tilde{\mathbf{T}}$ имеет вид $\tilde{\mathbf{T}} = [\mathbf{0} \ ; \ \mathbf{I}]$).

Представим \mathbf{u}_m следующим образом:

$$\mathbf{u}_m = [-\mathbf{K} \ ; \ \mathbf{L}] \mathbf{x}_m^{\circ} = \mathbf{N}_1 \mathbf{y}_m^n + \mathbf{N}_2 \mathbf{w}_m \quad (3.98)$$

или, учитывая (3.97),

$$[-\mathbf{K} \ ; \ \mathbf{L}] = \mathbf{N}_1 \mathbf{C}^{\circ} + \mathbf{N}_2 \tilde{\mathbf{T}},$$

откуда

$$[\mathbf{N}_1 \ ; \ \mathbf{N}_2] = [-\mathbf{K} \ ; \ \mathbf{L}] \begin{bmatrix} \mathbf{C}^{\circ} \\ \tilde{\mathbf{T}} \end{bmatrix}^{-1}.$$

Матрица $\begin{bmatrix} C^0 \\ \dots \\ T \end{bmatrix}$ унитарная, поэтому

$$[N_1 | N_2] = [-K | L] \begin{bmatrix} C^0 \\ \dots \\ T \end{bmatrix}.$$

Полученное выражение определяет правило выбора матриц N_1 и N_2 .

Поскольку w_m не измеряется, то для реализации управления (3.98), как и ранее, вместо w_m используем оценку этого вектора, которую будем строить следующим образом. В соответствии с разбиением (3.97) представим уравнение (3.78) в виде

$$\begin{bmatrix} y_{m+1}^n \\ w_{m+1} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y_m^n \\ w_m \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u_m$$

или

$$\begin{cases} y_{m+1}^n = A_{11}y_m^n + A_{12}w_m + B_1u_m; \\ w_{m+1} = A_{22}w_m + A_{21}y_m^n + B_2u_m. \end{cases} \quad (3.99)$$

Из первого уравнения системы (3.99) имеем

$$A_{12}w_m = y_{m+1}^n - A_{11}y_m^n - B_1u_m. \quad (3.100)$$

На основании второго уравнения системы (3.99) будем строить наблюдатель пониженной размерности в виде

$$\hat{w}_{m+1} = A_{22}\hat{w}_m + A_{21}y_m^n + B_2u_m + K_n(A_{12}w_m - A_{12}\hat{w}_m)$$

или, учитывая (3.100),

$$\hat{w}_{m+1} = A_{22}\hat{w}_m + A_{21}y_m^n + B_2u_m + K(y_{m+1}^n - A_{11}y_m^n - B_1u_m - A_{12}\hat{w}_m). \quad (3.101)$$

Рассмотрим ошибку этого наблюдателя

$$\tilde{w}_m = \hat{w}_m - w_m.$$

Используя (3.100) и второе уравнение (3.99), после алгебраических преобразований получим

$$\tilde{w}_{m+1} = (A_{22} - K_n A_{12}) \tilde{w}_m.$$

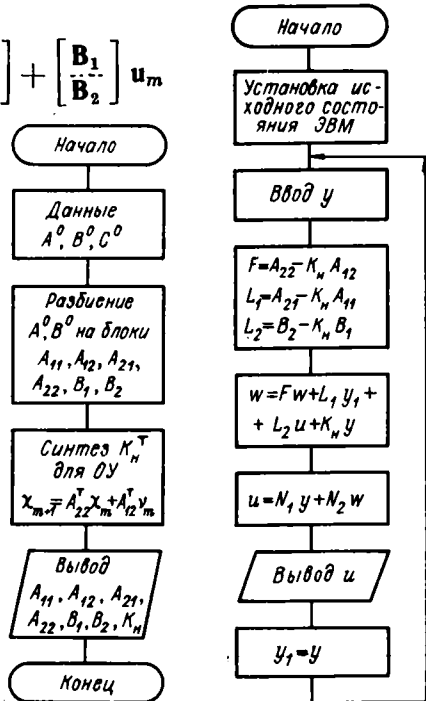


Рис. 3.18. Процедура синтеза наблюдателя пониженной размерности

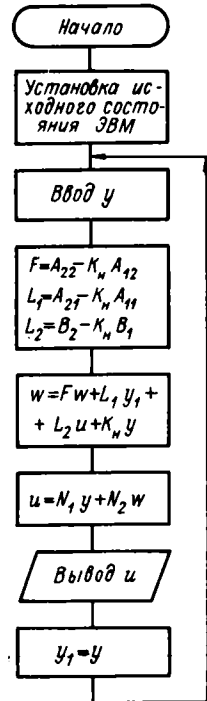


Рис. 3.19. Алгоритм комбинированного управления с ускоренным наблюдателем пониженной размерности

При надлежащем выборе матрицы K_n обеспечивается асимптотическое стремление \tilde{w}_m к нулю, т. е. наблюдатель (3.99) осуществляет оценку неизменяемого вектора w_m . На основании сказанного алгоритм синтеза наблюдателя пониженной размерности (3.101) имеет вид, изображенный на рис. 3.18. Матрица K_n^T в этом алгоритме синтезируется как матрица обратных связей регулятора собственного движения для объекта с уравнением

$$\chi_{m+1} = A_{22}^T \chi_m + A_{12}^T v_m,$$

причем $v_m = -K_n^T \chi_m$.

Синтез матрицы K_n^T осуществляется одним из методов, описанных в п. 3.2. На рис. 3.19 изображен алгоритм управляющей ЭВМ, реализующий регулятор с наблюдателем пониженной размерности. Этот алгоритм составлен на основании выражений (3.98) и (3.100).

3.8. Специальные алгоритмы оценивания состояний

В системах управления встречаются ситуации, когда управление объектом (3.78) осуществляется с интервалом T , а измерение вектора y_m^n — с интервалом $T_1 > T$. Интервал T определяется требуемым качеством процессов в управляемом объекте, а интервал T_1 — принципом действия измерительных преобразователей. Такая ситуация может иметь место, например, в цифровых системах радиолокации, светолокации, гидролокации, в системах гидрирования с накоплением и т. д.

В любом случае при различных интервалах управления и измерения для выработки управляющего воздействия с тактом T необходимо построить наблюдатель объекта (3.78), работающий с тактом T по сигналам измерения, которые поступают с тактом T_1 . Положим, что $T_1 = kT$ ($k > 1$ — целое число). Тогда уравнения (3.78) для объекта можно переписать в виде

$$\begin{cases} x^\circ(mT + T) = A^\circ x^\circ(mT) + B^\circ u(mT); \\ y^n(nT_1) = C^\circ x^\circ(nT_1), \end{cases} \quad (3.102)$$

где $m = 0, 1, 2, \dots$; $n = \left[\frac{m}{k} \right]$ — целая часть числа m/k .

Для восстановления $x^\circ(mT)$ с интервалом T по измерениям $y^n(nT_1)$ используем наблюдатель полной размерности, алгоритм функционирования которого опишем уравнением

$$\hat{x}^\circ(mT + T) = A^\circ \hat{x}^\circ(mT) + B^\circ u(mT) + K_n v(mT), \quad (3.103)$$

где невязка $v(mT)$ определяется следующим образом:

$$v(mT) = \begin{cases} y^n(mT) - C \hat{x}^\circ(mT) & \text{отсчет } y^n(mT) \text{ есть;} \\ 0 & \text{отсчет } y^n(mT) \text{ отсутствует.} \end{cases}$$

Выберем матрицу K_n в (3.103) таким образом, чтобы ошибка

$$\tilde{x}(nT_1) = x^\circ(nT_1) - \hat{x}^\circ(nT_1)$$

стремила к нулю, при $n \rightarrow \infty$.

Эту ошибку рассматриваем только в дискретные моменты времени подачи корректирующего воздействия на идентификатор (3.103). Для решения поставленной задачи получим разностное уравнение ошибки

$$\tilde{x}(nT_1 + T_1) = x^\circ(nT_1 + T_1) - \hat{x}^\circ(nT_1 + T_1).$$

Переходя в последнем выражении от интервала T_1 к интервалу T , получим

$$\tilde{x}(mT + kT) = x^\circ(mT + kT) - \hat{x}^\circ(mT + kT). \quad (3.104)$$

Из уравнений (3.102) и (3.103) получим соответственно:

$$\begin{aligned} x^\circ(mT + kT) &= (A^\circ)^k x^\circ(mT) + \sum_{i=0}^{k-1} (A^\circ)^{k-1-i} B^\circ u(mT + iT); \\ \hat{x}^\circ(mT + kT) &= [(A^\circ)^k - (A^\circ)^{k-1} K_n C^\circ] \hat{x}^\circ(mT) + \\ &+ \sum_{i=0}^{k-1} [(A^\circ)^{k-1-i} B^\circ u(mT + iT) + (A^\circ)^{k-1} K_n C^\circ x^\circ(mT)]. \end{aligned}$$

Подставив эти выражения в (3.104), получим

$$\tilde{x}(mT + kT) = [(A^\circ)^k - (A^\circ)^{k-1} K_n C^\circ] \tilde{x}(mT)$$

или, возвратившись к интервалу T_1 ,

$$\tilde{x}(nT_1 + T_1) = [(A^\circ)^k - (A^\circ)^{k-1} K_n C^\circ] \tilde{x}(nT_1).$$

Из последнего уравнения видно, что выбором матрицы K_n можно обеспечить любое наперед заданное качество собственного движения автономной системы, определяющей ошибку наблюдения $\tilde{x}(nT_1)$. Выбор этой матрицы по методике, описанной в предыдущем параграфе для наблюдателей полной размерности, не вызывает трудностей, если матрица A° обратима. В этом случае любым из описанных ранее способов синтезируем матрицу

$$\tilde{B} = (A^\circ)^{k-1} K_n,$$

а затем находим

$$K_n = (A^\circ)^{-k+1} \tilde{B}.$$

Матрица A° обратима в том случае, когда в модели объекта отсутствуют звенья чистого запаздывания. Однако учет запаздывания управляющей ЭВМ приводит к появлению таких звеньев, т. е. к необратимости матрицы A° .

Предположим, что в случае необратимости матрица A° имеет вид

$$A^\circ = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & 0 \end{array} \right], \quad (3.105)$$

где A_{11} — квадратная матрица.

Представим матрицы C° и K_N в виде

$$C^\circ = [C_1 | C_2]; \quad K_N = \left[\begin{array}{c} N_1 \\ \hline N_2 \end{array} \right],$$

тогда

$$\begin{aligned} & (A^\circ)^k - (A^\circ)^{k-1} K_N C^\circ = \\ = & \left[\begin{array}{c|c} A_{11}^k - A_{11}^{k-1} N_1 C_1 - A_{11}^{k-2} A_{12} N_2 C_2 & A_{11}^{k-1} A_{12} - A_{11}^{k-1} N_1 C_2 - A_{11}^{k-2} A_{12} N_2 C_2 \\ \hline 0 & 0 \end{array} \right]. \end{aligned}$$

Отсюда видно, что нулевые собственные значения матрицы $[(A^\circ)^k - (A^\circ)^{k-1} K_N C^\circ]$ никаким выбором матрицы K_N изменить нельзя. Нулевые собственные значения обуславливают составляющие переходного процесса наблюдателя конечной длительности, поэтому нет особой нужды изменять эти собственные значения. Остальные характеристические числа матрицы $[(A^\circ)^k - (A^\circ)^{k-1} K_N C^\circ]$ выбором матрицы K_N можно сделать любыми, при этом целесообразно выбрать $N_2 = 0$, тогда

$$\begin{aligned} & (A^\circ)^k - (A^\circ)^{k-1} K_N C^\circ = \\ = & \left[\begin{array}{c|c} A_{11}^k - A_{11}^{k-1} N_1 C_1 & A_{11}^{k-1} A_{12} - A_{11}^{k-1} N_1 C_2 \\ \hline 0 & 0 \end{array} \right]. \end{aligned}$$

Выбором матрицы N_1 обеспечиваем $\tilde{w}_m \rightarrow 0$.

Используя выход наблюдающего устройства (3.103), управляющее воздействие на объект рассчитываем не по (3.82), а по следующей формуле:

$$u_m = [-K | L] \hat{x}_m^\circ.$$

Алгоритм управляющей ЭВМ, в которой реализуется регулятор с экстраполирующим наблюдателем, изображен на рис. 3.20; алгоритм синтеза экстраполирующего наблюдателя — на рис. 3.21. В последнем алгоритме предусмотрен переход к новому базису в том случае, когда имеются нулевые собственные числа матрицы A° . Матрица перехода к базису (3.105) является унитарной и получается из единичной матрицы переменной тех строк, которые соответствуют переставляемым координатам вектора x_m° . При переходе к базису (3.105) последними координатами вектора x_m°

в новом базисе должны быть значения управляющих сигналов на предыдущем такте управления. Именно эти координаты состояния, учитывающие запаздывание управляющей ЭВМ, обуславливают нулевые собственные значения матрицы A^0 .

Рассмотренный экстраполирующий наблюдатель полной размерности необходимо использовать в том случае, когда все измерительные устройства в системе управления выдают сигналы с тактом T_1 , большим такта выдачи управляющего воздействия T .

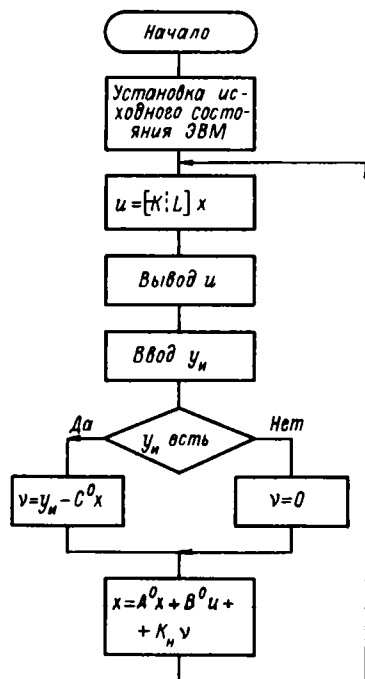


Рис. 3.20. Алгоритм комбинированного управления с экстраполирующим наблюдателем

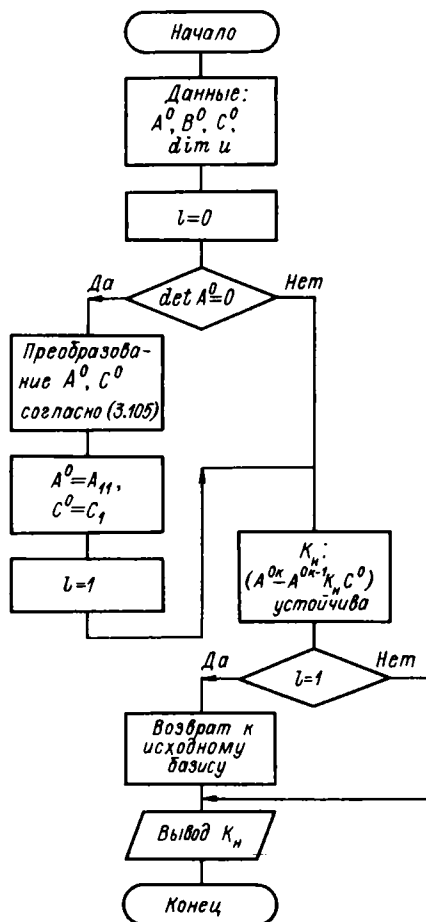


Рис. 3.21. Процедура синтеза экстраполирующего наблюдателя

Если же в системе управления кроме измерителей, работающих с тактом T_1 , имеются также датчики, выдающие сигналы измерения с тактом выдачи управляющего воздействия T , то сигналы таких датчиков целесообразно использовать для организации расчетов управляющего воздействия. При этом необходимо оценивать меньшее число неизмеряемых сигналов (для этой цели используются экстраполирующие наблюдатели пониженной раз-

мерности). Итак, считаем, что на объекте (3.78) установлено два типа измерителей. Один из них с выходной величиной y_1 работает с интервалом T , с этим же интервалом вырабатывается и управляющее воздействие на систему. Отсчеты второго измерителя, выходная величина которого y_2 , поступают с интервалом T_1 , причем $T_1 = kT$, целое число $k > 1$. Необходимо с тактом T восстановить те состояния системы, которые не входят в вектор изменения y_1 . Будем рассматривать нетривиальный случай, когда объект (3.78) по каждому из выходов не является полностью наблюдаемым, поэтому при построении наблюдателя необходимо использовать как отсчеты y_1 , так и отсчеты y_2 .

Пусть уравнение (3.78) динамической системы имеет вид

$$\begin{aligned} \begin{bmatrix} y_1(mT + T) \\ w(mT + T) \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y_1(mT) \\ w(mT) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(mT); \\ y_1(mT) &= [I \mid 0] \begin{bmatrix} y_1(mT) \\ w(mT) \end{bmatrix}; \\ y_2(nT_1) &= [C_1 \mid C_2] \begin{bmatrix} y_1(nT_1) \\ w(nT_1) \end{bmatrix}, \end{aligned}$$

где $n = \left[\frac{m}{k} \right]$ — целая часть числа m/k .

Такое представление уравнения (3.78) всегда может быть получено неособым преобразованием базиса. Закон управления (3.39) в этом случае реализуется в виде (3.98), причем матрицы N_1 и N_2 вычисляются точно также, как и для (3.98). В (3.98) вместо w_m используем оценку \hat{w}_m этого вектора, которая должна вырабатываться наблюдателем. Для построения наблюдателя перепишем уравнение динамической системы в виде системы разностных уравнений:

$$\begin{aligned} y_1(mT + T) &= A_{11}y_1(mT) + A_{12}w(mT) + B_1u(mT); \\ w(mT + T) &= A_{21}y_1(mT) + A_{22}w(mT) + B_2u(mT); \\ y_2(nT_1) &= C_1y_1(nT_1) + C_2w(nT_1). \end{aligned}$$

В соответствии с полученными уравнениями будем строить наблюдатель в виде динамической системы, описываемой уравнением

$$\hat{w}(mT + T) = A_{22}\hat{w}(mT) + A_{21}y_1(mT) + B_2u(mT) + K_{\pi}v(mT). \quad (3.106)$$

где невязка $v(mT)$ определяется следующим образом:

$$v(mT) = \begin{cases} y_2(mT) - C_1y_1(mT) - C_2\hat{w}(mT) & \text{при наличии отсчета } y_2; \\ 0 & \text{при отсутствии отсчета } y_2. \end{cases}$$

Будем выбирать матрицу K_n таким образом, чтобы ошибка

$$\tilde{w}(nT_1) = w(nT_1) - \hat{w}(nT_1)$$

стремилась к нулю при $n \rightarrow \infty$.

При помощи рассуждений, аналогичных проведенным для экстраполирующего наблюдателя полной размерности, получим разностное уравнение относительно ошибки

$$\tilde{w}(nT_1 + T_1) = (A_{22}^k - A_{22}^{k-1} K_n C_2) \tilde{w}(nT_1).$$

Отметим, что в данном случае A_{22} обычно обратима, так как практически всегда переменные состояния, учитываемые запаздывании ЭВМ, измеряются с интервалом T , т. е. входят в y_1 . Поэтому несложно выбором матрицы K_n обеспечить желаемое качество стремления ошибки $\tilde{w}(nT_1)$ к нулю.

Алгоритм синтеза наблюдателя с экстраполяцией пониженной размерности представлен на рис. 3.22; реализация регулятора с экстраполирующим наблюдателем пониженной размерности на управляющей ЭВМ — на рис. 3.23.

Частичная оценка состояний. Рассмотренные наблюдающие устройства строятся так, чтобы получить все координаты вектора состояния x_m^o расширенного объекта (3.78) частично в виде измеряемых координат y_m^n , частично в виде оцениваемых координат \hat{w}_m , т. е. $\dim y_m^n + \dim w_m = \dim x_m^o$. Однако нетрудно представить себе такие практические ситуации, когда не требуется информация о всех координатах вектора x_m^o объекта (3.78). Это, например, тот случай, когда вводятся комбинированные связи не по всем компонентам вектора ξ_m модели внешнего воздействия, а только по части этих компонент и достаточно получить установившуюся ошибку, не превосходящую величины ϵ^y , вместо нулевой установившейся ошибки. В процессе разработки регуляторов свободного движения могут выявляться случаи, в которых коэффициенты обратных связей по отдельным переменным состояниям объекта настолько малы, что этими связями вообще можно пренебречь (в инженерной практике такие связи часто называют недоминирующими). Таким образом, вполне реальны ситуации, когда для расчета управляющего воздействия не требуется знания всех координат вектора x_m^o и реализация на управляющей ЭВМ наблюдающих устройств, обеспечивающих получение всей возможной неизмеряемой информации, оказывается неэффективной. Поэтому возникает задача построения наблюдающих устройств, осуществляющих только частичную оценку неизмеряемых координат вектора x_m^o , а именно только тех координат, которые необходимы для реализации управления. Итак, рассмотрим задачу, в которой для объекта (3.78)

$$\begin{aligned} x_{m+1}^o &= A^o x_m^o + B^o u_m; \\ y_m^n &= C^o x_m^o, \end{aligned}$$

необходимо оценить вектор $w_{1m} = T x_m^o$ и $\dim y_m^n + \dim w_{1m} < \dim x_m^o = n$.

Всегда можно выбрать такой базис, при котором разностное уравнение объекта будет иметь вид

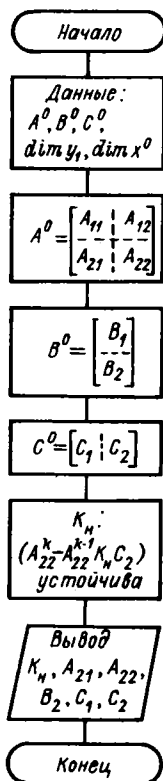


Рис. 3.22. Процедура синтеза экстраполирующего наблюдателя пониженной размерности

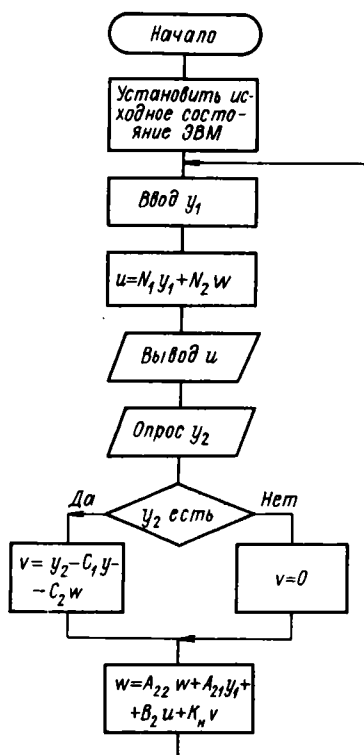


Рис. 3.23. Алгоритм комбинированного управления с экстраполирующим наблюдателем пониженной размерности

$$\begin{bmatrix} y_{m+1}^n \\ w_{1m+1} \\ w_{2m+1} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} y_m^n \\ w_{1m} \\ w_{2m} \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} u_m,$$

где w_{2m} есть те состояния, которые не измеряются и не оцениваются.

Это матричное уравнение эквивалентно системе:

$$y_{m+1}^n = A_{11}y_m^n + A_{12}w_{1m} + A_{13}w_{2m} + B_1u_m; \quad (3.107)$$

$$w_{1m+1} = A_{22}w_{1m} + A_{21}y_m^n + A_{23}w_{2m} + B_2u_m; \quad (3.108)$$

$$w_{2m+1} = A_{31}y_m^n + A_{32}w_{1m} + A_{33}w_{2m} + B_3u_m.$$

Основываясь на структуре уравнения (3.108), будем строить наблюдатель пониженной размерности, поведение которого описывается уравнением

$$\hat{w}_{1m+1} = A_{21}y_m^n + A_{22}\hat{w}_{1m} + B_2u_m + K_N v_m, \quad (3.109)$$

где невязку v_m определим следующим образом:

$$v_m = A_{12}w_{1m} + A_{13}w_{2m} - A_{12}\hat{w}_{1m}. \quad (3.110)$$

При реализации наблюдателя невязка v_m в соответствии с (3.107) выражается через входную и выходную величины объекта

$$v_m = y_{m+1}^n - A_{11}y_m^n - B_1u_m - A_{12}\hat{w}_{1m},$$

так что при реализации требуется программировать уравнение наблюдателя вида

$$\hat{w}_{1m+1} = (A_{22} - K_N A_{12})\hat{w}_{1m} + (A_{21} - K_N A_{11})y_m^n + (B_2 - K_N B_1)u_m + K_N y_{m+1}^n.$$

Рассмотрим ошибку работы устройства оценки

$$\tilde{w}_{1m} = w_{1m} - \hat{w}_{1m}.$$

Найдем

$$\tilde{w}_{1m+1} = w_{1m+1} - \hat{w}_{1m+1}$$

или на основании (3.108)—(3.110)

$$\tilde{w}_{1m+1} = A_{22}w_{1m} + A_{23}w_{2m} - A_{22}\hat{w}_{1m} - K_N A_{12}w_{1m} - K_N A_{13}w_{2m} + K_N A_{12}\hat{w}_{1m},$$

откуда получаем разностное уравнение ошибки

$$\tilde{w}_{1m+1} = (A_{22} - K_N A_{12})\tilde{w}_{1m} + (A_{23} - K_N A_{13})w_{2m}.$$

Из этого уравнения видно, что ошибка наблюдателя с течением времени стремится к нулю, если выбрать такую матрицу K_N , что одновременно выполняются условия $A_{23} - K_N A_{13} = 0$ и матрица $(A_{22} - K_N A_{12})$ устойчива. Однако эти два условия одновременно выполняются не всегда. В этом случае выбирается такая матрица K_N , которая обеспечивает устойчивость матрицы $(A_{22} - K_N A_{12})$ при минимизации нормы матрицы $(A_{23} - K_N A_{13})$.

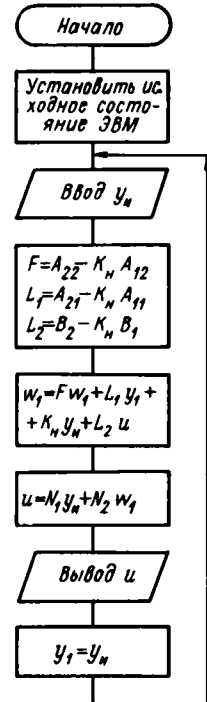


Рис. 3.24. Алгоритм управления с частичной оценкой состояний

Управляющее воздействие при частичной оценке состояний имеет вид (3.86); соответствующий этому алгоритм управляющей ЭВМ изображен на рис. 3.24.

3.9. Грубые регуляторы с экстраполированием

Рассмотрим возможности использования грубого регулятора в том случае, когда необходима экстраполяция, обусловленная большими интервалами квантования всех или некоторых измерительных устройств. Предположим в этой связи, что в системе, структурная схема которой приведена на рис. 3.9, может измеряться только ошибка ε с интервалом квантования T_1 ; входная g и выходная y величины порознь измеряться не могут. Управляющее воздействие необходимо вырабатывать с интервалом квантования T , причем $T_1 = kT$ ($k > 1$ — целое число).

Будем решать эту задачу в той же последовательности, что и в п. 3.4. Обозначим:

$$v_m = \begin{cases} \varepsilon_m & \text{при наличии } \varepsilon_m; \\ 0 & \text{при отсутствии } \varepsilon_m \end{cases} \quad (3.111)$$

и запишем уравнение объекта (3.48) в виде

$$\begin{bmatrix} \mathbf{x}_{m+1} \\ z_{m+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \Gamma \end{bmatrix} \begin{bmatrix} \mathbf{x}_m \\ z_m \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u_m + \begin{bmatrix} 0 \\ \mathbf{E} \end{bmatrix} v_m, \quad (3.112)$$

имея в виду только задачу слежения, когда $\mathbf{D} = 0$.

При этом считаем, что весь регулятор, в том числе и та его часть, которая реализует модель (3.34) внешнего воздействия g , работает с интервалом квантования T . Для объекта (3.112) выберем регулятор собственного движения, функционирующий в соответствии с (3.49) с интервалом T .

Уравнение замкнутой системы, состоящей из этого регулятора и объекта (3.112), имеет вид

$$\begin{bmatrix} \mathbf{x}_{m+1} \\ z_{m+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{BK}_1 & -\mathbf{BK}_2 \\ 0 & \Gamma \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_m \\ z_m \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{E} \end{bmatrix} v_m. \quad (3.113)$$

Рассмотрим развитие процессов в этой системе в интервале времени между поступлениями двух соседних импульсов измерения ошибки ε , т. е. развитие процессов в течение интервала T_1 . Пусть в момент времени $t = mT$ поступил импульс измерения ошибки ε_m , тогда в соответствии с (3.111) $v_m = \varepsilon_m = g_m - y_m = = \mathbf{H}\xi_m - \mathbf{C}\mathbf{x}_m$ и выражение (3.113) принимает вид

$$\begin{bmatrix} \mathbf{x}_{m+1} \\ z_{m+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{BK}_1 & -\mathbf{BK}_2 \\ -\mathbf{EC} & \Gamma \end{bmatrix} \begin{bmatrix} \mathbf{x}_m \\ z_m \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{EH} \end{bmatrix} \xi_m. \quad (3.114)$$

В моменты времени $t = mT + T, t = mT + 2T, \dots, t = mT + (k - 1)T$ измерения ошибки e_m отсутствуют и поэтому $v_m = 0$. Решая уравнение (3.113) при этом условии, получим

$$\begin{bmatrix} \frac{x_{m+k}}{z_{m+k}} \end{bmatrix} = \begin{bmatrix} A - BK_1 & -BK_2 \\ 0 & \Gamma \end{bmatrix}^{k-1} \begin{bmatrix} \frac{x_{m+1}}{z_{m+1}} \end{bmatrix}.$$

Подставляя в это выражение (3.114), после алгебраических преобразований получим

$$\begin{aligned} & \begin{bmatrix} \frac{x_{m+k}}{z_{m+k}} \end{bmatrix} = \\ & = \begin{bmatrix} (A - BK_1)^k + GEC & -(A - BK_1)^{k-1}BK_2 - G\Gamma \\ -\Gamma^{k-1}EC & \Gamma^k \end{bmatrix} \begin{bmatrix} \frac{x_m}{z_m} \end{bmatrix} + \\ & \quad + \begin{bmatrix} -GEH \\ \Gamma^{k-1}EH \end{bmatrix} \xi_m, \end{aligned}$$

где

$$G = (A - BK_1)^{k-2}BK_2 + (A - BK_1)^{k-3}BK_2\Gamma + \dots + BK_2\Gamma^{k-2}.$$

Перейдем к новому дискретному времени $n = m/k$, в результате чего получим разностное уравнение замкнутой системы для интервала квантования T_1 в виде

$$\begin{aligned} & \begin{bmatrix} \frac{x_{n+1}}{z_{n+1}} \end{bmatrix} = \\ & = \begin{bmatrix} (A - BK_1)^k + GEC & -(A - BK_1)^{k-1}BK_2 - G\Gamma \\ -\Gamma^{k-1}EC & \Gamma^k \end{bmatrix} \begin{bmatrix} \frac{x_n}{z_n} \end{bmatrix} + \\ & \quad + \begin{bmatrix} -GEH \\ \Gamma^{k-1}EH \end{bmatrix} \xi_n. \end{aligned} \quad (3.115)$$

Уравнения (3.34) командного генератора задающего воздействия для интервала $T_1 = kT$ имеют вид

$$\begin{cases} \xi_{n+1} = \Gamma^k \xi_n; \\ g_n = H \xi_n. \end{cases} \quad (3.116)$$

Будем считать, что система (3.114) устойчива, тогда можно найти вынужденное движение и вычислить установившуюся ошибку

$$e_n = g_n - y_{vn} = H \xi_n - Cx_{vn}.$$

В соответствии с (3.51) вектор вынужденного движения в системе (3.115) будет

$$\begin{bmatrix} x_{вн} \\ z_{вн} \end{bmatrix} = M \xi_n = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \xi_n,$$

где матрица M находится как решение уравнения (3.8), которое для системы (3.115), (3.116) имеет вид

$$M \Gamma^k - \left[\begin{array}{c|c} (A - BK_1)^k + GEC & -(A - BK_1)^{k-1} BK_2 - G\Gamma \\ \hline -\Gamma^{k-1} EC & \Gamma^k \end{array} \right] M = \begin{bmatrix} -GEN \\ \Gamma^{k-1} EN \end{bmatrix}.$$

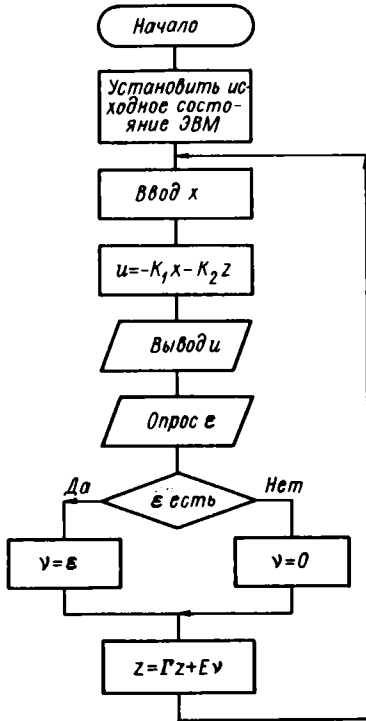


Рис. 3.25. Грубый алгоритм управления с экстраполяцией

Учитывая блочную структуру матрицы M , перепишем последнее уравнение в виде системы матричных уравнений:

$$M_1 \Gamma^k - [(A - BK_1)^k + GEC] M_1 + [(A - BK_1)^{k-1} BK_2 + G\Gamma] M_2 = -GEN;$$

$$M_2 \Gamma^k - \Gamma^k M_2 = \Gamma^{k-1} E (H - CM_1).$$

Последнее уравнение имеет такой же смысл, что и (3.52), следовательно, $H - CM_1 = 0$ и установившаяся ошибка в моменты времени nT_1 равна нулю. Таким образом, наличие двух интервалов квантования не меняет основного свойства системы с грубым регулятором, рассмотренного в п. 3.6, если матрица

$$\left[\begin{array}{c|c} (A - BK_1)^k + GEC & -(A - BK_1)^{k-1} BK_2 - G\Gamma \\ \hline -\Gamma^{k-1} EC & \Gamma^k \end{array} \right]$$

устойчива (устойчивость этой матрицы обеспечивается надлежащим выбором матриц обратной связи K_1 и K_2).

С учетом выражения (3.111) грубый алгоритм управления вынужденным движением с экстраполяцией будет иметь вид, изображенный на рис. 3.25.

3.10. Цифровое управление нестационарными объектами

Объекты с изменяющимися параметрами имеют достаточно широкое распространение в различных технологических установках и поэтому задача управления такими объектами имеет важное практическое значение. В дальнейшем будем рассматривать линейные нестационарные объекты, для которых получена разностная модель в виде

$$x_{m+1} = A_m x_m + B_m u_m, \quad (3.117)$$

при этом будем полагать, что элементы матрицы A_m и B_m являются известными функциями времени.

Так как вычисления текущего времени в управляющих микроЭВМ реализуются весьма просто, то расчет известных временных зависимостей затруднений не вызывает.

Элементы матриц A_m , B_m могут быть получены также в результате параметрической идентификации объекта управления [28, 99], выполняемой микроЭВМ в реальном масштабе времени. Параметрическая идентификация и расчет управляющего воздействия могут выполняться на различных микроЭВМ, обменивающихся данными по общей магистрали.

К управлению объектом (3.117) с известными в каждом такте параметрами A_m , B_m сводятся нелинейные задачи управления, рассматриваемые в гл. 4, а также другие задачи программного и пространственного управления [9, 44, 101]. И, наконец, модель (3.117) может быть получена в результате пошаговой статической линеаризации [75] нелинейного объекта (2.59).

Приведенные примеры иллюстрируют практическую важность нестационарной задачи, в которой матрицы A_m , B_m модели (3.117) известны в каждом такте управления.

Управление одномерным объектом. Пусть x_m , u_m — скалярные функции, т. е. $n = 1$, $m_u = 1$. Тогда $A_m = a_m$, $B_m = b_m$ — скаляры, изменяющиеся во времени. Исследуем возможность управления таким объектом при помощи введения стационарной обратной связи

$$u_m = -kx_m. \quad (3.118)$$

Уравнение замкнутой системы, состоящей из объекта (3.117) и регулятора (3.118), представим как

$$x_{m+1} = (a_m - b_m k) x_m.$$

Коэффициент обратной связи k необходимо выбирать таким образом, чтобы характеристическое число замкнутой системы по модулю не превышало бы некоторой величины η ,

$$|a_m - b_m k| \leq \eta,$$

где $0 < \eta < 1$.

При этом условии можно гарантировать асимптотическую устойчивость нестационарной замкнутой системы. Полученное неравенство перепишем в виде

$$\eta + a_m > b_m k > a_m - \eta.$$

Если $b_m > 0$, то коэффициент обратной связи, гарантирующий устойчивость процессов в замкнутой системе, должен удовлетворять условию:

$$(\eta + a_m)/b_m > k > (a_m - \eta)/b_m.$$

Так как неравенство справедливо для любых $t > 0$, то можем записать

$$\inf_{t>0} [(a_m + \eta)/b_m] > k > \sup_{t>0} [(a_m - \eta)/b_m]. \quad (3.119)$$

Достоинством стационарного закона управления (3.118) является простота его реализации. Однако при большом диапазоне изменения коэффициентов a и b не будет выполняться условие (3.119) и тогда при каких-то значениях t модуль собственного числа замкнутой системы может превзойти η и приблизиться к единице. В этом случае нельзя гарантировать асимптотическую устойчивость замкнутой нестационарной системы.

Условие (3.119) выполняется при малом диапазоне изменения параметров нестационарного объекта, поэтому стационарные законы управления для нестационарных объектов находят ограниченное применение. Более перспективны нестационарные законы управления, обеспечивающие приемлемое качество регулирования и в случае большого диапазона изменения параметров нестационарных объектов. Один из способов выбора нестационарного управления заключается в том, что задают динамические свойства замкнутой системы в виде

$$x_{m+1} = r(x_m),$$

где $r(x_m)$ — функция, обращающаяся в нуль при $x_m = 0$ и обеспечивающая устойчивость движения системы. Для скалярного объекта (3.117) целесообразно выбирать $r(x_m) = \eta x_m$.

На основании (3.117) и (3.119) при выбранной функции $r(\cdot)$ имеем

$$a_m x_m + b_m u_m = \eta x_m,$$

откуда находим алгоритм функционирования управляющей ЭВМ в виде

$$u_m = -[(a_m - \eta)/b_m] x_m. \quad (3.120)$$

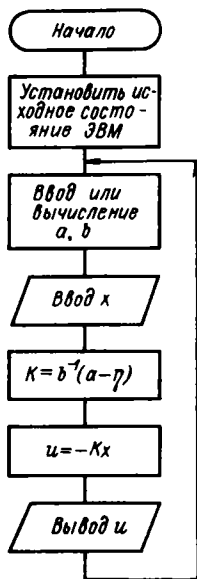


Рис. 3.26. Алгоритм управления нестационарным объектом первого порядка

динамические свойства замкнутой системы в виде

Замкнутая система управления, состоящая из нестационарного объекта (3.117) и нестационарного регулятора (3.120), является стационарной; качество процессов в такой системе определяется значением η .

Алгоритм управляющей ЭВМ, реализующей (3.120), изображен на рис. 3.26. Отличие этого алгоритма от рассматриваемых ранее заключается в наличии блока определения параметров объекта. Эти параметры вводятся с другой ЭВМ (если они получаются в результате параметрической идентификации объекта) или вычисляются по известным формулам.

Пример 3.5. Уравнение нестационарного объекта управления имеет вид

$$x_{m+1} = ((2m + 0,5)/(1 + m)) x_m + ((m + 0,2)/(1 + m)) u_m. \quad (3.121)$$

Сравнивая это уравнение с (3.117), получаем:

$$a_m = (2m + 0,5)/(1 + m); \quad b_m = (m + 0,2)/(1 + m).$$

Для придания желаемых свойств замкнутой системе используем закон управления (3.120), т. е.

$$u_m = -((2m + 0,5 - \eta(1 + m))/(m + 0,2)) x_m. \quad (3.122)$$

Выберем $\eta = 0,7$, тогда переходный процесс в замкнутой системе, состоящей из объекта (3.121) и регулятора (3.122), при начальном условии $x_0 = 2$ будет иметь вид, представленный на рис. 3.27, кривая 1. Кривая 2 на этом рисунке представляет изменение нестационарного коэффициента передачи $k = (1,3m - 0,2)/(m + 0,2)$, а кривая 3 — управляющего воздействия. Таким образом, использование нестационарного алгоритма управления согласно рис. 3.26 обеспечивает необходимое качество процессов в рассматриваемой системе.

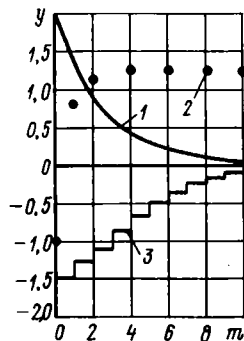


Рис. 3.27. Переходные процессы в системе управления нестационарным объектом: 1 — выходные координаты; 2 — коэффициента передачи; 3 — управляющего воздействия

Управление объектом канонической структуры. Использованный прием практически без изменений может быть применен для построения алгоритма управления линейным нестационарным объектом высокого порядка с управляемым каноническим представлением пары матриц (A_m , B_m). Действительно, пусть в уравнении (3.117) объекта управления матрицы A_m и B_m имеют вид:

$$A_m = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ a_{1m} & a_{2m} & a_{3m} & \dots & a_{nm} \end{bmatrix}; \quad B_m = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \\ b_m \end{bmatrix}, \quad (3.123)$$

Будем искать управляющее воздействие для такого объекта в виде нестационарной обратной связи

$$u_m = -K_m x_m, \quad (3.124)$$

тогда уравнение замкнутой системы будет

$$x_{m+1} = A_{cm}x_m,$$

где на основании (3.123) и (3.124)

$$A_{cm} = A_m - B_m K_m =$$

$$= \left[\begin{array}{ccc|ccc} 0 & & & 1 & & & \dots & & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & & & 0 & & & \cdot & \cdot & 1 \\ \hline a_{1m} - k_{1m}b_m & a_{2m} - k_{2m}b_m & \dots & a_{nm} - k_{nm}b_m & & & & & \end{array} \right]. \quad (3.125)$$

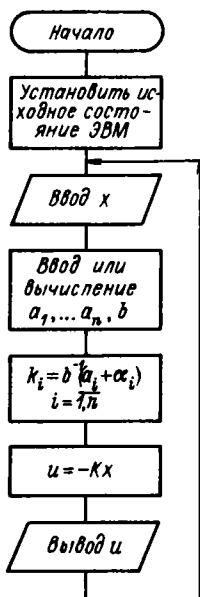


Рис. 3.28. Алгоритм управления нестационарным объектом n -го порядка

Найдем такую матрицу обратных связей, чтобы матрица состояния замкнутой системы A_{cm} имела заданный характеристический полином с постоянными коэффициентами

$$D(z) = z^n + \alpha_n z^{n-1} + \dots + \alpha_2 z + \alpha_1.$$

Характеристический полином выбирается, исходя из обеспечения требуемых свойств собственного движения замкнутой системы, как это описано, например, в п. 3.2. Матрица A_{cm} будет обладать характеристическим полиномом $D(z)$, если

$$A_{cm} = \left[\begin{array}{ccc|ccc} 0 & & & 1 & & & \dots & & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & & & 0 & & & \cdot & \cdot & 1 \\ \hline -\alpha_1 & -\alpha_2 & \dots & & & & & & -\alpha_n \end{array} \right]. \quad (3.126)$$

Из (3.125) и (3.126) видно, что для обеспечения заданных динамических свойств замкнутой системы коэффициенты нестационарной обратной связи необходимо выбирать в виде

$$k_{im} = b_m^{-1} (a_{im} + \alpha_{im}), \quad i = \overline{1, n}.$$

Использование нестационарных обратных связей для управления нестационарным объектом позволяет построить стационарную замкнутую систему с желаемыми динамическими свойствами. В данном случае существенную роль играет каноническое представление матрицы нестационарного объекта. Этот метод не удастся применить, когда матрицы объекта управления не имеют канонического представления.

Алгоритм управляющей ЭВМ, реализующий нестационарный закон управления нестационарным объектом, изображен на рис. 3.28.

Пример 3.6. Уравнение объекта имеет вид

$$y_{m+1} + [1 + 0,5(-1)^m] y_{m+1} + [(m + 0,2)/(m + 1)] y_m = (1 + 0,5^{m+1}) u_m.$$

Введем переменные состояния по формулам $x_{1m} = y_m$, $x_{2m} = y_{m+1}$, тогда уравнение объекта приводится к виду (3.117), причем

$$A_m = \begin{bmatrix} 0 & 1 \\ -(m + 0,2)/(m + 1) & -1 - 0,5(-1)^m \end{bmatrix};$$

$$B_m = \begin{bmatrix} 0 \\ 1 - 0,5^{m+1} \end{bmatrix}.$$

Выберем такие нестационарные обратные связи в законе управления (3.124), чтобы матрица замкнутой системы была стационарной и имела собственные значения, равные $z_1 = 0,5$; $z_2 = 0,6$; тогда характеристический полином замкнутой системы $D(z) = z^2 - 1,1z + 0,3$, а коэффициенты нестационарной обратной связи

$$k_{1m} = [(1 - 0,5^{m+1})(m + 1)]^{-1} (0,1 - 0,7m);$$

$$k_{2m} = (1 - 0,5^{m+1})^{-1} [-2,1 - 0,5(-1)^m].$$

Уравнение замкнутой системы при использовании найденных обратных связей

$$\begin{bmatrix} x_{1m+1} \\ x_{2m+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0,3 & 1,1 \end{bmatrix} \begin{bmatrix} x_{1m} \\ x_{2m} \end{bmatrix}.$$

На рис. 3.29 изображены: переходный процесс при $x_0 = [1 \ 1]^T$ (кривая 1); изменения коэффициентов обратной связи k_1 (последовательность 2) и k_2 (последовательность 3).

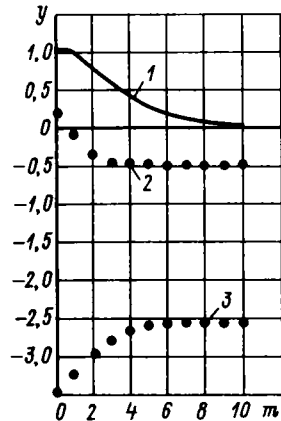


Рис. 3.29. Переходные процессы в системе управления нестационарным объектом канонической структуры:

1 — выходной координаты; 2 — коэффициента обратной связи k_1 ; 3 — коэффициента обратной связи k_2 .

Управление объектом произвольной структуры. Рассмотрим способ управления нестационарным объектом (3.117) для случая, когда матрицы A_m , B_m имеют произвольный вид, но как и прежде все элементы матриц являются известными ограниченными функциями времени. Будем искать управляющее воздействие для объекта (3.117) в виде нестационарной обратной связи (3.124), где матрицу K_m попытаемся выбрать таким образом, чтобы матрица состояния замкнутой системы A_{cm} ,

$$A_{cm} = A_m - B_m K_m, \quad (3.127)$$

в каждом такте расчета управляющего воздействия была подобна наперед заданной стационарной матрице F с собственными числами, удовлетворяющими условию

$$|\lambda_i[F]| < \eta, \quad 0 < \eta < 1. \quad (3.128)$$

Неравенство (3.128) устанавливает область расположения собственных чисел F в виде круга с радиусом η (см.рис. 3.1), не пре-

вышающим единицу, т. е. обеспечивает устойчивость матрицы F .
Условие подобия F и A_{cm} запишем в виде

$$A_{cm} = M_m F M_m^{-1}, \quad (3.129)$$

где M_m — матрица преобразования базисов (своя для каждого m).
Используя (3.127), перепишем (3.129) в виде

$$A_m M_m - M_m F = B_m K_m M_m. \quad (3.130)$$

В каждом такте расчета управляющего воздействия будем считать, что

$$K_m M_m = H,$$

где H — стационарная матрица, такая, что пара матриц (F, H) полностью наблюдаема.

При выполнении этого условия, а также при условии, что пара матриц (A_m, B_m) управляема для каждого m , уравнение (3.130) имеет невырожденное решение и тогда

$$K_m = H M_m^{-1}.$$

Матрица замкнутой системы (3.127) будет иметь вид

$$A_{cm} = A_m - B_m H M_m^{-1},$$

т. е. в общем случае является нестационарной.

Из условия (3.129) следует, что собственные числа матриц A_{cm} и F для любых m совпадают, т. е.

$$\lambda_i [A_{cm}] = \lambda_i [F] = \text{const}. \quad (3.131)$$

Учитывая (3.128), получим

$$\lambda_i [A_{cm}] \leq \eta, \quad (3.132)$$

т. е. матрица замкнутой системы (3.116), (3.124) для любых m устойчива.

В общем случае, так как матрица A_{cm} имеет переменные параметры, стационарности ее собственных чисел и выполнения условия (3.132) недостаточно для обеспечения устойчивости системы (3.126), (3.124). Несмотря на это, для многих практических приложений рассмотренные эвристические алгоритмы управления оказываются работоспособными. Оценка устойчивости синтезированной системы может быть осуществлена с привлечением метода функций Ляпунова (см. [40, 92] и п. 4.6) или экспериментальным путем.

Алгоритм работы управляющей ЭВМ, обеспечивающий стабилизацию свободного движения нестационарной системы, представлен на рис. 3.30.

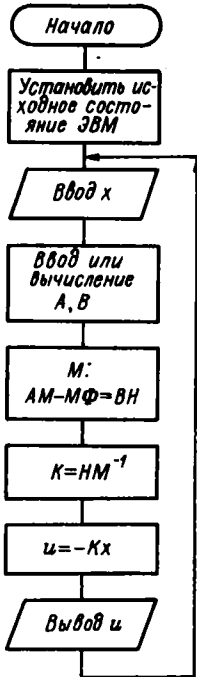


Рис. 3.30. Алгоритм управления нестационарным объектом

Пример 3.7. Осуществим синтез управления для нестационарного объекта второго порядка. В общем случае имеем:

$$A_m = \begin{bmatrix} a_{1m} & a_{2m} \\ a_{3m} & a_{4m} \end{bmatrix}; \quad B_m = \begin{bmatrix} b_{1m} \\ b_{2m} \end{bmatrix}.$$

Пусть собственные числа матрицы F будут λ_1 и λ_2 , тогда

$$F = \begin{bmatrix} 0 & 1 \\ -\lambda_1\lambda_2 & \lambda_1 + \lambda_2 \end{bmatrix}.$$

Зададим $H = [0 \ 1]$. Решив уравнение (3.130), вычислим

$$k_{1m} = \frac{(-\lambda_1 - \lambda_2 + a_{1m} + a_{4m})(a_{3m}b_{1m} - a_{1m}b_{2m}) - b_{2m}(\lambda_1\lambda_2 - a_{1m}a_{4m} + a_{2m}a_{3m})}{b_{1m}(a_{3m}b_{1m} - a_{1m}b_{2m}) - b_{2m}(a_{2m}b_{2m} - a_{4m}b_{1m})};$$

$$k_{2m} = \frac{(-\lambda_1 - \lambda_2 + a_{1m} + a_{4m})(a_{2m}b_{2m} - a_{4m}b_{1m}) - b_{1m}(\lambda_1\lambda_2 - a_{1m}a_{4m} + a_{2m}a_{3m})}{b_{2m}(a_{2m}b_{2m} - a_{4m}b_{1m}) - b_{1m}(a_{3m}b_{1m} - a_{1m}b_{2m})}.$$

Рассмотрим нестационарный объект, для которого

$$A_m = \begin{bmatrix} 1 & 1 + 0,5 \sin m \\ \cos m & 0 \end{bmatrix}; \quad B_m = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

причем выберем $\lambda_1 = 0,5$; $\lambda_2 = 0,4$.

Коэффициенты обратной связи для этого объекта:

$$\begin{cases} k_{1m} = (0,3 + \cos m + 0,25 \sin 2m) / (1 + 0,5 \sin m); \\ k_{2m} = 0,1. \end{cases} \quad (3.133)$$



Рис. 3.31. Переходный процесс в нестационарной замкнутой системе

Подставив (3.133) в (3.127), получим матрицу нестационарной замкнутой системы в виде

$$A_{cm} = \begin{bmatrix} 1 & 1 + 0,5 \sin m \\ \cos m - \frac{0,3 + \cos m + 0,25 \sin 2m}{1 + 0,5 \sin m} & -0,1 \end{bmatrix}.$$

Переходный процесс по координатам x_1 в замкнутой системе при начальном состоянии $x_0 = [1 \ 0]^T$ изображен на рис. 3.31.

Глава 4

АЛГОРИТМЫ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ НЕЛИНЕЙНЫХ СИСТЕМ

4.1. Нелинейные задачи управления

Нелинейные свойства управляемых объектов в наибольшей степени проявляются при больших отклонениях от заданного установившегося состояния системы. Линейные модели ОУ, полученные методами линеаризации в точке, с ростом отклонения все в меньшей степени отражают реальные динамические свойства объекта. Существенными факторами при работе системы становятся время переходного процесса и энергетические затраты на управление; повышается вероятность нарушения ограничений на управляющие сигналы и другие физические переменные системы. Тем самым качественно изменяются требования к процессу управления.

Особенности моделей объектов управления и задачи САУ при отработке значительных начальных рассогласований вызывают необходимость подробного исследования методов синтеза и алгоритмов управления нелинейных систем.

Будем рассматривать динамические системы, условия функционирования которых предусматривают регулярную отработку больших начальных рассогласований. Задавая в момент времени $t = 0$ начальное состояние объекта управления $\mathbf{x} = \mathbf{x}_0$ и в момент $t = t_f$ конечное состояние $\mathbf{x} = \mathbf{x}_f$, а также сформулировав требования к поведению системы на интервале $[0, t_f]$, определим тем самым *терминальную задачу* автоматического управления [10, 32].

Постановка задачи синтеза. Будем полагать, что поведение нелинейного объекта управления описывается дифференциальным уравнением

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x}) \mathbf{u}, \quad (4.1)$$

где \mathbf{x} — вектор пространства состояний R^n ($\mathbf{x} \in R^n$); \mathbf{u} — вектор управления ($\mathbf{u} \in R^h$); $\mathbf{f}(\mathbf{x})$ и $\mathbf{G}(\mathbf{x})$ — матрицы гладких функций соответствующих размерностей.

Уравнение (4.1) отражает динамические свойства широкого класса объектов управления, в частности, электроприводов и многоканальных систем, рассмотренных в п. 1.1.

В большинстве практических случаев ограничения на управляющие воздействия $\{u_j\} = \mathbf{u}$, $j = \overline{1, h}$ приводятся к виду

$$|u_j| \leq 1, \quad (4.2)$$

т. е. областью допустимых значений управления является гиперкуб $\Gamma \subset R^h$ с вершинами в точках U_s , $s = \overline{1, \pi}$; U_s — предельные векторы с координатами ± 1 .

Ограничения на координаты вектора состояния \mathbf{x} задаются уравнениями предельно допустимых значений соответствующих физических переменных (см. п. 1.1), которые в общем случае можно записать как

$$\varphi_s(\mathbf{x}) = 0, \quad (4.3)$$

где $\varphi_s(\mathbf{x})$ — гладкая функция ($s = 1, 2, \dots$).

Уравнения (4.3) в пространстве R^n описывают гиперповерхности S_s , ограничивающие область допустимых состояний системы RCR^n .

Качество работы системы управления оценивается функционалом (1.28). При больших начальных рассогласованиях особое значение приобретают быстродействие и энергетические затраты САУ. Выбирая различные подынтегральные функции $f_0(\mathbf{x}, \mathbf{u})$ получим [6, 26]:

функционал быстродействия САУ (при $f_0(\mathbf{x}, \mathbf{u}) = 1$)

$$J(\mathbf{u}) = t_f; \quad (4.4)$$

функционал расхода топлива

$$J(\mathbf{u}) = \int_0^{t_f} \mathbf{l}^T |\mathbf{u}| dt; \quad (4.5)$$

функционал затрат энергии

$$J(\mathbf{u}) = \int_0^{t_f} \mathbf{u}^T \mathbf{R} \mathbf{u} dt, \quad (4.6)$$

где \mathbf{l} — постоянный вектор весовых коэффициентов, $|\mathbf{u}| = \{|u_j|\}$; \mathbf{R} — положительно определенная симметрическая матрица ($\mathbf{R} = \mathbf{R}^T > 0$).

Комбинированные критерии быстродействия и затрат энергии или быстродействия и расхода топлива формируются в виде суммы выражений (4.4), (4.6) или (4.4), (4.5) соответственно.

Теперь можем сформулировать следующую терминальную задачу управления.

Задача 4.1. Найти управление $\mathbf{u} = \mathbf{u}^*(t)$ и переходный процесс $\mathbf{x} = \mathbf{x}^*(t)$ объекта управления (4.1), при $t \in [0, t_f]$ (t_f не задано):

а) удовлетворяющие граничным условиям $\mathbf{x}^*(0) = \mathbf{x}_0$, $\mathbf{x}^*(t_f) = \mathbf{x}_f$;

б) удовлетворяющие ограничениям на управление $u \in \Gamma$ и состояние $x \in R$;

в) доставляющие минимум заданному функционалу (1.28), т. е.

$$J(u^*) = \min_{u \in \Gamma} J(u).$$

Оптимальному переходному процессу в пространстве R^n будет соответствовать оптимальная фазовая траектория, проходящая через точки x_0 и x_f .

Оставляя пока открытым вопрос о методах нахождения численных или аналитических решений задачи 4.1, отметим, что поставленная оптимальная задача может иметь не единственное решение или не иметь решений вообще [6, 26]. Последний случай характерен для задач минимизации топлива и энергии с заданным временем t_f , в которых уменьшение функционала может достигаться за счет экономии управляющего воздействия u и, следовательно, за счет затягивания переходного процесса (в пределе $t_f \rightarrow \infty$ и задача теряет решение, см. [6]). В таких случаях решение задачи может быть все-таки получено при использовании *комбинированных критериев*, «штрафующих» систему за недостаточное быстроедействие. Поэтому в дальнейшем под управлением, оптимальным по затратам топлива или энергии, будем понимать управления, минимизирующие комбинированные функционалы (4.4) плюс (4.5), или (4.4) плюс (4.6).

Нахождение оптимальных законов изменения состояния и управляющего воздействия является лишь первым шагом решения задачи синтеза САУ. Непосредственное использование $u^*(t)$ как управляющего сигнала объекта (4.1), т. е. построение разомкнутой системы, на практике приводит к явно неудовлетворительным результатам. Отличие свойств реального объекта от используемой модели (4.1), начальные ошибки $x^*(0) - x_0$ и влияние неучтенных возмущений вызывают отклонения движения САУ от оптимального переходного процесса $x^*(t)$. Поэтому вторым шагом синтеза является построение замкнутой оптимальной системы, обеспечивающей решение задачи 4.1.

Задача 4.2. Найти замкнутый алгоритм управления $u = u(x, \dots)$, обеспечивающий для объекта управления (4.1) получение оптимального переходного процесса $x = x^*(t)$, удовлетворяющего условиям задачи 4.1.

Характерной особенностью терминальных задач с рассматриваемыми типами критериев качества является использование на некоторых временных интервалах из отрезка $[0, t_f]$ кусочно-постоянных управлений и, в частности, предельных значений U_{\dots} . С практической точки зрения очевидно, что подключение предельно допустимых управляющих воздействий повышает быстроедействие системы, уменьшает время интегрирования функционалов (4.5)—(4.6) и поэтому их применение вполне закономерно.

В открытой области (ядре) законы оптимального управления часто удается получить в следующей форме [6, 26]:

для оптимальных по быстродействию систем

$$u = \text{Sign } \Theta_T(x); \quad (4.7)$$

для систем, оптимальных по расходу топлива,

$$u = \text{Dez } \Theta_F(x); \quad (4.8)$$

для систем, оптимальных по расходу энергии,

$$u = \text{Sat } \Theta_E(x), \quad (4.9)$$

где $\text{Sign } \Theta_T = \{\text{sign } \Theta_{Tj}\}$; $\text{Dez } \Theta_F = \{\text{dez } \Theta_{Fj}\}$; $\text{Sat } \Theta_E = \{\text{sat } \Theta_{Ej}\}$ (нелинейные функции $\text{sign } (\cdot)$, $\text{dez } (\cdot)$, $\text{sat } (\cdot)$ представлены на рис. 4.1); $\Theta_{Tj}(x)$, $\Theta_{Fj}(x)$, $\Theta_{Ej}(x)$ — кусочно-гладкие функции.

Тогда в области R могут быть выделены множества знакопостоянства R_s , в которых управляющее воздействие принимает значения U_s . Поверхности

$$\Theta_s(x) = 0, \quad s = 1, 2, \dots, \quad (4.10)$$

где $\Theta_s(x)$ — гладкие функции, отделяющие указанные области знакопостоянства, называются *поверхностями переключения*.

При пересечении поверхностей (4.10) на входе в R_s координаты вектора управления меняют знак (алгоритм (4.7)), скачкообразно изменяют свои значения (алгоритм (4.8)) от 0 до ± 1 или плавно достигают предельных значений (алгоритм 4.9).

Пересечения гиперповерхностей (4.10) образуют в пространстве R^n *многообразия переключения*: *многообразия оптимального движения* (или просто *оптимальные многообразия*) — интегральные многообразия, содержащие оптимальные фазовые траектории объекта (4.1) и многообразия, не содержащие таких траекторий. Последний тип характерен для осциллирующих ОУ, а также систем, оптимальных по затратам энергии и топлива [6, 49]. Кроме указанного типа многообразий переключения к многообразиям оптимального движения будем относить характерные для нелинейных систем вырожденные многообразия (см. п. 4.3), а также граничные поверхности (4.3) и их пересечения, по которым проходят оптимальные фазовые траектории (см. рис. 1.21). При движении вдоль многообразия переключения вектор управления принимает предельные значения U_s в соответствии с выражениями (4.7), (4.8). Для других типов оптимальных многообразий характерно уменьшение модуля одной или нескольких переменных в соответствии со специально разрабатываемыми алгоритмами стабилизации пространственного движения (см. п. 4.5).

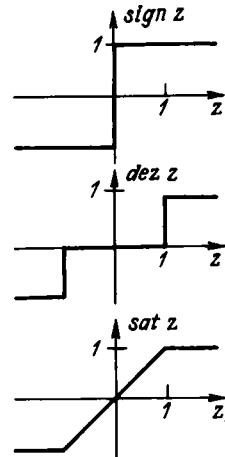


Рис. 4.1. Нелинейные функции

Общее описание оптимального многообразия S_s размерности $n - v_s$ задается уравнением

$$\Phi_s(\mathbf{x}) = 0, s = 1, 2, \dots, \quad (4.11)$$

где $\Phi_s(\mathbf{x}) = \{\varphi_{si}(\mathbf{x})\}$ ($i = \overline{1, v_s}$) — гладкая вектор-функция размерности v_s .

В перечень многообразий включены также граничные поверхности (4.3) и их пересечения, содержащие участки оптимальных фазовых траекторий. Итак, при проектировании оптимальных систем возникает необходимость решения важной промежуточной задачи.

Задача 4.3. Найти аналитическое описание поверхностей переключения (4.10) и оптимальных многообразий (4.11).

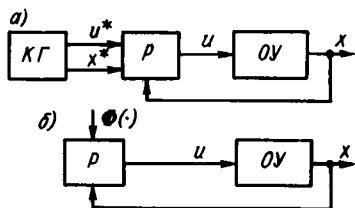


Рис. 4.2. Структурные схемы систем: а — программного управления; б — пространственного управления

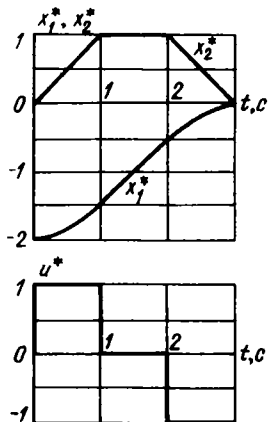


Рис. 4.3. Оптимальные переходные процессы

Общность характерных признаков рассматриваемого класса оптимальных систем позволяет подойти к решению задачи управления с единых позиций. В дальнейшем основное внимание уделим системам, оптимальным по быстродействию. Принципы организации управления в областях знакопостоянства и на оптимальных многообразиях остаются общими для всего класса систем. Управление на линейных участках алгоритма (4.9) носит специфический характер [6, 26, 49] и поэтому в этом разделе рассматриваться не будет.

Методы проектирования оптимальных систем. Выделим два основных метода проектирования и соответственно два типа замкнутых систем: программного управления и пространственного управления [9, 59, 69, 80].

Системы программного управления обеспечивают обработку оптимального задающего воздействия $\mathbf{x}^*(t)$. Функциональными элементами таких САУ служат задающее устройство или командный генератор (КГ) и регулятор (Р) (рис. 4.2, а). В простейших случаях задающее устройство может быть построено на основе запоминающего устройства микроЭВМ, в котором хранятся чис-

ловые массивы квантованных переменных вектора состояния и вектора управления, т. е.

$$\mathbf{x}^* = \mathbf{x}^*(mT); \quad \mathbf{u}^* = \mathbf{u}^*(mT),$$

где T — интервал квантования; $m = 1, 2, \dots$, или соответствующие этим массивам аппроксимирующие функции (см. п. 4.4). При необходимости обработки большого числа программных сигналов, полученных для различных значений \mathbf{x}_0 и \mathbf{x}_f , объем информации, хранящейся в памяти ЭВМ, увеличивается. Более рациональная организация задающего устройства предусматривает использование для генерации $\mathbf{x}^*(t)$ дискретной модели объекта управления (4.1).

Регулятор системы программного управления строится из условия минимизации ошибки

$$\mathbf{e} = \mathbf{x}^* - \mathbf{x}.$$

Пример 4.1. Рассмотрим структуру цифровой системы программного управления для линейного объекта [6, 26, 49]:

$$\dot{x}_1 = x_2; \quad \dot{x}_2 = u, \quad (4.12)$$

где функции x_1, x_2, u выражены в относительных единицах измерения, а аргументом служит безразмерное время.

Ограничение на управляющее воздействие задано в виде $|u| \leq 1$, скорость x_2 ограничена неравенством $|x_2| \leq 1$.

Требуется обеспечить оптимальный по быстродействию переходный процесс ОУ (4.12) из точки $\mathbf{x}_0 = \begin{bmatrix} x_{01} \\ 0 \end{bmatrix}$ в начало координат $\mathbf{x}_f = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Оптимальные переходные процессы по x_1, x_2 и закон изменения управления u^* приведены на рис. 4.3.

Определим отклонения:

$$e_1 = x_1^* - x_1; \quad e_2 = x_2^* - x_2. \quad (4.13)$$

Регулятор реализует алгоритм программного управления

$$P: u = u^* + k_1 e_1 + k_2 e_2. \quad (4.14)$$

Выбор коэффициентов обратной связи k_1, k_2 обеспечивает нужную динамику переходных процессов по ошибке при $e_1(0) \neq 0, e_2(0) \neq 0$ и грубость управления по отношению к возможным аддитивным и параметрическим возмущениям. Генерация оптимальных переходных процессов осуществляется задающим устройством, построенным на базе дискретного аналога модели (4.12), т. е.

$$\text{КГ: } \begin{cases} x_{1m+1}^* = x_{1m}^* + T x_{2m}^* + T^2 u_m^*/2; \\ x_{2m+1}^* = x_{2m}^* + T u_m^*, \end{cases} \quad (4.15)$$

где T — безразмерный интервал квантования; $m = 0, 1, 2, \dots$

Системами пространственного управления называют системы, в которых программа движения задается в виде функции переменных состояния объекта (рис. 4.2, б) [69, 80]. Управление в таких САУ определяется положением вектора состояния в пространстве R^n относительно заданных многообразий (4.11). Последовательное прохождение многообразий оптимального движения обеспечивает оптимальность фазовой траектории объекта управления и, следовательно, оптимальность системы в целом. Совпадение

участков фазовой траектории для различных начальных значений x_0 и тривиальный характер управления на других ($u = U_s$) позволяет избежать необходимости хранения в памяти микроЭВМ больших массивов информации.

Стабилизация движения относительно многообразия осуществляется регуляторами релейного типа [26, 34, 86], обеспечивающими получение скользящих режимов, или регуляторами, реализующими непрерывные алгоритмы стабилизации [9, 58—60, 69].

Аналоговая реализация релейных регуляторов широко известна и включает функциональные преобразователи, релейные и коммутационные элементы. Использование ЭВМ в контуре

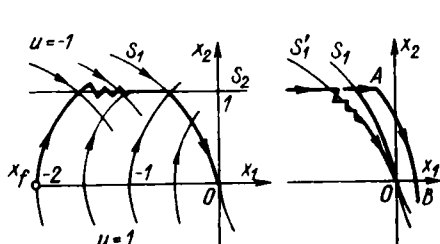


Рис. 4.4. Фазовые траектории релейной системы

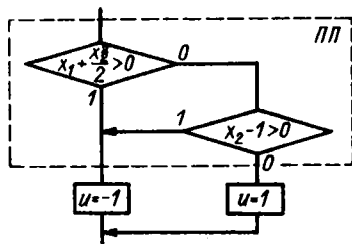


Рис. 4.5. Алгоритм управления релейной системы

Управления позволяет обойти ряд практических трудностей, возникающих при реализации нелинейных и логических функций.

Пример 4.2. Для задачи оптимального быстрогодействия из примера 4.1 рассмотрим структуру релейной системы, обеспечивающей перевод состояния ОУ (4.12) из произвольной точки $x_0 = \begin{bmatrix} x_{01} \\ x_{02} \end{bmatrix}$, $x_{01} < 0$, в начало координат. Фазовые траектории системы при $u = \pm 1$ и оптимальные переходные процессы представлены на рис. 4.4.

К многообразиям оптимального движения относятся: линия переключения S_1

$$x_1 + x_2^2/2 = 0; \quad (4.16)$$

линия предельно допустимой скорости S_2

$$x_2 - 1 = 0. \quad (4.17)$$

Можно выделить три характерных участка оптимальной фазовой траектории: разгон, т. е. начальный участок движения при $u = 1$; движение по границе S_2 и торможение вдоль линии переключения S_1 при $u = -1$. Программная реализация релейного регулятора (рис. 4.5) содержит планирующую программу, определяющую область расположения текущего состояния ОУ в пространстве R^2 , и модули присвоения значений управления. Разгон объекта осуществляется за счет присвоения $u = 1$; движение по границе — скольжением вдоль S_2 с поочередным подключением $u = 1$ и $u = -1$; торможение — за счет присвоения $u = -1$.

Отметим, что на последнем этапе движения система фактически оказывается разомкнутой и может удалиться от начала координат ввиду неточности используемой модели объекта, запаздывания микроЭВМ и т. д. (отрезок АВ, рис. 4.4).

Уменьшение чувствительности к этим факторам обычно осуществляется путем искажения оптимальной линии переключения S_1 . На рис. 4.4 показана субоптимальная линия переключения S'_1

$$1_{\Delta} x_1 + x_2^2/2 = 0,$$

соответствующая фазовой траектории при $u = -1_{\Delta} > -1$.

При использовании S'_1 на участке торможения возникнет скользящий режим, т. е. переключение двух ветвей управляющей программы, обеспечивающее скольжение вдоль S'_1 .

Релейные цифровые системы служат простейшими представителями систем с переменной структурой, в которых скольжение вдоль оптимальных многообразий осуществляется за счет поочередного использования нескольких ветвей управляющей программы. Их основным недостатком является наличие колебаний в окрестности многообразий, вызванных влиянием неучтенных в модели ОУ инерционных звеньев и люфтов, а также запаздыванием управляющей ЭВМ. Колебания приводят к ускоренному износу механических элементов и увеличению энергетических потерь.

Подход, предлагаемый авторами, предусматривает использование гладких алгоритмов управления, подключаемых в окрестностях оптимальных многообразий и обеспечивающих получение плавных переходных процессов. Необходимость последовательного прохождения различных областей знакопостоянства R_s и нескольких многообразий оптимального движения S_s требует многорежимной организации САУ и применения регуляторов переменной структуры, например, типа (1.20). В отличие от широко распространенных систем с переменной структурой, реализующих скользящие режимы движения [86], в рассматриваемых далее САУ коммутация структуры осуществляется эпизодически по мере прохождения характерных областей пространства состояний.

Многорежимная организация системы управления. Функциональными элементами многорежимной САУ являются алгоритм переключения структуры (1.19) и локальные алгоритмы управления или регуляторы (1.20) (см. рис. 1.10). Алгоритм переключения структуры определяет область пространства R^n , которой принадлежит текущее состояние объекта управления x , и назначает режим работы $r = r_s$.

Кроме областей знакопостоянства R_s выделим в пространстве малые окрестности оптимальных многообразий (4.11). Для этого зададим отклонение от S_s

$$e_s = \Phi_s(x), \quad (4.18)$$

или *вектор относительного движения* [61, 64] режима $r = r_s$.

Окрестности многообразий, т. е. области X_s , задаются неравенствами

$$\|e_s\| \leq \delta_s, \quad (4.19)$$

где $\delta_s > 0$.

К многообразиям оптимального движения при необходимости может быть отнесена также конечная точка x_f , т. е. многообразие размерности 0, определяемое в неявном виде как

$$e_0 = x_f - x = 0. \quad (4.20)$$

Соответствующая окрестность X_0 задается неравенством типа (4.19).

Определим четыре типа режимов работы оптимальной системы: разгон, движение по границе, торможение и стабилизацию (см. примеры 4.2, 4.3). Под *разгоном* будем понимать начальный этап движения, на котором система еще не достигает окрестностей X_s , а находится в одной из областей знакопостоянства R_s , где $u = U_s$. Режим *движения по границе* имеет место при достижении одного из множеств X_s , на котором $u \neq U_s$. На последних участках движения состояние САУ попадает в окрестности X_s поверхностей переключения, характеризующихся предельными значениями управления $u = U_s$. Этот тип режимов, обеспечивающих движение ОУ к конечной точке, называется *торможением*. И наконец, режим *стабилизации*, подключаемый в окрестности X_0 , обеспечивает стабилизацию объекта управления в конечной точке (см. гл. 3).

Отметим, что приведенная классификация условна ввиду своей неоднозначности, проявляющейся, например, для многоканальных систем, в которых разным управляющим переменным могут соответствовать различные режимы. Более строго процедура выделения режимов рассмотрена в п. 4.7.

По принадлежности вектора состояния одной из окрестностей X_s или области знакопостоянства R_s (вне X_s) устанавливается режим работы системы $r \in \{r_1, r_2, \dots, r_s, \dots, r_N\}$. Каждому режиму ставится в соответствие локальный алгоритм (регулятор) программного или пространственного управления в зависимости от избранного метода проектирования САУ. При использовании метода программного управления структура САУ содержит комплект локальных регуляторов P_s , осуществляющих обработку программных сигналов $x^*(t)$ в различных режимах работы, и комплект задающих устройств (генераторов задающего воздействия KG_s), переключаемых с изменением режима. Особенности рассматриваемого класса терминальных задач, т. е. возможность выделения кусочно-гладких участков переходного процесса $x^*(t)$ и кусочно-постоянных управлений $u^*(t)$ позволяют упростить аналитическое описание локальных алгоритмов и исключить необходимость хранения оптимальных временных последовательностей $x^*(mT)$ и $u^*(mT)$.

Метод пространственного управления вообще не требует генерации программных сигналов $x^*(t)$. Локальные алгоритмы (регуляторы P_s), подключаемые в различных режимах работы, обеспечивают стабилизацию движения объекта относительно текущего оптимального многообразия S_s (4.11) и оптимальное движение

к последующему многообразию S_{s+1} с использованием предельно допустимых значений соответствующих координат вектора управления u . Задача стабилизации движения по многообразию решается как задача минимизации вектора относительного движения e_s (4.18).

Пример 4.3. Рассмотрим структуру многорежимной системы оптимального управления для линейного ОУ (примеры 4.1, 4.2); определим отклонение от оптимального многообразия S_1 как

$$e_1 = x_1 + x_2^2/2$$

и отклонение от многообразия S_2 как

$$e_2 = x_2 - 1,$$

а также соответствующие им окрестности

$$X_1: |e_1| \leq \delta_1; \quad X_2: |e_2| \leq \delta_2$$

и области знакопостоянства R_1, R_2 (рис. 4.6).

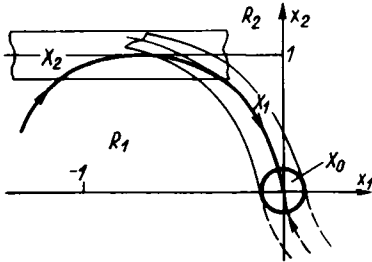


Рис. 4.6. Пространство состояний многорежимной системы

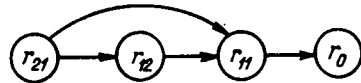


Рис. 4.7. Граф переключения режимов

Кроме того, определим окрестность начала координат

$$X_0: \|e_0\| \leq \delta_0,$$

где $e_0 = - \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

Выделим следующие режимы работы САУ: $r = r_{21}$ при $x \in R_1$ (разгон); $r = r_{22}$ при $x \in R_2$ (разгон); $r = r_{12}$ при $x \in X_2$ (движение по границе); $r = r_{11}$ при $x \in X_1$ (торможение); $r = r_0$ при $x \in X_0$ (стабилизация). Для случая, когда $x_{01} < 0$ и $x_{02} = 0$, граф-схема переключения режимов при оптимальном движении объекта управления приведена на рис. 4.7. Отметим, что при таких начальных условиях система не попадает в режим r_{22} (см. рис. 4.6).

Фрагмент управляющей программы представлен на рис. 4.8. Алгоритм переключения структуры реализуется планирующей программой ПП, осуществляющей выбор режима работы и подключение локального регулятора P_r . Содержание локальных алгоритмов определяется выбранным методом управления. Для систем программного управления имеем:

$$P_{21}: u = 1 + k_1(x_1^* - x_1) + k_2(x_2^* - x_2);$$

$$P_{12}: u = k_1(x_1^* - x_1) + k_2(x_2^* - x_2);$$

$$P_{11}: u = -1 + k_1(x_1^* - x_1) + k_2(x_2^* - x_2);$$

$$P_0: u = -k_1x_1 - k_2x_2.$$

Сигналы x_1^* и x_2^* вырабатываются командными генераторами КГ_s. В простейшем случае генераторы формируют оптимальные временные последовательности

$$x_1^* = C_{1s}^0 + mTC_{1s}^1 + m^2T^2C_{1s}^2;$$

$$x_2^* = C_{2s}^0 + mTC_{2s}^1 + m^2T^2C_{2s}^2,$$

где значения C_{is}^j коммутируются с изменением режима r .

Так для $x_{01} = -2$ значения коэффициентов выбирают в соответствии с табл. 4.1 (см. рис. 4.12). Для задач управления с переменными значениями x_{01}, x_{02} требуется в каждом случае назначать собственные коэффициенты разложения.

Алгоритмы пространственного управления принимают вид:

$$P_{21} : u = 1;$$

$$P_{12} : u = k_{12}e_2;$$

$$P_{11} : u = -1 + k_{11}e_1;$$

$$P_0 : u = k_{10}e_{10} + k_{20}e_{20}.$$

Таким образом, замыкание в таких системах осуществляется по отклонениям e_s от соответствующих многообразий. Выбор коэффициентов обратной связи $k_{11}, k_{12}, k_{10}, k_{20}$ производится из условия обеспечения нужной динамики относительного движения в областях X_s (см. пп. 4.5, 4.6).

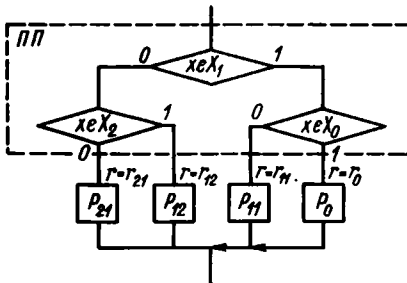


Рис. 4.8. Алгоритм управления много-режимной системы

Таблица 4.1

Коэффициенты командных генераторов

Гене- ратор	C_{1s}^0	C_{1s}^1	C_{1s}^2	C_{2s}^0	C_{2s}^1	C_{2s}^2
КГ ₂₁	-2,0	0,0	0,5	0,0	1,0	0,0
КГ ₁₂	-2,5	1,0	0,0	1,0	0,0	0,0
КГ ₁₁	-4,5	3,0	-0,5	-3,0	-1,0	0,0
КГ ₀	0,0	0,0	0,0	0,0	0,0	0,0

Как было показано, процедура синтеза оптимальных систем включает решение задач: 4.1 — расчета эталонных переходных процессов; 4.3 — расчета поверхностей переключения и многообразий оптимального движения; 4.2 — синтеза замкнутой САУ, т. е. алгоритма переключения структуры и локальных алгоритмов управления.

4.2. Необходимые условия оптимальности переходных процессов

Условия оптимальности переходных процессов и управляющих воздействий для рассматриваемых терминальных задач устанавливаются принципом максимума Понтрягина и теоремами Гамкрелдзе [6, 26, 49]. Формально условия носят необходимый характер и предназначены для анализа уже известных решений. Однако ввиду отсутствия иного конструктивного подхода они используются и для нахождения так называемых *экстремальных* решений, среди которых впоследствии отыскивают оптимальные (одно или несколько).

Аналитический характер принципа максимума не позволяет непосредственно перейти к решению задачи 4.1, т. е. получить процедуру расчета оптимальных переходных процессов. Поэтому

сформулируем ряд дополнительных положений, необходимых для построения такой процедуры. Ограничим рассмотрение оптимальными по быстродействию системами, отмечая, что результаты могут быть распространены и на другие системы терминального управления.

Условия оптимальности в открытом ядре. Использование принципа максимума позволяет свести задачу минимизации функционала (1.28) к более простой задаче поиска оптимального управления $u^*(t)$, доставляющего максимум функции Гамильтона

$$H = H(x, p, u),$$

где $p \in R^n$ — вектор состояния сопряженной динамической системы.

В рамках задачи 4.1 временно исключим ограничения на координаты вектора состояния и рассмотрим переходные процессы системы в открытой области допустимых состояний R , или в *открытом ядре* [26, 49]. Для модели объекта управления (4.1) сопряженная система описывается дифференциальным уравнением

$$\dot{p} = - \left[\frac{\partial f}{\partial x} + \sum_{j=1}^n \frac{\partial g_j}{\partial x} u_j \right]^T p, \quad (4.21)$$

где $\{g_j\} = G(x)$.

Гамильтониан оптимальной по быстродействию системы принимает вид

$$H = -1 + p^T [f(x) + G(x)u]. \quad (4.22)$$

Определим максимум функции H для произвольных $u \in \Gamma$ как

$$\mu(x, p) = \max_{u \in \Gamma} H(x, p, u).$$

Тогда справедлива следующая формулировка принципа максимума [6, 49].

Теорема 4.1. Необходимым условием оптимальности управления $u^*(t)$ и переходного процесса $x^*(t)$ ОУ (4.1) из начальной точки x_0 в конечную точку x_f будет существование ненулевого решения $p^*(t)$ уравнения (4.21), удовлетворяющего условиям:

$$a) \max_{u \in \Gamma} H(x^*, p^*, u) = \mu(x^*, p^*);$$

$$b) \mu(x^*, p^*)|_{t=t_f} = 0.$$

Сделаем ряд замечаний по формулировке теоремы 4.1.

1. Для рассматриваемого класса задач, в которых гамильтониан H явно не зависит от времени, а t_f не задано, справедливо условие более сильное, чем b , а именно [6]:

$$\mu^* = (x^*, p^*) \equiv 0, \quad (4.23)$$

для всех $t \in [0, t_f]$.

2. Будем называть *вырожденной* задачу (систему), в которой на конечном интервале времени $t \in [t^*, t^{**}]$ выполняется

$$G^T(x^*)p^* = 0,$$

и *нормальной* — задачу, в которой таких интервалов нет.

Для нормальной задачи из условия a получим [6, 26]:

$$u^* = \text{Sign } G^T(x^*)p^*. \quad (4.24)$$

Таким образом, как уже отмечалось в п. 4.1, в нормальной задаче минимизации функционала (4.4) управление принимает значения U_* , переключающиеся с изменением знака функции $G^T p^*$. Вопросы поиска решений для вырожденных задач рассмотрены в п. 4.3.

3. Формулировка теоремы включает $2n + h + 1$ уравнений: (4.1), (4.21), условия a , b . Функции x , p , u и время переходного процесса t_f являются неизвестными. Число неизвестных также равно $2n + h + 1$, т. е. формулировка содержит полный набор уравнений, необходимый для решения задачи. Замечание 2 позволяет сократить на h число условий и неизвестных путем замены условия a на (4.24). Если система нормальная, то поиск оптимальных решений сводится к решению краевой двухточечной задачи для *канонической модели* M , заданной системой дифференциальных уравнений (4.1), (4.21), (4.24) (рис. 4.9). Конечное значение одной из координат вектора p находится из условия b теоремы 4.1, а начальные и конечные значения вектора x заданы. Аналитическое решение двухточечной задачи, т. е. функции $x^*(t)$, $p^*(t)$ и число t_f , почти никогда получить не удастся. В работах [6, 49] рассмотрен подход, по которому поиск решения производится в результате пошагового моделирования канонической модели с последовательным перебором начальных значений сопряженного вектора p . Вычислительные сложности подхода приводят к его практической нереализуемости. Для нахождения оптимальных решений широко используются методы численного расчета в обратном времени пучка фазовых траекторий, исходящих из конечной точки движения [26, 34]. Развитие этого подхода, предусматривающее интегрирование канонической модели, предлагается в п. 4.4.

Прежде чем сформулировать необходимые условия оптимальности для систем с ограниченной областью допустимых состояний, рассмотрим отрезки оптимальных траекторий, лежащие на граничных поверхностях.

Условия оптимальности на границах. Рассмотрим одну из граничных поверхностей S_s (4.3), опуская в дальнейшем индекс s . Пусть $t = t_{in}$ — начальный момент движения по границе или момент входа траектории на поверхность S , а t_{out} — конечный момент или момент выхода. Пусть также $\mathbf{x}(t_{in}) = \mathbf{x}_{in}$, $\mathbf{x}(t_{out}) = \mathbf{x}_{out}$, $\mathbf{p}(t_{in}) = \mathbf{p}_{in}$, $\mathbf{p}(t_{out}) = \mathbf{p}_{out}$ (рис. 4.10). При движении по поверхности по крайней мере одна из координат вектора управления должна обеспечивать стабилизацию относительного движения, характеризующегося функцией

$$e = \varphi(\mathbf{x}),$$

т. е. обеспечивать $e = 0$.

Обозначим *стабилизирующую* координату управления u_σ . Если отрезок траектории целиком лежит на поверхности (4.3), то

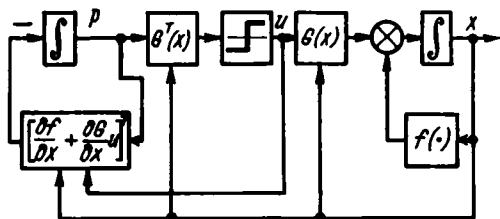


Рис. 4.9. Структурная схема канонической модели M оптимальной задачи

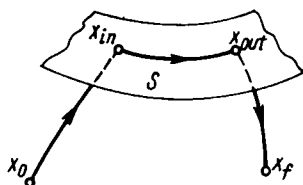


Рис. 4.10. Оптимальная фазовая траектория в пространстве с ограничением

$e(t_{in}) = 0$ и, следовательно, в рамках теоретического рассмотрения вопроса достаточно обеспечить выполнение условия

$$\dot{e} \equiv 0. \quad (4.25)$$

Остальные управляющие воздействия образуют *редуцированный вектор управления* $\bar{\mathbf{u}} \in R^{h-1}$ и *оптимизируют* движение по границе.

Естественно, что стабилизирующие и оптимизирующие компоненты вектора \mathbf{u} нельзя назначать произвольно, так как от $\bar{\mathbf{u}}$ зависит время переходного процесса из \mathbf{x}_{in} в \mathbf{x}_{out} . Более того, возможны случаи, когда оптимальное поведение системы на границе достигается при переключении ролей координат вектора \mathbf{u} в ходе переходного процесса.

Разделение функций управляющих воздействий производится с помощью преобразования

$$\mathbf{u} = [\mathbf{E}_1 | \mathbf{E}_2] \begin{bmatrix} \bar{\mathbf{u}} \\ u_\sigma \end{bmatrix} = \mathbf{E}_1 \bar{\mathbf{u}} + \mathbf{E}_2 u_\sigma, \quad (4.26)$$

где $[\mathbf{E}_1 | \mathbf{E}_2]$ — матрица перестановок, столбцы которой содержат нулевые компоненты кроме единственной 1, расположенной в каждом столбце в различных строках [29].

Тогда, если u_e — e -я координата вектора \mathbf{u} , то

$$\mathbf{G}(\mathbf{x}) \mathbf{E}_1 = \{\mathbf{g}_i(\mathbf{x})\}, \quad j \neq e; \quad \mathbf{G}(\mathbf{x}) \mathbf{E}_2 = \mathbf{g}_e.$$

Выбор матрицы преобразования гарантирует, что ограничения на координаты преобразованного вектора $\begin{bmatrix} \bar{\mathbf{u}} \\ u_e \end{bmatrix}$ сохранятся в виде (4.2), т. е.

$$|u_e| \leq 1 \quad (4.27)$$

и редуцированный вектор $\bar{\mathbf{u}}$ ограничен гиперкубом $\bar{\Gamma} \subset R^{h-1}$ с вершинами в точках $\bar{\mathbf{U}}_s, s = 1, 2, \dots,$

$$|\bar{u}_j| \leq 1. \quad (4.28)$$

Найдем стабилизирующее управление u_e , обеспечивающее выполнение условия (4.25). Дифференцируя e по времени с учетом (4.1), получим:

$$\dot{e} = \alpha(\mathbf{x}) + \mathbf{b}^T(\mathbf{x}) \bar{\mathbf{u}} + \beta(\mathbf{x}) u_e,$$

где

$$\begin{cases} \alpha(\mathbf{x}) = \frac{\partial \varphi}{\partial \mathbf{x}} \mathbf{f}; \\ \mathbf{b}^T(\mathbf{x}) = \frac{\partial \varphi}{\partial \mathbf{x}} \mathbf{G} \mathbf{E}_1; \\ \beta(\mathbf{x}) = \frac{\partial \varphi}{\partial x} \mathbf{G} \mathbf{E}_2 = \frac{\partial \varphi}{\partial x} \mathbf{g}_e. \end{cases} \quad (4.29)$$

Сравнивая полученное уравнение с (4.25), запишем

$$u_e = V(\mathbf{x}, \bar{\mathbf{u}}) = -[\alpha(\mathbf{x}) + \mathbf{b}^T(\mathbf{x}) \bar{\mathbf{u}}] / \beta(\mathbf{x}). \quad (4.30)$$

Для управления движением по S с помощью выбранной компоненты u_e вектора \mathbf{u} требуется, чтобы удовлетворялось (4.27) и условие $\beta(\mathbf{x}) \neq 0$. Если одно из этих условий не выполняется, то нужно попытаться подобрать другую матрицу преобразования $[\mathbf{E}_1; \mathbf{E}_2]$, т. е. другую стабилизирующую компоненту. Когда условия не удовлетворяются для всех матриц перестановок, движение по поверхности (4.3) из данной точки \mathbf{x}_{in} осуществить невозможно.

Будем полагать, что искомая матрица преобразования $[\mathbf{E}_1; \mathbf{E}_2]$ найдена. Тогда движение по границе обеспечивается управлением

$$\mathbf{u} = \mathbf{E}_1 \bar{\mathbf{u}} + \mathbf{E} V(\mathbf{x}, \bar{\mathbf{u}}).$$

Подставляя это выражение в (4.1), получим дифференциальное уравнение ОУ, замкнутого по стабилизирующей компоненте

$$\dot{\mathbf{x}} = \mathbf{f}_x(\mathbf{x}) + \mathbf{G}_x(\mathbf{x}) \bar{\mathbf{u}}, \quad (4.31)$$

где

$$\mathbf{f}_x(\mathbf{x}) = \mathbf{E} \mathbf{f}; \quad \mathbf{G}_x(\mathbf{x}) = \mathbf{E} \mathbf{G}; \quad \mathbf{E} = \mathbf{E}(\mathbf{x}) = \mathbf{I} - \frac{\mathbf{g}_e (\partial \varphi / \partial \mathbf{x})^T}{\partial \varphi / \partial x \mathbf{g}_e}.$$

Определим частную реализацию гамильтониана (4.22) для управления рассмотренной структуры

$$H_x(\mathbf{x}, \mathbf{p}, \bar{\mathbf{u}}) = H(\mathbf{x}, \mathbf{p}, \mathbf{E}_1 \bar{\mathbf{u}} + \mathbf{E}_2 V)$$

и ее наибольшее значение

$$\mu_x(\mathbf{x}, \mathbf{p}) = \max_{\bar{\mathbf{u}} \in \bar{\Gamma}} H_x(\mathbf{x}, \mathbf{p}, \bar{\mathbf{u}}).$$

Учитывая (4.31), найдем

$$H_x = -1 + \mathbf{p}^\top \mathbf{f}_x(\mathbf{x}) + \mathbf{p}^\top \mathbf{G}_x(\mathbf{x}) \bar{\mathbf{u}}. \quad (4.32)$$

Сопряженная система в этом случае задается уравнением

$$\dot{\mathbf{p}} = - \left(\frac{\partial H_x}{\partial \mathbf{x}} \right)^\top = - \left[\frac{\partial f_x}{\partial \mathbf{x}} + \sum_{j=1}^{h-1} \frac{\partial g_{xj}}{\partial \mathbf{x}} \bar{u}_j \right]^\top \mathbf{p}, \quad (4.33)$$

где $\{g_{xj}\} = \mathbf{G}_x(\mathbf{x})$, $\{\bar{u}_j\} = \bar{\mathbf{u}}$.

Определим также функцию $\lambda = \lambda(t)$ [49]:

$$\lambda = \frac{[G(\mathbf{x}) \mathbf{E}_2]^\top}{\beta(\mathbf{x})} \mathbf{p} = \frac{g_i^\top(\mathbf{x})}{\beta(\mathbf{x})} \mathbf{p}. \quad (4.34)$$

Теорема 4.2. Для оптимальности управления $\mathbf{u}^* = \mathbf{E}_1 \bar{\mathbf{u}}^* + \mathbf{E}_2 V$ и траектории $\mathbf{x}^*(t)$, при $t \in [t_{in}, t_{out}]$ целиком лежащей на поверхности (4.3), необходимо существование ненулевого решения сопряженной системы (4.33) $\mathbf{p}^*(t)$ и функции $\lambda^*(t)$, удовлетворяющих условиям:

- a) $\max_{\bar{\mathbf{u}} \in \bar{\Gamma}} H_x(\mathbf{x}^*, \mathbf{p}^*, \bar{\mathbf{u}}) = \mu_x$;
- б) $\mu_x \equiv 0$ для любых $t \in [t_{in}, t_{out}]$;
- в) $\frac{d\lambda^*}{dt} \leq 0$.

Доказательство теоремы 4.2 непосредственно следует из теоремы Гамкрелидзе [49] и приведено в приложении 1.

Сделаем следующие замечания.

1. Для нормальной задачи управления, т. е. при $(\mathbf{G}\mathbf{E}_1)^\top \mathbf{p}^* \neq 0$, из условия а получим:

$$\bar{\mathbf{u}}^* = \text{Sign}(\mathbf{G}\mathbf{E}_1)^\top \mathbf{p}^*, \quad (4.35)$$

т. е. редуцированный вектор управления $\bar{\mathbf{u}}$ принимает значения $\bar{\mathbf{U}}_s$ и, следовательно, оптимальным будем управление

$$\mathbf{u}^* = \mathbf{E}_1 \bar{\mathbf{U}}_s + \mathbf{E}_2 V. \quad (4.36)$$

2. Каноническая модель для участка траектории от \mathbf{x}_{in} до \mathbf{x}_{out} описывается уравнениями (4.31), (4.33), (4.36).

3. Представленный здесь результат предполагает, что выбор стабилизирующей компоненты u_0 или матрицы преобразования $[E_1; E_2]$ произведен оптимальным образом. Этим объясняется упрощение формулировки условий оптимальности по сравнению с теоремой Гамкрелидзе. При использовании теоремы 4.2 для синтеза экстремальных решений, т. е. как достаточных условий оптимальности, нужно перебрать все возможные h вариантов выбора стабилизирующей компоненты и сравнить результаты по критерию (4.4).

4. Для частного случая объектов управления с одним входом ($h = 1$) выражение (4.30) приобретает вид

$$u = V(x) = -\alpha(x)/\beta(x), \quad (4.37)$$

а каноническая модель описывается уравнениями:

$$\dot{x} = f_x(x); \quad \dot{p} = -\left(\frac{\partial f_x}{\partial x}\right)^T p.$$

Теорема 4.2 обосновывает возможность расчета отрезков оптимальных траекторий при интегрировании в обратном времени канонической модели порядка $2n$. Размерность модели можно уменьшить, если учесть, что координаты вектора состояния при движении по поверхности связаны уравнением (4.3). Для сокращения размерности задачи используется нелинейное преобразование уравнений объекта управления и сопряженной системы (см. приложение 1).

Осуществим преобразование координат объекта управления в соответствии с выражениями:

$$\begin{cases} e = \varphi(x); \\ \bar{x} = \Psi(x), \end{cases} \quad (4.38)$$

где \bar{x} — вектор продольного движения, характеризующий движение ОУ по поверхности S .

Будем полагать, что для $x \in S$ $\partial\varphi/\partial x \neq 0$, а $\Psi(x)$ выбирается как гладкая функция, для всех $x \in S$ (локально) удовлетворяющая условию

$$\det \left[\frac{\partial\varphi/\partial x}{\partial\Psi/\partial x} \right] \neq 0. \quad (4.39)$$

В простейшем случае \bar{x} — редуцированный вектор состояния, представленный неполным набором координат вектора x . Тогда $\Psi(x) = \Psi_0 x$, где Ψ_0 — прямоугольная матрица перестановок. Условие (4.39) гарантирует в любой точке $x \in S$ существование обратного преобразования

$$x = \Gamma(e, \bar{x}), \quad (4.40)$$

причем

$$\left[\frac{\partial \Gamma}{\partial e} \mid \frac{\partial \Gamma}{\partial \bar{x}} \right] = \left[\frac{\partial \varphi / \partial x}{\partial \Psi / \partial x} \right]^{-1} \quad (4.41)$$

(теорема об обратном преобразовании [35]).

Уравнение объекта управления (4.31) после преобразования (4.38) принимает вид:

$$\begin{cases} \dot{\bar{x}} = \frac{\partial \Psi}{\partial x} f_x(x) + \frac{\partial \Psi}{\partial x} G_x(x) \bar{u}; \\ \dot{e} = 0. \end{cases} \quad (4.42)$$

Определим сопряженный вектор состояния преобразованной системы как $\left[\begin{smallmatrix} p_e \\ \bar{p} \end{smallmatrix} \right]$, где p_e — скалярная функция, $\bar{p} \in R^{n-1}$. Уравнения связи с вектором p , как показано в приложении 1, определяется выражением

$$p = \left[\left(\frac{\partial \varphi}{\partial x} \right)^T \mid \left(\frac{\partial \Psi}{\partial x} \right)^T \right] \left[\begin{smallmatrix} p_e \\ \bar{p} \end{smallmatrix} \right]. \quad (4.43)$$

Для преобразованного объекта управления (4.42) частная реализация гамильтониана (4.32) принимает вид:

$$\begin{aligned} H_x \left(\Gamma(e, \bar{x}), \left[\left(\frac{\partial \varphi}{\partial x} \right)^T \mid \left(\frac{\partial \Psi}{\partial x} \right)^T \right] \left[\begin{smallmatrix} p_e \\ \bar{p} \end{smallmatrix} \right], \bar{u} \right) = \\ = -1 + \bar{p}^T \left(\frac{\partial \Psi}{\partial x} f_x + \frac{\partial \Psi}{\partial x} G_x \bar{u} \right) \Big|_{x=\Gamma(e, \bar{x})}. \end{aligned}$$

Таким образом, H_x не зависит от p_e и \dot{e} , что служит основанием для сокращения размерности задачи.

Получим описание *редуцированных* моделей. На поверхности S (при $e = 0$) продольное движение описывается уравнением, полученным из (4.32),

$$\dot{\bar{x}} = \bar{f}(\bar{x}) + \bar{G}(\bar{x}) \bar{u}, \quad (4.44)$$

где

$$\bar{f}(\bar{x}) = \frac{\partial \Psi}{\partial x} f_x(x) \Big|_{x=\Gamma(0, \bar{x})}; \quad \bar{G}(\bar{x}) = \frac{\partial \Psi}{\partial x} G_x(x) \Big|_{x=\Gamma(0, \bar{x})}.$$

Гамильтониан (4.42) описывается выражением

$$\begin{aligned} \bar{H}(\bar{x}, \bar{p}, \bar{u}) = H_x \left(\Gamma(0, \bar{x}), \left(\frac{\partial \Psi}{\partial x} \right)^T \bar{p}, \bar{u} \right) = \\ = -1 + \bar{p}^T [\bar{f}(\bar{x}) + \bar{G}(\bar{x}) \bar{u}], \end{aligned} \quad (4.45)$$

и, следовательно, редуцированный вектор сопряженной системы —

$$\dot{\bar{\mathbf{p}}} = - \left(\frac{\partial \bar{H}}{\partial \bar{\mathbf{x}}} \right)^\top = - \left[\frac{\partial \bar{f}}{\partial \bar{\mathbf{x}}} + \sum_{j=1}^{h-1} \frac{\partial \bar{g}_j}{\partial \bar{\mathbf{x}}} \bar{u}_j \right]^\top \bar{\mathbf{p}}, \quad (4.46)$$

где $\{\bar{\mathbf{g}}_j\} = \bar{\mathbf{G}}(\bar{\mathbf{x}})$.
Пусть

$$\bar{\mu} = \max_{\bar{\mathbf{u}} \in \bar{\Gamma}} \bar{H}(\bar{\mathbf{x}}, \bar{\mathbf{p}}, \bar{\mathbf{u}}).$$

Тогда из теоремы 4.2 вытекает следующий результат.

Теорема 4.3. Для оптимальности управления $\mathbf{u}^* = \mathbf{E}_1 \bar{\mathbf{u}}^* + \mathbf{E}_2 V$ и траектории $\mathbf{x}^*(t) = \Gamma[0, \bar{\mathbf{x}}^*(t)]$, при $t \in [t_{in}, t_{out}]$, целиком лежащей на границе (4.3), необходимо существование ненулевого решения $\bar{\mathbf{p}}^*(t)$ уравнения сопряженной системы (4.46), удовлетворяющего условиям:

$$\begin{aligned} a) \max_{\bar{\mathbf{u}} \in \bar{\Gamma}} \bar{H}(\bar{\mathbf{x}}^*, \bar{\mathbf{p}}^*, \bar{\mathbf{u}}) &= \bar{\mu}; \\ b) \bar{\mu} &\equiv 0; \\ в) [d(\bar{\mathbf{x}}^*) + D(\bar{\mathbf{x}}^*) \bar{\mathbf{u}}^*]^\top \bar{\mathbf{p}}^* &\leq 0, \end{aligned} \quad (4.47)$$

где

$$\begin{aligned} d(\bar{\mathbf{x}}) &= \frac{\partial \Psi}{\partial \bar{\mathbf{x}}} \left[\left(\frac{\partial}{\partial \bar{\mathbf{x}}} \frac{\mathbf{g}_i}{\beta} \right) \mathbf{f}_* - \left(\frac{\partial \mathbf{f}_*}{\partial \bar{\mathbf{x}}} \right) \frac{\mathbf{g}_i}{\beta} \right]_{\mathbf{x}=\Gamma(0, \bar{\mathbf{x}})}; \\ D(\bar{\mathbf{x}}) &= \frac{\partial \Psi}{\partial \bar{\mathbf{x}}} \left[\left(\frac{\partial}{\partial \bar{\mathbf{x}}} \frac{\mathbf{g}_i}{\beta} \right) \mathbf{G}_* - \left(\frac{\partial \mathbf{G}_*}{\partial \bar{\mathbf{x}}} \right) \frac{\mathbf{g}_i}{\beta} \right]_{\mathbf{x}=\Gamma(0, \bar{\mathbf{x}})}. \end{aligned}$$

Доказательство условия в, являющегося аналогом условия в теоремы 4.2, приведено в приложении 1.

Сделаем ряд замечаний.

1. Для нормальной задачи оптимального управления из условия а получим выражение, эквивалентное (4.35):

$$\bar{\mathbf{u}}^* = \text{Sign}[\bar{\mathbf{G}}^\top(\bar{\mathbf{x}}^*) \bar{\mathbf{p}}^*]. \quad (4.48)$$

2. Уравнение (4.48) связывает динамические системы (4.44) и (4.46) редуцированной канонической модели \bar{M} . Аналогично полной модели редуцированная модель будет использоваться для расчета оптимальных траекторий в обратном масштабе времени. Полученные в результате переходные процессы $\bar{\mathbf{x}}^*(t)$ и управляющие воздействия $\bar{\mathbf{u}}^*(t)$ пересчитываются в искомые решения задачи 4.1 с использованием преобразований $\mathbf{x} = \Gamma(0, \bar{\mathbf{x}})$ и (4.26).

Условия сопряжения отрезков оптимальной траектории. Оптимальная траектория, проходящая в замкнутом множестве R , содержит участки, лежащие в открытом ядре и на границе. Условия сопряжения этих участков в точках стыка $\mathbf{x}(t')$ (где t' — мо-

мент стыка) называются условиями скачка и формулируются следующим образом [26, 49].

Теорема 4.4. Для оптимальности траектории $\mathbf{x}^*(t)$, содержащей конечное число точек стыка $\mathbf{x}^*(t')$, и управления $\mathbf{u}^*(t)$ необходимо, чтобы вектор состояния сопряженной системы $\mathbf{p}^*(t)$ удовлетворял условиям скачка:

$$\mathbf{p}^*(t' + 0) = \mathbf{p}^*(t' - 0) + c \left(-\frac{\partial \Phi}{\partial \mathbf{x}} \right)^T,$$

где c — вещественное число.

Отметим, что на участке $[t_{in}, t_{out}]$ вектор \mathbf{p}^* определен с точностью до слагаемого $c(\partial \Phi / \partial \mathbf{x})^T$ [26]. Поэтому условие скачка можем отнести к одной из точек стыка, полагая, что в другой точке вектор \mathbf{p}^* разрыва не претерпевает. В дальнейшем условие скачка будем относить к точке входа на поверхность, т. е.

$$\mathbf{p}^*(t_{in} + 0) = \mathbf{p}^*(t_{in} - 0) + c \left(-\frac{\partial \Phi}{\partial \mathbf{x}} \right)^T, \quad (4.49)$$

тогда в точке выхода можем принять

$$\mathbf{p}^*(t_{out} + 0) = \mathbf{p}^*(t_{out} - 0). \quad (4.49a)$$

Использование условий скачка, а также условий б теорем 4.1 и 4.2 позволяет получить некоторые дополнительные условия, характеризующие состояние сопряженной системы в моменты t_f , t_{in} и t_{out} (см. приложение 1). Сформулируем следующее положение.

Теорема 4.5. Для оптимальности траектории $\mathbf{x}^*(t)$ и управляющего воздействия $\mathbf{u}^*(t)$ необходимо существование вектор-функции $\mathbf{p}^*(t)$, удовлетворяющей условиям:

а) в точках входа на поверхность $\mathbf{x}^* = \mathbf{x}_{in}^*$ — условию (4.49) и

$$[\mathbf{G}(\mathbf{x}_{in}^*) \mathbf{E}_2]^T \mathbf{p}^*(t_{in} - 0) = 0;$$

б) в точках выхода с поверхности $\mathbf{x}^* = \mathbf{x}_{out}^*$ — условию (4.49a) и

$$[\mathbf{G}(\mathbf{x}_{out}^*) \mathbf{E}_2]^T \mathbf{p}^*(t_{out}) = 0;$$

в) в конечной точке движения $\mathbf{x}^* = \mathbf{x}_f$ —

$$[\mathbf{f}(\mathbf{x}_f) + \mathbf{G}(\mathbf{x}_f) \mathbf{U}_s]^T \mathbf{p}^*(t_f) - 1 = 0, \quad s = 1, 2, \dots$$

Условия а, б определяют гиперплоскости S_{in} и S_{out} сопряженного пространства R^n , на которые попадает вектор \mathbf{p}^* в моменты стыка t_{in} и t_{out} . Для любого конкретного значения \mathbf{U}_s условие в задает в R^n гиперплоскость S_f допустимых значений конечного состояния сопряженной системы. Поведение вектора \mathbf{p}^* в сопряженном пространстве иллюстрируется рис. 4.11, где приведена фазовая траектория оптимальной системы и гиперплоскости S_f , S_{in} и S_{out} для случая, когда две последние совмещаются. На рисунке введены следующие обозначения: $\mathbf{p}_{in}^- = \mathbf{p}(t_{in} - 0)$, $\mathbf{p}_{in}^+ = \mathbf{p}(t_{in} + 0)$. При расчете оптимальных пере-

ходных процессов теорема 4.5 позволяет установить краевые условия интегрирования полной канонической модели. На участках движения по границе будем применять редуцированные модели типа (4.44), (4.46), (4.48). В связи с этим требуется получить связь краевых условий для векторов \mathbf{p} и $\bar{\mathbf{p}}$. Поскольку интегрирование ведется в обратном времени, то в момент входа на поверхность представляет интерес преобразование $\bar{\mathbf{p}}$ в \mathbf{p} , а в момент выхода — \mathbf{p} в $\bar{\mathbf{p}}$. Введем обозначения: $\bar{\mathbf{x}}_{in} = \bar{\mathbf{x}}(t_{in})$; $\bar{\mathbf{x}}_{out} = \bar{\mathbf{x}}(t_{out})$; $\bar{\mathbf{p}}_{in} = \bar{\mathbf{p}}(t_{in})$; $\bar{\mathbf{p}}_{out} = \bar{\mathbf{p}}(t_{out})$, $\mathbf{p}_f = \mathbf{p}(t_f)$.

Справедливо следующее положение (см. приложение 1).

Теорема 4.6. Для оптимальности траектории $\mathbf{x}^*(t)$ и управляющего воздействия $\mathbf{u}^*(t)$ необходимо существование $\mathbf{p}^*(t)$ — решения уравнения (4.21) и $\bar{\mathbf{p}}^*(t)$ — решения уравнения (4.46), удовлетворяющих условиям:

а) в точках входа на поверхность

$$\mathbf{p}_{in}^* = \Xi \left[\frac{\partial \Psi}{\partial \mathbf{x}} \right]^T \bar{\mathbf{p}}_{in}^*; \quad (4.50)$$

б) в точках выхода с поверхности

$$\bar{\mathbf{p}}_{out}^* = \left[\frac{\partial \Gamma}{\partial \mathbf{x}} \right]^T \mathbf{p}_{out}^*; \quad (4.51)$$

$$[\mathbf{G}(\mathbf{x}_{out}^*) \mathbf{E}_2]^T \mathbf{p}_{out}^* = 0; \quad (4.52)$$

в) в конечной точке движения

$$[\mathbf{f}(\mathbf{x}_f) + \mathbf{G}(\mathbf{x}_f) \mathbf{U}_s]^T \mathbf{p}_f^* - 1 = 0. \quad (4.53)$$

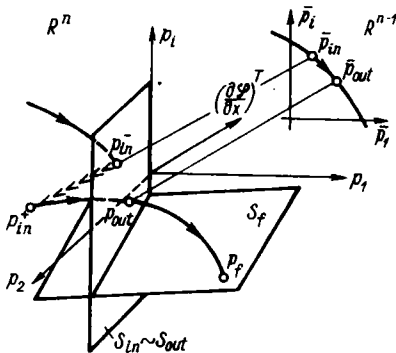


Рис. 4.11. Оптимальные траектории в сопряженных пространствах

Поведение редуцированного вектора $\bar{\mathbf{p}}^*$ в R^{n-1} и характер связи краевых значений показан на рис. 4.11.

Основной результат. Теоремы 4.1, 4.3 и 4.6 устанавливают необходимые условия оптимальности переходных процессов для траекторий, состоящих из участков движения в открытом ядре (разгон и торможение) и на граничных поверхностях S_s . Исполняя замечание 2 теоремы 4.1, замечание 1 теоремы 4.3 и понятие канонической модели, сформулируем следующее положение.

Теорема 4.7. Пусть оптимальная траектория нормальной системы из начальной точки \mathbf{x}_0 в конечную точку \mathbf{x} , имеет конечное число точек стыка \mathbf{x}_{in} и \mathbf{x}_{out} . Тогда оптимальный переходный процесс $\mathbf{x}^*(t)$ и оптимальное управление $\mathbf{u}^*(t)$ удовлетворяют следующим условиям:

а) на отрезках, лежащих в открытом ядре, $\mathbf{x}^*(t)$ и $\mathbf{u}^*(t)$ являются решениями канонической модели M : (4.1), (4.21), (4.24);

б) на отрезках, лежащих на граничных поверхностях S_s (4.3),

$$\mathbf{x}^* = \Gamma(0, \bar{\mathbf{x}}^*);$$

$$\mathbf{u}^* = \mathbf{E}_1 \bar{\mathbf{u}}^* + \mathbf{E}_2 V,$$

где $\bar{\mathbf{x}}^*$ и $\bar{\mathbf{u}}^*$ являются решениями редуцированной канонической модели \bar{M} (4.44), (4.46), (4.48) и удовлетворяют условию (4.47);

в) решения $\mathbf{p}^*(t)$ модели M и $\bar{\mathbf{p}}^*(t)$ модели \bar{M} в точках сопряжения $t = t_{in}$ и $t = t_{out}$ удовлетворяют условиям (4.50)—(4.52), а в конечной точке движения $t = t_f$ — условию (4.53).

Предложенная формулировка необходимых условий позволяет осуществить синтез ансамбля экстремальных переходных процессов, заканчивающихся в точке x_f , в результате интегрирования в обратном времени группы последовательно переключаемых канонических моделей. Начальные условия для интегрирования берутся в соответствии с условием в теоремы 4.7.

Пример 4.4. Проанализируем оптимальную задачу, рассмотренную в примере 4.1, при движении ОУ из точки $x_0 \begin{bmatrix} -2 \\ 0 \end{bmatrix}$.

В открытом ядре каноническая модель описывается уравнениями:

$$M : \begin{cases} \dot{x}_1 = x_2; \\ \dot{p}_1 = 0; \end{cases} \quad \begin{cases} \dot{x}_2 = u; \\ \dot{p}_2 = -p_1, \end{cases} \quad (4.54)$$

причем управляющее воздействие u находится из условия (4.24), как

$$u = \text{Sign } p_2.$$

Граничная поверхность описывается уравнением (4.17). Дифференцируя это выражение, найдем

$$\dot{\epsilon} = 0$$

и, следовательно, при движении по границе

$$u = V = 0.$$

Полная каноническая модель для отрезка фазовой траектории, лежащего на границе, описывается уравнениями:

$$\dot{x}_1 = x_2; \quad \dot{x}_2 = 0; \quad \dot{p}_1 = 0; \quad \dot{p}_2 = -p_1.$$

Редуцированную модель \bar{M} получим, определив предварительно функцию продольного движения $\bar{x} = x_1$ и сопряженную переменную $\bar{p} = p_1$, тогда

$$\bar{M} : \dot{\bar{x}} = 1; \quad \dot{\bar{p}} = 0. \quad (4.55)$$

Условие (4.47) принимает вид:

$$\bar{p} \leq 0. \quad (4.56)$$

В точке входа, т. е. при $t = t_{in}$, одновременно удовлетворяются условия (4.17) и

$$p_{in1} = \bar{p}; \quad p_{in2} = 0, \quad (4.57)$$

а в точке выхода (где $t = t_{out}$) — (4.17) и

$$p_2 = 0; \quad \bar{p} = p_1. \quad (4.58)$$

Конечное условие в момент t_f задается уравнениями плоскостей

$$p_2 - 1/U_s = 0, s = 1, 2, \quad (4.59)$$

где $U_1 = 1, U_2 = -1$.

На рис. 4.12 приведены кривые оптимального переходного процесса из $x_0 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$ в $x_f = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Из рисунка видно, что $t_{in} = 1$ с, $t_{out} = 2$ с, $t_f = 3$ с.

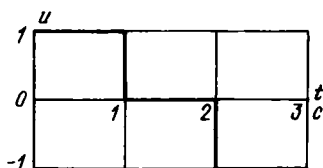
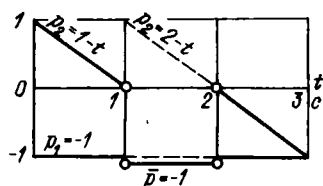
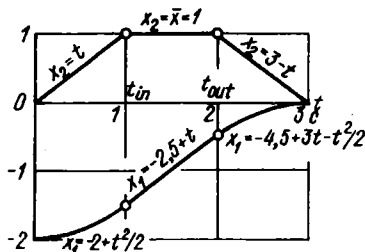


Рис. 4.12. Оптимальные переходные процессы для линейного ОУ

Нетрудно проверить, что на отрезках $t \in [0, 1]$ и $t \in [2, 3]$ переходный процесс является решением канонической модели (4.54), а на отрезке $t \in [1, 2]$ — редуцированной модели (4.55), причем условие (4.56) удовлетворяется. В точках стыка выполняются условия (4.57), (4.58), а в конечной точке — условие (4.59) в форме $p_2 = -1$ ($U_s = -1$).

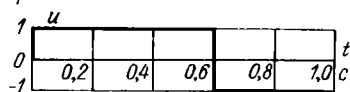
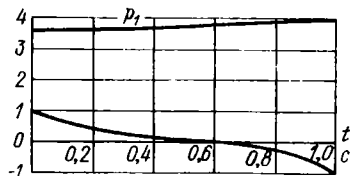
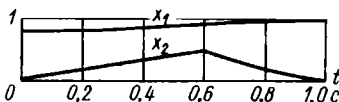


Рис. 4.13. Оптимальные переходные процессы для нелинейного ОУ

Отметим, что в точке входа, т. е. при $t = 1$ с, претерпевает скачок координата сопряженного вектора p_2 , которая не входит в уравнение редуцированной модели (4.55).

Пример 4.5. Рассмотрим задачу оптимального по быстродействию разгона электропривода, управляемого по обмотке возбуждения [9, 23, 97]. Объект управления описывается уравнениями:

$$\dot{x}_1 = -x_1 x_2^2 + x_2; \quad \dot{x}_2 = -x_2 + u, \quad (4.60)$$

где функции x_1, x_2, u — относительные величины скорости электродвигателя, тока и напряжения обмотки возбуждения, соответственно, а аргументом служит безразмерное время.

Конечное состояние объекта задается как $x_f = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Управляющее воздействие ограничено: $|u| \leq 1$, а на координаты вектора состояния ограничений не наложено.

Спряженная система в соответствии с выражением (4.21) описывается уравнениями:

$$\begin{cases} \dot{p}_1 = x_2^2 p_1; \\ \dot{p}_2 = (2x_1 x_2 - 1) p_1 + p_2. \end{cases} \quad (4.61)$$

Если для $t \in [0, t_f]$ не найдется интервалов, где $p_2 = 0$, т. е. задача нормальна, то в соответствии с (4.24)

$$u = \text{Sign } p_2. \quad (4.62)$$

Таким образом, для нормальной задачи управление u принимает значения $U_1 = 1$, $U_2 = -1$. Уравнения (4.60) — (4.62) описывают динамику модели М. Для определения конечных условий интегрирования сопряженной системы воспользуемся выражением (4.53). Найдем

$$S_f: U_s p_2 - 1 = 0,$$

и следовательно, $p_{f2} = 1$ при $s = 1$, $p_{f2} = -1$ при $s = 2$.

На рис. 4.13 приведены кривые оптимального переходного процесса разгона электропривода из точки $x_0 = \begin{bmatrix} 0,8 \\ 0,0 \end{bmatrix}$ в точку $x_f = \begin{bmatrix} 1,0 \\ 0,0 \end{bmatrix}$. Как показано в примере 4.7, гипотеза нормальности для ОУ (4.60) в общем случае не выполняется. При разгоне электропривода от начальных скоростей $x_1 < 0,8$ возможно существование временных интервалов, на которых $p_2 \equiv 0$.

4.3. Анализ вырожденной задачи оптимального управления

Рассмотренные в п. 4.2 условия оптимальности справедливы в предположении нормальности оптимальной задачи 4.1 (см. замечание 2 теоремы 4.1 и замечание 1 теоремы 4.3). В общем случае для нелинейных систем это условие не выполняется. На временных интервалах $[t^*, t^{**}]$, в которых

$$G^T(x) p = 0, \quad (4.63)$$

выражение (4.24) теряет смысл.

Аналогично этому и на граничных поверхностях могут быть выделены отрезки времени, на которых не определено выражение (4.48). Поэтому возникает необходимость поиска других условий, определяющих оптимальное поведение системы на интервалах вырожденности, и решения *вырожденной задачи управления*.

Ограничимся рассмотрением участков оптимальных траекторий, лежащих в открытом ядре области допустимых состояний R , для объектов управления с одним входом ($h = 1$), в которых u — скалярная функция. Следуя [6], подставим (4.63) в необходимое условие оптимальности (4.23) и получим

$$p^T f(x) = 1. \quad (4.64)$$

Учитывая, что на интервале $t \in [t^*, t^{**}]$ (4.63) — тождество, запишем

$$\frac{d^i}{dt^i} [p^T G(x)] = 0, \quad (4.65)$$

где $i = 0, 1, 2, \dots$

Последовательное дифференцирование $p^T G(x)$ с учетом уравнений ОУ (4.1) и сопряженной системы (4.21) позволяет привести (4.65) к виду

$$p^T [C_x(x) + [0 | C_u(x)] u] = 0, \quad (4.66)$$

где $C_x(x) = [C_{x_1} | C_{x_2} | \dots | C_{x_\eta}]$; C_{xi} , C_u — векторы, определяемые по формулам

$$C_{x_1} = G(x); \quad C_{xi} = \frac{\partial C_{x_{i-1}}}{\partial x} f(x) - \frac{\partial f}{\partial x} C_{x_{i-1}}, \quad i = \overline{2, \eta};$$

$$C_u = \frac{\partial C_{x_{\eta-1}}}{\partial x} G(x) - \frac{\partial G}{\partial x} C_{x_{\eta-1}},$$

и $\eta \leq n$.

Отметим, что для линейного объекта, где $f(x) = Ax$, $G(x) = \text{const}$, получим $\eta = n$, $C_u = 0$, а матрица C_x является матрицей управляемости. Объединяя (4.64) и (4.66), запишем

$$\left[\begin{array}{c} f^T(x) \\ C_x^T(x) + \left[\begin{array}{c} 0 \\ C_u^T(x) \end{array} \right] u \end{array} \right] p = \left[\begin{array}{c} 1 \\ \dots \\ 0 \end{array} \right]. \quad (4.67)$$

Полученное уравнение определяет в пространстве R^n нестационарное линейное многообразие вырожденных экстремальных процессов $p = p^*(t)$ сопряженной системы (4.21). Выполнение (4.67) является необходимым условием существования вырожденного решения задачи 4.1. Выражение (4.67) содержит $\eta + 1$ скалярное уравнение с n неизвестными координатами вектора p . По теореме Кронекера — Капелли [4] оно имеет решение тогда и только тогда, когда для некоторых решений уравнения (4.1) $x = x^*(t)$ и управлений $u = u^*(t)$ выполняется

$$\text{rang} \left[\begin{array}{c|c} f^T & 1 \\ \hline C_x^T + \left[\begin{array}{c} 0 \\ C_u^T \end{array} \right] u & 0 \end{array} \right] = \text{rang} \left[\begin{array}{c} f^T \\ C_x^T + \left[\begin{array}{c} 0 \\ C_u^T \end{array} \right] u \end{array} \right]. \quad (4.68)$$

Отметим, что последняя матрица прямоугольная и ее ранг не превосходит η .

Если указанные функции $x^*(t)$ и $u^*(t)$ удастся найти, то тем самым устанавливается возможность существования на интервале $[t^*, t^{**}]$ вырожденного решения задачи 4.1. Такое решение будет экстремальным и поэтому должно быть сопоставлено с другими решениями в целях выявления оптимума.

В тех случаях, когда для любых x и u условие (4.68) не удовлетворяется, система (4.67) противоречива и задача оптимального управления является нормальной. Так для линейной полностью

управляемой системы $\text{rang } C_x = n$. Следовательно, ранг матрицы в левой части (4.68) равен $n + 1$, а в правой — n , и такая система нормальна.

Нетрудно показать, что при $\eta < n$ условию (4.68) удовлетворяют любые x и u . Следовательно, экстремальное решение $p^*(t)$ всегда существует.

Практический интерес представляет случай $\eta = n$, для которого матрица $C_x + [0 \mid C_u]u$ квадратная. Из (4.68) получим условие

$$\det [C_x(x) + [0 \mid C_u(x)]u] = 0,$$

которому должны удовлетворять искомые решения $x^*(t)$ и $u^*(t)$.

После вычисления определителя выражение приводится к скалярному уравнению вида

$$\varphi_x(x) + \varphi_u(x)u = 0, \quad (4.69)$$

где $\varphi_x(x)$, $\varphi_u(x)$ — гладкие функции.

Если для $t \in [t^*, t^{**}]$ $\varphi_u(x) \neq 0$, из (4.69) найдем управляющее воздействие

$$u^* = -\varphi_x(x)/\varphi_u(x). \quad (4.70)$$

Если, кроме того, $|u^*| \leq 1$, то найденное управление, а также соответствующий ему переходный процесс $x^*(t)$, относятся к экстремальным решениям задачи 4.1. Для случая $\varphi_u(x) = 0$, что в частности имеет место при $C_u(x) = 0$, левую часть (4.69) представим в виде $\varphi_x(x) = \prod_s \varphi_s(x)$, где функции $\varphi_s(x)$ удовлетворяют условиям регулярности [62]. Тогда из (4.69) получим уравнения вида (4.3), которые задают в пространстве состояний R^n регулярные *гиперповерхности*, называемые *вырожденными*, или *особыми* [6]. Их возможные пересечения образуют *вырожденные многообразия* в R^n , описываемые уравнениями вида (4.11).

Отрезки интегральных фазовых траекторий объекта управления (4.1), принадлежащие вырожденным многообразиям, также могут оказаться оптимальными. Задача нахождения управления u^* , стабилизирующего движение на таких множествах, решается по схеме, рассмотренной в п. 4.2 (см. (4.30)), и более подробно исследуется в пп. 4.5, 4.6. Так как задача оптимального управления на вырожденных многообразиях во многом сходна с задачей оптимизации на ограничениях, то для нахождения вырожденных отрезков переходных процессов применим подход, описанный в п. 4.2. В частности, можем воспользоваться редуцированной канонической моделью вида (4.44), (4.46), (4.48). Координаты вектора сопряженной системы при $t \in [t^*, t^{**}]$ связаны соотношением (4.67), поэтому порядок динамической модели (4.46) можно понизить. Так, если ранги матриц в (4.68) равны n , то (4.67) имеет единственное решение, аналитическое описание которого замещает модель (4.46).

Пример 4.6. Продолжим рассмотрение задачи оптимального по быстродействию управления объектом (4.60). Для поиска вырожденных решений используется условие (4.63)

$$p_2 = 0. \quad (4.71)$$

Дифференцируя (4.71) по времени с учетом (4.61), найдем

$$(2x_1x_2 - 1)p_1 + p_2 = 0. \quad (4.72)$$

Гамильтониан задачи имеет вид

$$H = -1 + p_1(-x_1x_2^2 + x_2) + p_2(-x_2 + u).$$

Из условий $H = 0$ и (4.71) получим

$$(-x_1x_2 + 1)x_2p_1 = 1. \quad (4.73)$$

Таким образом,

$$C_x^\top = \begin{bmatrix} 0 & 1 \\ 2x_1x_2 - 1 & 1 \end{bmatrix}, \quad C_u^\top = 0.$$

Из условия $\det C_x = 0$ найдем уравнение вырожденного многообразия (линии) (см. рис. 4.14).

$$S_1: 2x_1x_2 - 1 = 0. \quad (4.74)$$

Проверим выполнение условия (4.68). Для этого используем уравнения (4.71) — (4.73) при x_1, x_2 , удовлетворяющих уравнению (4.74).

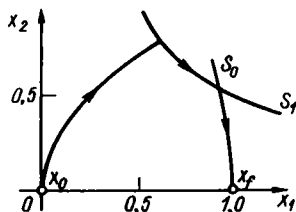


Рис. 4.14. Оптимальная фазовая траектория

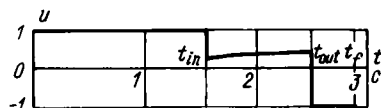
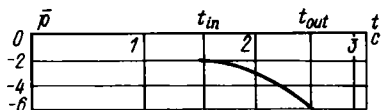
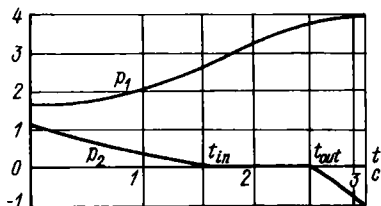
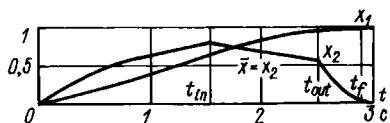


Рис. 4.15. Оптимальные переходные процессы для нелинейного ОУ (вырожденное решение)

Тогда

$$\text{rang} \begin{bmatrix} -x_2/2 & -x_2 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \text{rang} \begin{bmatrix} -x_2/2 & -x_2 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} = 2 \quad (4.75)$$

и, следовательно, на многообразии (4.74) существует решение сопряженной системы (4.61)

$$p_1 = 2/x_2; \quad p_2 = 0, \quad (4.76)$$

удовлетворяющее условию (4.68);

поэтому рассматриваемая задача является вырожденной и ее оптимальные траектории могут содержать участки, принадлежащие вырожденному многообразию S_1 .

Оптимальная фазовая траектория из точки $x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, вырожденное многообразие S_1 (линия), и линия оптимального переключения S_2 , которая в данном слу-

чае является интегральной кривой ОУ, приведены на рис. 4.14. Фазовая траектория включает участок разгона, участок движения по вырожденному многообразию S_1 и участок торможения по S_2 . Для описания движения системы по S_1 преобразуем координаты и построим каноническую модель \bar{M} . Определим отклонение от S_1 и продольное движение:

$$\begin{cases} e = x_2 - 1/(2x_1); \\ \bar{x} = x_2. \end{cases} \quad (4.77)$$

Дифференцируя e по времени, получим уравнение относительного движения

$$\dot{e} = -x_2 - x_2^2/(2x_1) + x_2/(2x_1^2) + u \quad (4.78)$$

и, следовательно,

$$u = V(x_1, x_2) = x_2 \left(1 + x_2/(2x_1) - 1/(2x_1^2) \right). \quad (4.79)$$

Аналогично, дифференцируя по времени \bar{x} и подставляя (4.77), найдем уравнение продольного движения

$$\dot{\bar{x}} = -\bar{x}^3. \quad (4.80)$$

Для нахождения связи (4.43) с преобразованным вектором сопряженной системы вычислим

$$\begin{aligned} \begin{bmatrix} \frac{\partial \Phi / \partial x}{\partial \Psi / \partial x} \end{bmatrix} &= \begin{bmatrix} 1/(2x_1^2) & 1 \\ 0 & 1 \end{bmatrix}; \\ \begin{bmatrix} \frac{\partial \Gamma}{\partial e} \mid \frac{\partial \Gamma}{\partial \bar{x}} \end{bmatrix} &= \begin{bmatrix} \frac{\partial \Phi / \partial x}{\partial \Psi / \partial x} \end{bmatrix}^{-1} = \begin{bmatrix} 1/(2\bar{x}^3) & -1/(2\bar{x}^2) \\ 0 & 1 \end{bmatrix} \end{aligned}$$

и, следовательно,

$$\bar{p} = \left(\frac{\partial \Gamma}{\partial \bar{x}} \right)^T p = \begin{bmatrix} -1/(2\bar{x}^2) & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}.$$

Подставляя (4.76), найдем

$$p = -1/\bar{x}^3. \quad (4.81)$$

Таким образом, уравнения (4.80), (4.81) описывают редуцированную каноническую модель \bar{M} . Условия сопряжения моделей M и \bar{M} выглядят следующим образом:

$$p_{in 1} = 2/\bar{x}; \quad p_{in 2} = 0; \quad \bar{p}_{out} = -1/\bar{x}^3,$$

причем во всех точках движения по S_1 (в том числе в точке выхода) должны одновременно удовлетворяться условия (4.74), (4.76).

Переходные процессы и кривая изменения управления, соответствующие оптимальной фазовой траектории (см. рис. 4.14) представлены на рис. 4.15.

4.4. Исследование оптимальных переходных процессов

Прежде, чем перейти к синтезу замкнутых алгоритмов управления САУ, требуется найти решения задач 4.1 и 4.3 (см. п. 4.1), т. е. получить аналитическое описание оптимальных переходных процессов и характерных многообразий рассматриваемой терминальной задачи. Процедура поиска оптимальных решений включает следующие этапы: 1) аналитической подготовки; 2) расчета

оптимальных переходных процессов; 3) аппроксимации переходных процессов и построение многообразий. На первом этапе с помощью математической модели объекта управления и уравнений граничных поверхностей находится аналитическое описание канонических моделей и условий их сопряжения.

Расчет переходных процессов (этап 2) осуществляется путем численного интегрирования канонических моделей. Этот этап исследования наиболее трудоемок, производится на ЭВМ и требует разработки специального математического обеспечения.

На третьем этапе полученные числовые массивы оптимальных переходных процессов для эффективного их использования в САУ должны быть приведены к аналитическому виду. Это делается с помощью процедур аппроксимации. Результатом аппроксимации могут быть аналитические выражения, описывающие оптимальные решения $x^*(t)$ и $u^*(t)$, используемые в простейших системах программного управления, или описания многообразий переключения и оптимального движения, необходимые для построения алгоритмов пространственного управления и переключения структуры.

Аналитическая подготовка. Использование теоремы 4.7 и результатов п. 4.3 позволяет осуществить расчет в обратном времени ансамбля экстремальных фазовых траекторий, проходящих через заданную конечную точку x_f . Неоптимальные решения впоследствии отсеиваются путем сравнения по критерию (1.28) траекторий, исходящих из одной и той же точки x_0 . Метод расчета предусматривает введение функции обратного времени

$$v = t_f - t$$

и соответствующего преобразования канонических моделей.

Каноническая модель (4.1), (4.21), (4.24), описывающая движение системы в режиме разгона и торможения в открытом ядре множества R^n , при подстановке $t = t_f - \tau$ приводится к виду:

$$M: \begin{cases} \frac{dx}{d\tau} = -f(x) - G(x)u; \\ \frac{dp}{d\tau} = \left[\frac{\partial f}{\partial x} + \sum_{j=1}^h \frac{\partial g_j}{\partial x} u_j \right]^T p; \\ u = \text{Sign } G^T(x) p. \end{cases}$$

Для построения моделей \bar{M}_s , необходимых для расчета отрезков фазовых траекторий, лежащих на граничных поверхностях S_s , используются описания ОУ (4.1) и граничных условий (4.3), которые формируются непосредственно из требований технического задания на разрабатываемую САУ (см. п. 1.1). Аналитическая подготовка моделей \bar{M}_s включает выбор операторов преобразования векторов управления $[E_1 \mid E_2]$ и состояния $\begin{bmatrix} \Phi(x) \\ \Psi(x) \end{bmatrix}$,

нахождение обратного преобразования $\Gamma(e, \bar{x})$. После этого осуществляется синтез стабилизирующего управления V_s (4.30), формируется каноническая модель (4.44), (4.46), (4.48), которая с учетом обратного масштаба времени преобразуется к виду:

$$\bar{M}_s : \begin{cases} \frac{d\bar{x}}{d\tau} = -\bar{f}(\bar{x}) - \bar{G}(\bar{x})\bar{u}; \\ \frac{d\bar{p}}{d\tau} = \left[\frac{\partial \bar{f}}{\partial \bar{x}} + \sum_{j=1}^{b-1} \frac{\partial \bar{g}_j}{\partial \bar{x}} u_j \right]^T \bar{p}; \\ \bar{u} = \text{Sign } \bar{G}^T(\bar{x}) \bar{p}. \end{cases}$$

Условие оптимальности (4.47) в обратном времени принимает вид

$$[d(\bar{x}) + D(\bar{x})\bar{u}]^T \bar{p} \geq 0 \quad (4.82)$$

и наряду с условием (4.28) проверяется на каждом шаге интегрирования \bar{M}_s .

Аналогичные модели строятся для расчета вырожденных участков оптимального движения. Уравнения вырожденных поверхностей находят из выражения (4.69). При этом отсутствует необходимость проверки условия (4.82) и возможно упрощение модели сопряженной системы за счет использования уравнения (4.67).

Так как многообразия вида (4.11), полученные как пересечения граничных или вырожденных поверхностей, также могут оказаться оптимальными, то на этапе подготовки находится их аналитическое описание и формируются редуцированные канонические модели \bar{M} . Процедура формирования этих моделей принципиально не отличается от предложенной для моделей \bar{M} , что вполне естественно, если учесть следующее. Оптимальное многообразие размерности $n-2$ можно рассматривать как граничную или вырожденную поверхность пространства R^{n-1} , применяя к нему все результаты, полученные в пп. 4.2, 4.3. Аналогично решаются вопросы подготовки моделей для многообразий произвольной размерности.

Область начальных значений $p = p_f$ вектора сопряженной системы для интегрирования модели M на участке торможения, т. е. при $\tau \in [0, t_f - t_{out}]$, задается выражением (4.53). Условием перехода траектории на границу будет одновременное выполнение (4.3) и (4.52), а перехода на вырожденную поверхность — (4.3) и (4.67). Принимая во внимание естественные погрешности численного интегрирования, условия перехода задают в виде неравенств, определяющих δ -окрестности оптимальных поверхностей и поверхностей S_{out} (см., например, (4.19)).

Начальными условиями для интегрирования моделей \bar{M}_s будут значения x_{out} и p_{out} , полученные канонической моделью M в мо-

мент $\tau = t_j - t_{out}$ и преобразованные к \bar{x} и \bar{p} в соответствии с выражениями (4.40), (4.51). Все точки \bar{x} и \bar{p} , полученные на граничной поверхности, будут начальными точками \bar{x}_{in} и \bar{p}_{in} для расчета участков разгона на интервале $\tau \in [t_j - t_{in}, t_j]$. При переходе к интегрированию модели M в режиме разгона ее начальные значения x_{in} и p_{in} устанавливаются в соответствии с преобразованиями (4.40) и (4.50).

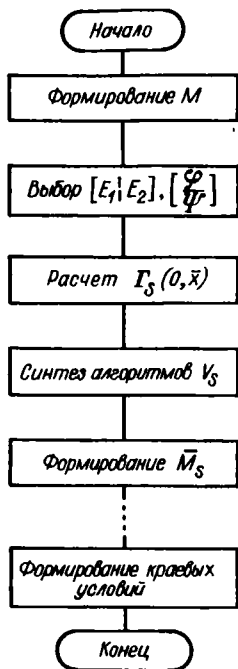


Рис. 4.16. Процедура аналитической подготовки

Аналогично подготавливаются условия переключения и начальные значения для всех моделей \bar{M}_s и т. д., предназначенных для расчета отрезков, лежащих на оптимальных многообразиях.

Процедура аналитической подготовки показана на рис. 4.16. Основными ее элементами являются плохо формализуемые операции нахождения частных производных, обращения нелинейных функций, а также процедуры линейной алгебры. Аналитическая подготовка требует больших временных затрат и адекватной математической культуры разработчиков. Поэтому представляют интерес унифицированные алгоритмы расчета переходных процессов, предусматривающие численное выполнение указанных операций в ходе общей вычислительной процедуры.

Расчет оптимальных переходных процессов. Численное интегрирование канонических моделей, переключаемых в определенной последовательности, позволяет построить оптимальные (экстремальные) фазовые траектории, соответствующие заданной конечной точке движения. Естественно, что процедура расчета ориентирована на широкое использование библиотечных подпрограмм решения нелинейных дифференциальных уравнений и матричной алгебры.

Укрупненная процедура расчета представлена на рис. 4.17. Она содержит модули интегрирования канонических моделей M , \bar{M} , \bar{M} и т. д., *планирующую программу*, модули расчета стабилизирующих управлений V_s , преобразования координат и формирования массивов экстремальных решений.

Основной функцией планирующей программы, так же как и аналогичного модуля цифровой САУ, является анализ условий попадания вектора состояния x в окрестности многообразий S_s , вектора p на гиперповерхности S_{out} и подключение нужной ветви процедуры. Кроме этого рассматриваемый модуль производит

перебор краевых условий канонических моделей для получения очередного участка фазовой траектории и обеспечивает окончание счета при достижении граничной поверхности, на которой не выполняются условия оптимальности (4.28) или (4.47).

Модули интегрирования с тактом T вычисляют дискретные значения векторов $\mathbf{x} = \mathbf{x}(mT)$, $\mathbf{p} = \mathbf{p}(mT)$ и $\mathbf{u} = \mathbf{u}(mT)$ модели M или соответствующих переменных редуцированных моделей \bar{M} и $\bar{\bar{M}}$, которые по формулам (4.40), (4.43), (4.36) преобразуются далее к переменным полной модели M . Модуль формирования массивов осуществляет упорядоченное размещение в памяти ЭВМ

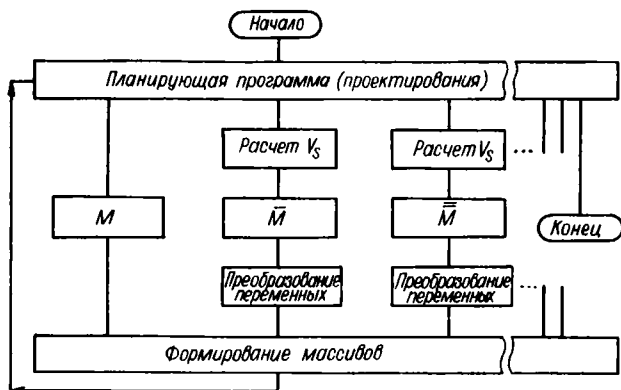


Рис. 4.17. Процедура расчета оптимальных процессов

полученных числовых значений в виде основных массивов $\mathbf{MX} = \{\mathbf{x}(mT)\}$ и $\mathbf{MU} = \{\mathbf{u}(mT)\}$, а также вспомогательных массивов $\mathbf{MP}_s = \{\mathbf{p}_{in}(mT)\}_s$, использующихся как массивы краевых условий для расчета участков разгона.

В ходе выполнения описанной процедуры формируются ансамбли разветвленных переходных процессов. Интегрирование модели M при различных последовательно перебираемых начальных значениях $\mathbf{p}_f \in S_f$ дает фазовые траектории, отличающиеся временем переключения управляющих воздействий. Такие траектории (рис. 4.18) имеют общий участок OA , соответствующий фиксированному значению управления $\mathbf{u} = \mathbf{U}_s$ и различные участки после переключения этого значения. При выборе в конечной точке другого начального значения (см. (4.53)) формируется другая ветвь оптимальных переходных процессов (линия OB). Среди допустимых значений $\mathbf{p}_f \in S_f$ можно выделить те, которые порождают $\mathbf{p} \in S_{in}$, т. е. отвечают условию перехода на граничную поверхность S_s , а также значения, при которых фазовая траектория покидает область допустимых состояний (участки, выходящие из R , далее не рассматриваются).

Траектории, удовлетворяющие условиям перехода на граничную поверхность, имеют отрезки, которые принадлежат S_s и рас-

считываются с помощью модели \bar{M} . Каждая рассчитанная точка является возможной точкой входа x_{in} и поэтому фиксируется в памяти ЭВМ вместе с элементом p_{in} вспомогательного массива MP_s . После попадания вектора состояния OY в процессе его движения по S_s на пересечение с другой граничной или вырожденной поверхностью S_{s+1} (точка B , рис. 4.19) осуществляется переход к интегрированию модели \bar{M}_{s+1} (линия BD) или \bar{M}_s , что соответствует движению по пересечению (линия BC).

Участки разгона формируются моделью M с использованием в качестве начальных значений элементов массивов MX и MP_s , полученных при движении по границе.

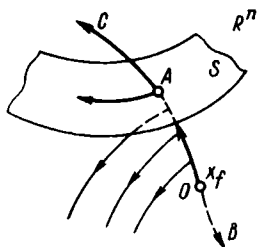


Рис. 4.18. Отрезки фазовых траекторий для различных значений p_f

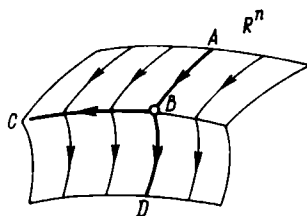


Рис. 4.19. Отрезки фазовых траекторий на граничных поверхностях

Процедура вычисления оптимальных траекторий заканчивается после перебора всех допустимых краевых значений: p_f в конечной точке x_f , элементов массивов MX и MP_s на поверхностях S_s . От принятого шага перебора зависит плотность найденных решений.

Итак, процедура расчета оптимальных переходных процессов является многоэтапной процедурой, обеспечивающей построение пучка разветвленных фазовых траекторий. Поэтому к блоку формирования массивов предъявляется требование рационального размещения результатов в памяти ЭВМ, позволяющего простейшим образом извлечь нужные элементы или отрезки хранящихся массивов.

Расчет завершает сортировка полученных результатов, в ходе которой выделяются массивы процессов, начинающихся в заданных начальных точках движения, и формируются массивы элементов, соответствующих поверхностям переключения. Перечень задач, решаемых в ходе сортировки, зависит от структурной модификации проектируемой САУ, т. е. выбранной стратегии программного или пространственного управления.

На начальном этапе сортировки выявляются экстремальные решения, связывающие одинаковые точки пространства состояния R^n . Из отобранных отрезков переходного процесса методом непосредственного сравнения значений функционала качества

(времени, топлива, энергии) выбираются оптимальные. Для выделения оптимального процесса, начинающегося из заданной точки x_0 , отыскивается ветвь массива \mathbf{MX} , содержащая x_0 или близкую к ней точку. После этого к найденной ветви последовательно пристыковываются отрезки траектории, приводящие в конечную точку x_f . Полученный в результате массив значений x , а также массив соответствующих значений вектора управления и параметризуются «прямым» временем $t = t_f - mT$.

Для формирования массивов значений вектора x , принадлежащего поверхности переключения (4.10), из \mathbf{MX} выделяются и объединяются в единый массив точки переключения, в которых происходит коммутация одной из координат вектора управления u . Другой способ позволяет сформировать массивы точек поверхностей переключения или их пересечений, являющихся многообразиями оптимального движения (торможения). В такие массивы объединяют значения x , полученные при торможении объекта с фиксированными значениями вектора управления $u = U_s$, а также значения, найденные на граничной поверхности при движении с постоянными значениями редуцированного вектора $\bar{u} = \bar{U}_s$, и т. д.

Как уже отмечалось, в алгоритм автоматизированного расчета переходных процессов целесообразно включить некоторые операции, осуществляемые обычно на этапе аналитической подготовки. В этом случае используется унифицированная структура алгоритма (см. рис. 4.17), содержащая по одной ветви для расчета движений по многообразиям S_s единой размерности. Приспособление такой ветви к конкретному многообразию осуществляется за счет изменения его аналитического описания (4.11), а основные компоненты алгоритма стабилизации (4.30), преобразований координат (4.40), (4.43) и канонической модели рассчитываются в ходе выполнения процедуры. При этом операции векторной алгебры и нахождения частных производных осуществляются с привлечением библиотечных подпрограмм; для расчета обратного преобразования $\Gamma(0, \bar{x})$ можно использовать линеаризованный алгоритм обращения (4.92).

Пример 4.7. Рассмотрим ход процедуры расчета переходных процессов для объекта (4.12). Канонические модели M и \bar{M} , полученные в примере 4.4, с учетом обратного времени интегрирования принимают вид:

$$M : \begin{cases} dx_1/d\tau = -x_2; & \begin{cases} dp_1/d\tau = 0; \\ dp_2/d\tau = p_1; \end{cases} \\ dx_2/d\tau = -\text{sign } p_2; \end{cases}$$

$$\bar{M} : d\bar{x}/d\tau = -1; \quad d\bar{p}/d\tau = 0$$

(остальные аналитические выражения, необходимые для расчета, приведены в примере 4.4).

Фазовые траектории в пространстве R^3 (кривые 1—6) и соответствующие им траектории в сопряженном пространстве представлены на рис. 4.20, а, б. Выбор

в конечной точке $u = U_3 = -1$ определяет одну из двух возможных плоскостей конечных значений (4.59)

$$S_{f_3}: p_3 + 1 = 0$$

и вместе с тем одну из двух ветвей оптимальных фазовых траекторий. Выбирая различные значения $p = \begin{bmatrix} p_{f1} \\ -1 \end{bmatrix} \in S_{f_3}$ при интегрировании M получим ансамбль переходных процессов, проходящих через x_f . Выделим четыре типа траекторий: 1 и 2 — получены при $p_{f1} \leq -1$ (на них переключение управляющего воздействия, вызванное изменением знака p_3 , происходит до достижения состоянием x граничной поверхности S_3);

3 — получена при $p_{f1} = -1$ (пересекает S_3 в момент переключения, когда $p \in S_{out}$, и удовлетворяет тем самым условию перехода на границу S_3);

4 — получена при $-1 < p_{f1} < 0$ (пересекает S_3 до переключения u и, следовательно, выходит из области допустимых состояний; участки, соответствующие $x_3 > 1$, из рассмотрения исключаются);

5 — получена при $p_{f1} \geq 0$ (пересекает S_3 и без переключения управления устремляется в бесконечность).

Расчет участка траектории 3, лежащего на S_3 , осуществляется с помощью редуцированной модели (4.55) с переменными состояния \bar{x} и \bar{p} . Этот участок в свою очередь порождает ансамбль участков разгона, интегрирование которых производится моделью M из точек $x_{out} \in S_3$ и $p_{out} \in S_{out}$ (в рассматриваемом примере S_{in} и S_{out} совпадают). На рис. 4.20, а приведены две траектории разгона 6 и 7. Для полноты картины в сопряженном пространстве (рис. 4.20, б) также показаны траектории 6 и 7, которые на участке движения по границе образованы с помощью полной модели (4.54). Зигзагом отмечены скачки переменной p_3 , необходимые для перехода к участку разгона.

Пример 4.8. Рассмотрим оптимальные переходные процессы ОУ (4.60), проходящие через конечную точку $x_f = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Канонические модели (см. примеры 4.5, 4.6) принимают вид:

$$M: \begin{cases} dx_1/d\tau = x_1 x_2^2 - x_2; \\ dx_2/d\tau = x_3 - \text{Sign } p_2; \end{cases} \begin{cases} dp_1/d\tau = -x_2^2 p_1; \\ dp_2/d\tau = -(2x_1 x_2 - 1) p_1 - p_3; \end{cases}$$

$$\bar{M}: d\bar{x}/d\tau = \bar{x}^3; \quad \bar{p} = -1/\bar{x}^3$$

(остальные соотношения, необходимые для расчета оптимальных траекторий, приведены в примерах 4.5 и 4.6).

Оптимальные фазовые траектории в пространстве состояний и сопряженном пространстве представлены соответственно на рис. 4.21, а и б. Выбор различных крайних значений $p_f \in S_f$ позволяет получить разные типы траекторий:

1, 2, на которых переключение знака p_2 и, следовательно, управления u происходит до достижения вектором x множества S_1 ;

3, на которой при $p_3 = 0$ $x \in S_1$ и выполняются условия перехода на вырожденную линию S_1 , где $p_3 \equiv 0$;

4, на которой $p_3 < 0$ и переключения управления u не происходит.

Все точки отрезка траектории 3, лежащего на множестве S_1 , могут быть использованы для образования отрезков разгона 5, 6. Отметим, что в отличие от оптимальных процессов, рассмотренных в примере 4.7, в данной задаче скачка сопряженных переменных не происходит, что является особенностью движения по вырожденным многообразиям.

Аппроксимация оптимальных решений. Массивы оптимальных переходных процессов позволяют получить аналитическое описание функций $x^*(t)$ и $u^*(t)$, необходимое для построения систем программного управления, а массивы поверхностей переключения — описание этих поверхностей, используемое для синтеза

алгоритма переключения структуры и локальных алгоритмов пространственного управления. Основным аппаратом формирования математических моделей, соответствующих массиву числовых значений неизвестной функции, служит процедура аппроксимации (см., например, [26, 34]). Аппроксимация сводится к нахождению функции, проходящей через множество точек, или точнее функции, по определенному критерию наиболее близкой к этим точкам.

В качестве аппроксимирующей функции переходного процесса $x^*(t)$ выберем выражение

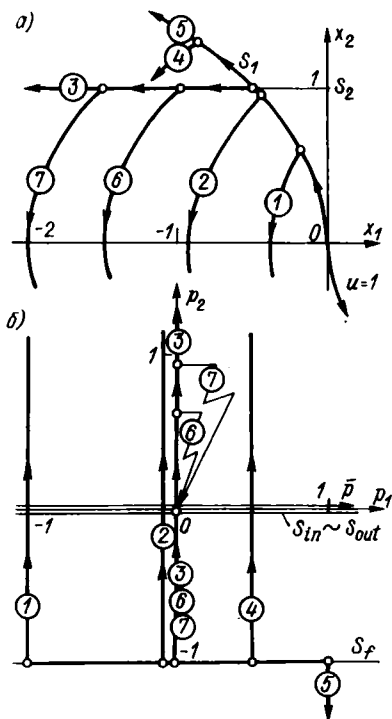


Рис. 4.20. Оптимальные фазовые траектории для линейного ОУ (4.12): а — в пространстве состояний; б — в сопряженном пространстве

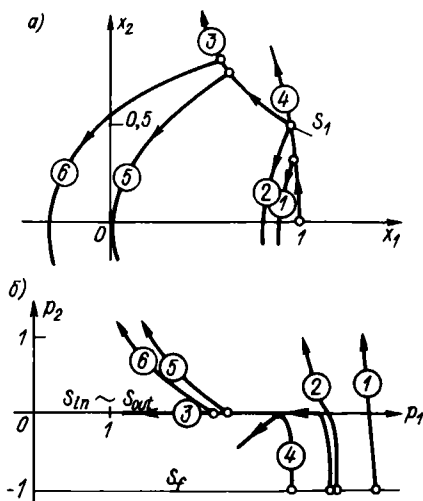


Рис. 4.21. Оптимальные фазовые траектории для нелинейного ОУ (4.60): а — в пространстве состояний; б — в сопряженном пространстве

$$x_a(t) = C^T z(t) = \sum_{k=1}^{k_1} C_k z_k(t),$$

где $z_k(t)$ — базисные функции; C_k — векторные коэффициенты аппроксимации.

От выбора системы базисных функций и числа членов разложения зависит точность результатов аппроксимации. Наиболее распространены степенные аппроксимирующие полиномы, где

$$z(t) = [\dots t^{-2} t^{-1} t^0 t^1 t^2 \dots]^T.$$

Векторные коэффициенты C_k находятся в процессе аппроксимации из условия наилучшего совпадения $x^*(t)$ и $x_a(t)$. Крите-

рием близости обычно служит интегральная квадратичная функция, а в рассматриваемом случае, когда известны только дискретные значения $x^*(mT)$, $m = m_1, m_2$, — суммарная ошибка квадратов расхождения координат вектора состояния в точках $t = mT$.

Ограничимся рассмотрением скалярного случая $x \in R^1$ или аппроксимацией одной из координат многомерного вектора x . Сформируем вектор размерности $m_2 - m_1 + 1$

$$Z_0 = [x^*(m_1T) \mid x^*((m_1 + 1)T) \mid \dots \mid x^*(m_2T)]^T$$

и матрицу базисных векторов

$$Z = [z(m_1T) \mid z((m_1 + 1)T) \mid \dots \mid z(m_2T)].$$

Суммарный критерий аппроксимации зададим в виде

$$J_a = 0,5\varepsilon^T Q_a \varepsilon, \quad (4.83)$$

где $\varepsilon = Z_0 - Z^T C$ — ошибка; Q_a — весовая матрица (обычно $Q_a = I$).

Задача минимизации (4.83) на множестве значений вектора решается из условия $dJ_a/dC = 0$, что позволяет получить [26]

$$C = (Z Q_a Z^T)^{-1} Z Q_a Z_0. \quad (4.84)$$

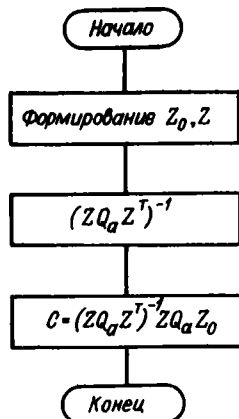


Рис. 4.22. Процедура аппроксимации

Процедура аппроксимации приведена на рис. 4.22. Наиболее трудоемким элементом процедуры является обращение матрицы $Z Q_a Z^T$, размерность которой растет с увеличением числа аппроксимируемых точек.

Обращение можно исключить при использовании поисковой процедуры минимизации функции (4.83), т. е.

$$C(l+1) = C(l) - K_a (dJ/dC)^T,$$

где l — шаг итерации; K_a — матрица, определяющая скорость поискового движения; $(dJ/dC)^T$ — градиент функционала (4.83).

Дифференцируя (4.83), найдем

$$\left(\frac{dJ}{dC}\right)^T = -Z^T Q_a \varepsilon$$

и, следовательно, процедура минимизации принимает вид

$$C(l+1) = C(l) + K_a Z^T Q_a \varepsilon. \quad (4.85)$$

Матрица K_a находится из условия оптимизации поиска [80]

$$K_a = \frac{(dJ/dC) [C - C(l)]}{\|dJ/dC\|^2} = -\frac{\varepsilon^T Q_a \varepsilon}{\|Z^T Q_a \varepsilon\|^2} I$$

и обеспечивает сходимость процедуры с заданной степенью точности за конечное число итераций.

Поисковая процедура аппроксимации представлена на рис. 4.23. Аналогично осуществляется аппроксимация гладких отрезков функции $u^*(t)$.

Отметим, что для систем, характеризующихся разрывным управлением, наилучшие результаты получаются при кусочной аппроксимации. Массивы MX и MU разбиваются на области, соответствующие гладким отрезкам функции $x^*(t)$, $u^*(t)$, и для каждой области отыскиваются свои коэффициенты аппроксимации, как это показано в примере 4.3.

Рассмотрим аппроксимацию m_3 точек $x = x_m$ поверхности переключения (4.10). Для ее описания будем использовать выражение

$$z_0 x - C^T z(x) = 0, \quad (4.86)$$

где C — вектор коэффициентов аппроксимации; $z(x) = [z_1 \mid z_2 \mid \dots \mid z_{k_1}]^T$; $z_k(x)$ ($k = 0, k_1$) — базисные функции, а в частном случае — степенные функции координат вектора x [26], тогда

$$[z_0 \mid z^T] = [1 \mid x_1 \mid x_2 \dots \mid x_1 x_2 \mid x_1^2 \mid x_2^2 \mid \dots].$$

Сформируем вектор Z_0 размерности m_3

$$Z_0 = [z_0(x_1) \mid z_0(x_2) \mid \dots \mid z_0(x_{m_3})]^T$$

и матрицу Z размерности $k_1 \times m_3$

$$Z = [z(x_1) \mid z(x_2) \mid \dots \mid z(x_{m_3})].$$

Критерий аппроксимации зададим в виде (4.83), где $e = Z_0 - Z^T C$. Минимизируя (4.83), получим

$$C = (ZQ_a Z^T)^{-1} ZQ_a Z_0.$$

Для исключения обращения матрицы $ZQ_a Z^T$ используется итерационная процедура (4.85). Алгоритмы аппроксимации поверхности соответствуют представленным на рис. 4.22, 4.23.

Рассмотренные процедуры обеспечивают необходимые для синтеза САУ описания программных переходных процессов и оптимальных многообразий торможения. Уравнения вырожденных и граничных многообразий могут быть получены аналогично. Однако их точное описание обычно находится аналитически, а процедура аппроксимации может быть применена лишь для упрощения достаточно сложных математических выражений.

4.5. Синтез локальных регуляторов

Основу алгоритмического обеспечения оптимальных систем, решающих терминальную задачу, составляют алгоритмы программного или пространственного управления, последовательно

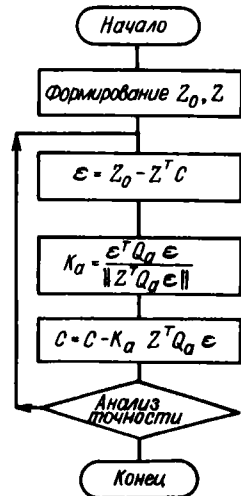


Рис. 4.23. Поисковая процедура аппроксимации

подключаемые к объекту в ходе его функционирования. Программное управление обеспечивает наилучшее приближение к оптимальному переходному процессу $x^*(t)$, так как контролирует состояние объекта управления по всем координатам пространства R^n . Однако для систем с широким диапазоном изменения начальных состояний ОУ требуется построение генератора командных сигналов, что усложняет реализацию САУ и несколько снижает ее точность ввиду естественных погрешностей численного интегрирования нелинейного дифференциального уравнения КГ.

Наиболее простое решение задачи как с точки зрения структуры синтезируемого регулятора, так и размерности его функциональных элементов предлагает метод пространственного управления. Упрощение достигается за счет вполне допустимого разомкнутого характера управления разгоном и продольным движением. Кроме чисто практических преимуществ, связанных с возможностью упрощения программного обеспечения цифровой САУ и уменьшения интервала дискретности, необходимо отметить гармоничное сочетание пространственного подхода с общей стратегией синтеза оптимальной системы, основанной на использовании топологических свойств пространства состояний R^n . Принимая во внимание перечисленные обстоятельства, в дальнейшем ограничимся рассмотрением алгоритмов пространственного управления.

Постановка задачи. На временном интервале $t \in [t_1, t_2]$ проанализируем динамику движения нелинейного объекта (4.1) по гладкому регулярному многообразию S размерности $n - v$,

$$S: \Phi(x) = 0, \quad (4.87)$$

где $\Phi(x) \in R^v$.

Будем полагать, что для $x \in S$ (локально) функциональные компоненты $f(x)$, $G(x)$ и $\Phi(x) = \{\varphi_i(x)\}$ — гладкие; $\Phi(x)$ — удовлетворяет условию регулярности [67]

$$\text{rang} [\partial\Phi/\partial x] = v,$$

а многообразие (4.87) для ОУ (4.1) в силу процедуры его построения является интегральным.

Осуществим нелинейное преобразование координат объекта управления (4.1). Для этого определим *вектор относительного движения* или отклонение от S , как

$$e = \Phi(x) \quad (4.88)$$

и *вектор продольного движения*

$$\bar{x} = \Psi(x), \quad (4.89)$$

где $\Psi(x) \in R^{n-v}$ — гладкая вектор-функция, выбираемая так, чтобы для любых $x \in S$ выполнялось

$$\det \left[\frac{\partial\Phi/\partial x}{\partial\Psi/\partial x} \right] \neq 0.$$

Тогда в соответствии с теоремой об обратном отображении [35] имеет место следующее свойство.

С в о й с т в о 4.1. Существуют окрестность X^o точки $x^o \in S$ и гладкая вектор-функция $\Gamma(e, \bar{x})$ такие, что для любых $x \in X^o$ выполняется

$$x = \Gamma(e, \bar{x}). \quad (4.90)$$

Аналитически обратное преобразование $\Gamma(e, \bar{x})$ обычно получить сложно. Поэтому напоминая, что функции $\Phi(x)$, $\Psi(x)$ и $\Gamma(e, \bar{x})$ связаны выражением вида (4.41), запишем линеаризацию преобразования (4.90) в малой окрестности произвольной точки $x' \in X^o$

$$\Gamma(e, \bar{x}) \cong x' - \frac{\partial \Gamma}{\partial x} \Psi(x') + \left[\frac{\partial \Gamma}{\partial e} \mid \frac{\partial \Gamma}{\partial \bar{x}} \right] \begin{bmatrix} e \\ \bar{x} \end{bmatrix}. \quad (4.91)$$

Для точек многообразия S , т. е. $x \in S$, из (4.91) найдем

$$\Gamma(0, \bar{x}) \cong x' + \frac{\partial \Gamma}{\partial x} [\bar{x} - \Psi(x')]. \quad (4.92)$$

Будем рассматривать движение объекта управления (4.1) в такой δ -окрестности многообразия S , в которой выполняется свойство 4.1, т. е. в области

$$X: x \in X^o; \quad \|e\| \leq \delta,$$

где $\delta > 0$, $\|e\|$ — евклидова норма вектора e .

Определим множество допустимых значений вектора \bar{x} как

$$\bar{X}: \bar{x} \in R^n; \quad \Gamma(e, \bar{x}) \in X.$$

Будем полагать, что начальное значение вектора относительного движения $e_0 = \Phi(x(t_1))$ удовлетворяет условию $\|e_0\| \leq \delta$. Тогда соответствующий выбор x^o всегда позволяет обеспечить $x(t_1) \in X$ и $\bar{x}(t_1) \in \bar{X}$. Множества X и \bar{X} в общем случае ограничены.

Сформулируем задачу синтеза.

Задача 4.4. Найти замкнутый алгоритм управления, обеспечивающий для ОУ (4.1), интегрального многообразия (4.87), любых $t \in [t_1, t_2]$ и $x(t_1) \in X$ стабилизацию нулевого значения вектора относительного движения.

При нулевых начальных отклонениях $e_0 = 0$ алгоритм управления на всем интервале $[t_1, t_2]$ должен обеспечить выполнение тождества $e \equiv 0$. В общем случае, когда $e_0 \neq 0$, необходимо добиться сходимости к нулю переходных процессов по отклонению e или, иначе говоря, устойчивости относительного движения и устойчивости многообразия (4.87). Отметим, что наиболее распространенное требование асимптотической устойчивости в условиях ограниченности временного интервала $[t_1, t_2]$ и множества X становится некорректным. Поэтому конкретизируя задачу ста-

билизации, выдвинем требование экспоненциальной устойчивости, методы достижения которой более подробно обсуждаются в п. 4.6.

Управление движением по гиперповерхности. Рассмотрим поведение объекта управления (4.1) на гиперповерхности или на многообразии размерности $n - 1$, полагая $\Phi(\mathbf{x}) = \varphi(\mathbf{x})$ (см. (4.3)); в этом случае e — скалярная функция.

Осуществим преобразование вектора управления к виду (см. п. 4.2)

$$\mathbf{u} = \mathbf{E}_1 \bar{\mathbf{U}} + \mathbf{E}_2 u_e, \quad (4.93)$$

где $\bar{\mathbf{U}}$ — постоянный вектор размерности $h - 1$ с координатами ± 1 , выбор которого осуществлен из условия оптимальности продольного движения; u_e — скалярное стабилизирующее воздействие.

Дифференцируя (4.88) и подставляя (4.93), найдем уравнение относительного движения

$$\dot{e} = \alpha(\mathbf{x}) + \mathbf{b}^\top(\mathbf{x}) \bar{\mathbf{U}} + \beta(\mathbf{x}) u_e, \quad (4.94)$$

где значения компонент $\alpha(\mathbf{x})$, $\mathbf{b}^\top(\mathbf{x})$, $\beta(\mathbf{x})$ определены формулами (4.29).

Используя свойство 4.1, преобразуем полученное выражение. Предварительно осуществим линеаризацию его компонент при $e \cong 0$:

$$\alpha(\mathbf{x}) = \alpha_0(\bar{\mathbf{x}}) + \alpha_1(\bar{\mathbf{x}})e + o_1(\bar{\mathbf{x}}, e);$$

$$\mathbf{b}^\top(\mathbf{x}) = \mathbf{b}_0^\top(\bar{\mathbf{x}}) + \mathbf{b}_1^\top(\bar{\mathbf{x}})e + \mathbf{O}(\bar{\mathbf{x}}, e);$$

$$\beta(\mathbf{x}) = \beta_0(\bar{\mathbf{x}}) + \beta_1(\bar{\mathbf{x}})e + o_2(\bar{\mathbf{x}}, e),$$

где $\alpha_0(\bar{\mathbf{x}}) = \alpha[\Gamma(0, \bar{\mathbf{x}})]$; $\mathbf{b}_0(\bar{\mathbf{x}}) = \mathbf{b}[\Gamma(0, \bar{\mathbf{x}})]$; $\beta_0(\bar{\mathbf{x}}) = \beta[\Gamma(0, \bar{\mathbf{x}})]$; $\alpha_1(\bar{\mathbf{x}}) = \frac{\partial \alpha}{\partial e} \Big|_{e=0}$; $\mathbf{b}_1(\bar{\mathbf{x}}) = \frac{\partial \mathbf{b}}{\partial e} \Big|_{e=0}$; $\beta_1(\bar{\mathbf{x}}) = \frac{\partial \beta}{\partial e} \Big|_{e=0}$; $o_1 \rightarrow 0$, $o_2 \rightarrow 0$ и $\mathbf{O} \rightarrow 0$ при $e \rightarrow 0$.

Полагая, что δ достаточно мало, и пренебрегая влиянием o_1 , o_2 , \mathbf{O} , из (4.94) получим [58, 59]:

$$\begin{aligned} \dot{e} = & [\alpha_1(\bar{\mathbf{x}}) + \mathbf{b}_1^\top(\bar{\mathbf{x}}) \bar{\mathbf{U}}] e + \alpha_0(\bar{\mathbf{x}}) + \mathbf{b}_0^\top(\bar{\mathbf{x}}) \bar{\mathbf{U}} + \\ & + [\beta_0(\bar{\mathbf{x}}) + \beta_1(\bar{\mathbf{x}}) e] u_e. \end{aligned} \quad (4.95)$$

Итак, в малой окрестности многообразия S относительное движение ОУ (4.1) описывается линейным дифференциальным уравнением (4.95), которое и будем использовать для синтеза управления u_e . Правая часть (4.95) содержит функцию $\alpha_0 + \mathbf{b}_0^\top \bar{\mathbf{U}}$, возмущающее влияние которой необходимо скомпенсировать управляющим воздействием u_e . Динамика свободной составляющей относительного движения определяется функцией $\alpha_1 + \mathbf{b}_1^\top \bar{\mathbf{U}}$. Так как в общем случае она может оказаться положительной, то при синтезе алгоритма управления следует пред-

усмотреть введение обратной связи по e , обеспечивающей устойчивость нулевого решения уравнения (4.95).

Для получения условия разрешимости задачи 4.4 проанализируем поведение объекта управления (4.1) на поверхности S . При $e \equiv 0$ из (4.95) получим

$$\alpha_0(\bar{\mathbf{x}}) + \mathbf{b}_0^\top(\bar{\mathbf{x}}) \bar{\mathbf{U}} + \beta_0(\bar{\mathbf{x}}) u_e = 0, \quad (4.96)$$

причем $\bar{\mathbf{x}}$ является решением уравнения продольного движения (4.44).

Так как рассматриваемое многообразие в силу постановки задачи 4.4 является для ОУ (4.1) интегральным, то решение уравнения (4.96) существует. Проанализируем различные варианты решения.

Если для всех $\mathbf{x} \in X$ и, следовательно, $\bar{\mathbf{x}} \in \bar{X}$, $\beta_0(\bar{\mathbf{x}}) \neq 0$, то (4.96) имеет единственное решение $u_e = V(\bar{\mathbf{x}})$, где

$$V(\bar{\mathbf{x}}) = -[\alpha_0(\bar{\mathbf{x}}) + \mathbf{b}_0^\top(\bar{\mathbf{x}}) \bar{\mathbf{U}}] / \beta_0(\bar{\mathbf{x}}). \quad (4.97)$$

В то же время можно предположить, что на поверхности S найдутся значения $\mathbf{x} = \Gamma(0, \bar{\mathbf{x}})$, для которых $\beta_0(\bar{\mathbf{x}}) = 0$. Множество таких значений задает в R^n многообразие S' размерности $n - 2$ (рис. 4.24), описываемое уравнением

$$\left[\frac{\Phi(\mathbf{x})}{\beta_0(\Psi(\mathbf{x}))} \right] = 0. \quad (4.98)$$

На S' уравнение (4.96) принимает вид

$$\alpha_0(\bar{\mathbf{x}}) + \mathbf{b}_0^\top(\bar{\mathbf{x}}) \bar{\mathbf{U}} = 0,$$

(в противном случае $e \neq 0$ и задача 4.4 решения не имеет), т. е. стабилизирующая составляющая u_e может быть выбрана произвольно.

В частности, возможно доопределение

$$u_e = \lim_{\beta_0(\bar{\mathbf{x}}) \rightarrow 0} V(\bar{\mathbf{x}}),$$

при котором непрерывность u_e и, следовательно, $\mathbf{x}^*(t)$ сохраняется (траектория ABC).

С другой стороны, многообразие $S' \subset S$ может тоже оказаться оптимальным. В этом случае после попадания состояния ОУ в его окрестность управление должно быть найдено из условия стабилизации движения в S' (траектория ABD). Задача движения по многообразиям рассмотрена ниже.

Для произвольных достаточно малых e выберем управление в форме комбинированного закона

$$u_e = V(\bar{\mathbf{x}}) - k(\bar{\mathbf{x}})e, \quad (4.99)$$

где $k(\bar{\mathbf{x}})$ — коэффициент обратной связи по отклонению.

Уравнение относительного движения (4.95) принимает вид

$$\dot{e} = \alpha_c(\bar{x}) e, \quad (4.100)$$

где $\alpha_c(\bar{x}) = \alpha_1 + b_0^T \bar{U} + \beta_1 V - \beta_0 k$.

Будем выбирать $k(\bar{x})$ так, чтобы для всех $\bar{x} \in \bar{X}$ и $t \in [t_1, \infty)$ выполнялось

$$\alpha_c(\bar{x}) \leq -\eta, \quad (4.101)$$

где $\eta > 0$.

Тогда достигается экспоненциальная устойчивость решения (4.100) со степенью η [4,92], т. е. при $|e_0| \leq \delta$

$$|e| \leq \delta \exp(-\eta t).$$

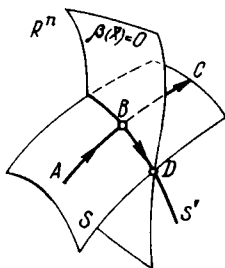


Рис. 4.24. Фазовые траектории (неединственное решение)

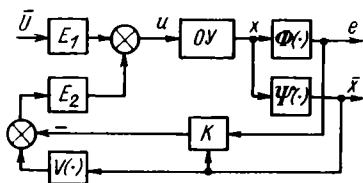


Рис. 4.25. Структура системы пространственного управления

В частном случае, соответствующем равенству в (4.101), получим

$$k(\bar{x}) = [\eta + \alpha_1(\bar{x}) + b_1^T(\bar{x}) \bar{U} + \beta_1(\bar{x}) V(\bar{x})] / \beta_0(\bar{x}).$$

Итак, из (4.93) и (4.99) найдем выражение, описывающее локальный регулятор (алгоритм управления),

$$P: u = E_1 \bar{U} + E_2 [V(\bar{x}) - k(\bar{x}) e]. \quad (4.102)$$

Структура системы пространственного управления представлена на рис. 4.25.

Напомним, что в п. 4.2 был получен алгоритм стабилизации движения по поверхности (4.30), использующийся при расчете оптимальных переходных процессов. Естественно, что на поверхности S значения стабилизирующих воздействий, рассчитанные по (4.30) и (4.99), совпадают. В то же время приведенные здесь формулы отличаются от полученных ранее, во-первых, использованием линеаризованного выражения (4.97) вместо точного соотношения (4.30) и, во-вторых, введением обратной связи по отклонению $k(\bar{x}) e$. Это упрощает расчеты функции V и обеспечивает экспоненциальную устойчивость пространственного движе-

ния. Для процедуры вычисления оптимальных переходных процессов вопросы устойчивости не имели значения в силу того, что на поверхности S использовались редуцированная модель M и преобразование $\mathbf{x}^* = \Gamma(0, \bar{\mathbf{x}}^*)$, для которого всегда $e = \Phi[\Gamma(0, \bar{\mathbf{x}}^*)] \equiv 0$. Для уточнения алгоритма с учетом дискретности процессов в цифровой системе приведем уравнение относительного движения (4.95) к рекуррентной форме. Подставляя (4.102) в (4.95) и пренебрегая бесконечно малыми второго порядка, получим

$$\dot{e} = [\alpha_1(\bar{\mathbf{x}}) + \mathbf{b}_1^T(\bar{\mathbf{x}})\bar{\mathbf{U}} + \beta_1(\bar{\mathbf{x}})V(\bar{\mathbf{x}})]e - \beta_0(\bar{\mathbf{x}})k(\bar{\mathbf{x}})e.$$

Проинтегрируем полученное уравнение в предположении, что $\bar{\mathbf{x}}(t) = \text{const}$ при $t \in [mT, (m+1)T]$ и, рассматривая решения в моменты $t = mT$ и $(m+1)T$, запишем

$$e_{m+1} = \alpha_d(\bar{\mathbf{x}}_m)e_m - \beta_d(\bar{\mathbf{x}}_m)k(\bar{\mathbf{x}}_m)e_m,$$

где

$$\alpha_d(\bar{\mathbf{x}}_m) = \exp\{[\alpha_1 + \mathbf{b}_1^T\bar{\mathbf{U}} + \beta_1V]T\};$$

$$\beta_d(\bar{\mathbf{x}}_m) = (\exp\{[\alpha_1 + \mathbf{b}_1^T\bar{\mathbf{U}} + \beta_1V]T\} - 1)\beta_0.$$

Коэффициент обратной связи $k(\bar{\mathbf{x}}_m)$ находится из условия экспоненциальной устойчивости замкнутой дискретной системы

$$e_{m+1} = \alpha_{dc}(\bar{\mathbf{x}}_m)e_m. \quad (4.103)$$

Здесь $\alpha_{dc}(\bar{\mathbf{x}}_m) = \alpha_d - \beta_d k$, т. е. так, чтобы

$$|\alpha_{dc}| \leq \eta_d,$$

где $0 \leq \eta_d < 1$.

Тогда имеет место следующая оценка переходного процесса системы (4.103):

$$|e_m| \leq \delta \eta_d^m$$

и дискретный процесс сходится со степенью затухания η_d .

Важно отметить, что в режиме торможения движение системы обычно осуществляется вдоль поверхности переключения, где стабилизирующая компонента u_e одновременно выполняет функции оптимизации движения в R^n и $V = \pm 1$. Поэтому алгоритм (4.102) удобно записать как

$$\mathbf{u} = \mathbf{U} - E_2 k(\bar{\mathbf{x}})e$$

(\mathbf{U} — вектор размерности h с координатами ± 1 ; см. замечание 2 к теореме 4.1), и задача синтеза регулятора сводится к нахождению коэффициента обратной связи $k(\bar{\mathbf{x}})$. Однако, так как величина $k(\bar{\mathbf{x}})e$ может принимать значения любого знака, то по соответствующей координате $u_e = \pm 1 - k(\bar{\mathbf{x}})e$ вектор управления \mathbf{u} может выйти из области допустимых значений Γ . Если такое на-

рушение условия (4.2) не допустимо даже на коротких интервалах времени, то при расчете оптимальных переходных процессов и многообразий предусматривают ужесточение ограничений на управление вида

$$|u_j| \leq 1_{\Delta},$$

где $0 < 1_{\Delta} < 1$ (см. п. 4.1 и рис. 4.4).

Тогда $V = \pm 1_{\Delta}$ и соответствующий выбор значения 1_{Δ} позволяют исключить возможное нарушение неравенств (4.2). В дальнейшем будем полагать, что такое уточнение произведено.

Пример 4.8. Найдем алгоритм управления движением ОУ (4.60) по вырожденному многообразию S_1 (4.74). В рассматриваемом одноканальном случае

$$u = V(\bar{x}) - k\epsilon, \quad (4.104)$$

где $\epsilon = x_2 - 1/2x_1$; $\bar{x} = x_2$.

Линеаризируя уравнение (4.78) для малых ϵ , получим

$$\dot{\epsilon} = -\epsilon + \bar{x}(\bar{x}^2 - 1) + u. \quad (4.105)$$

При $\epsilon \equiv 0$ найдем

$$u = V(\bar{x}) = -\bar{x}(\bar{x}^2 - 1).$$

Выбирая $k = 2$, получим

$$\epsilon = -3\epsilon,$$

что гарантирует экспоненциальную устойчивость переходного процесса со степенью $\eta = 3$.

Дискретный аналог (4.105) для найденной структуры алгоритма (4.104) имеет вид

$$\epsilon_{m+1} = \exp(-T)\epsilon_m + [\exp(-T) - 1]k\epsilon_m. \quad (4.106)$$

Пусть $T = 0,1$; полагая $k = 2$, получим

$$\epsilon_{m+1} = 0,71\epsilon_m,$$

т. е. дискретный переходный процесс сходится со степенью затухания $\eta_d = 0,71$.

Пример 4.9. Рассмотрим алгоритм управления движением ОУ (4.60) по линии переключения S_2 (см. рис. 4.14), достаточно точно описываемой аппроксимированным выражением [58],

$$S_2: -1 + x_1 + 0,15x_2 = 0.$$

Определим новые переменные состояния:

$$\epsilon = -1 + x_1 + 0,15x_2; \quad \bar{x} = x_2.$$

Уравнение относительного движения (4.94) имеет параметры $\alpha = (2 - x_1x_2)x_2$, $\beta = 0,15$. Выполнив линеаризацию α и β при малых ϵ , получим

$$\dot{\epsilon} = -\bar{x}^2\epsilon + 0,85\bar{x} - \bar{x}^3 + 0,15\bar{x}^3 + 0,15u.$$

Из условия $\epsilon \equiv 0$ найдем

$$u = V(\bar{x}) = -6,7\bar{x}(0,85 - \bar{x} + 0,15\bar{x}^2).$$

Используя алгоритм (4.104), получим замкнутую систему (4.100), где $\alpha_c = -0,15k - \bar{x}_2$. Выбирая $k = 20$, найдем $\alpha_c \leq -3$ и $\eta = 3$.

Управление движением по многообразию произвольной размерности. Проведем анализ движения по многообразию S (4.87)

размерности $n - v$. Управляющее воздействие будем формировать в виде (4.93), где $u_e \in R^{h_e}$ — вектор стабилизирующих управлений, \bar{U} — постоянный вектор размерности $h_x = h - h_e$.

Дифференцируя (4.88) по времени с учетом уравнения ОУ (4.1) и структуры управления (4.93), получим уравнение относительного движения

$$\dot{e} = a(x) + B_x(x)\bar{U} + B_e(x)u_e,$$

$$\text{где } a(x) = \frac{\partial \Phi}{\partial x} f; \quad B_x(x) = \{b_{xj}\} = \frac{\partial \Phi}{\partial x} E_1 G; \quad j = \overline{1, h_x}; \quad B_e(x) = \{b_{ej}\} = \frac{\partial \Phi}{\partial x} E_2 G; \quad j = \overline{1, h_e}.$$

Используя преобразование координат (4.88), (4.89), осуществим линеаризацию компонент последнего уравнения на множестве X :

$$\dot{e} = \left[A_1(x) + \sum_{j=1}^{h_x} B_{xj}(\bar{x}) \bar{U}_j \right] e + a_0(\bar{x}) + B_{x0}(\bar{x}) \bar{U} + \sum_{j=1}^{h_e} \{ [B_{ej}(\bar{x})e + b_{ej}(\bar{x})] u_{ej} \}, \quad (4.107)$$

$$\text{где } a_0(\bar{x}) = a(\Gamma(0, \bar{x})), \quad B_{x0}(\bar{x}) = B_x(\Gamma(0, \bar{x})), \quad B_{e0}(\bar{x}) = \{b_{ej}\} = B_e(\Gamma(0, \bar{x})), \quad A_1(\bar{x}) = \left. \frac{\partial a}{\partial e} \right|_{e=0}, \quad B_{xj}(\bar{x}) = \left. \frac{\partial b_{xj}}{\partial e} \right|_{e=0}, \quad B_{ej}(\bar{x}) = \left. \frac{\partial b_{ej}}{\partial e} \right|_{e=0}.$$

Выбор управления $u_e = \{u_{ej}\}$ производится из условия компенсации возмущающего влияния вектора $a_0 + B_{x0}\bar{U}$ и обеспечения устойчивости свободной составляющей относительного движения. Для этого запишем уравнение (4.107) на многообразии S , т. е. при $e = 0$:

$$a_0(\bar{x}) + B_{x0}(\bar{x})\bar{U} + B_{e0}(\bar{x})u_e = 0. \quad (4.108)$$

Отметим, что поведение вектора продольного движения \bar{x} описывается уравнением вида

$$\dot{\bar{x}} = \bar{f}(\bar{x}) + \bar{G}(\bar{x})\bar{U}, \quad (4.109)$$

где матричные компоненты $\bar{f}(\bar{x})$ и $\bar{G}(\bar{x})$ определены в комментариях к (4.44).

По условию задачи 4.4 многообразие S является интегральным и поэтому решение уравнения (4.108) $u_e = V(\bar{x})$ существует. Тогда по теореме Кронекера—Капелли [4,42] выполняется следующее условие.

Свойство 4.2. Для любых $\bar{x} \in \bar{X}$

$$\text{rang}[a_0 + B_{x0}\bar{U} \mid B_{e0}] = \text{rang } B_{e0}.$$

В первую очередь рассмотрим случай, когда в \bar{X} $\text{rang } \mathbf{B}_{e0} = h_e$. Тогда уравнение (4.108) имеет решение

$$\mathbf{u}_e = \mathbf{V}(\bar{\mathbf{x}}) = -\mathbf{B}_{e0}^+ [\mathbf{a}_0(\bar{\mathbf{x}}) + \mathbf{B}_{x0}(\bar{\mathbf{x}}) \bar{\mathbf{U}}], \quad (4.110)$$

где \mathbf{B}_{e0}^+ — псевдообратная матрица ($\mathbf{B}_{e0}^+ = (\mathbf{B}_{e0}^T \mathbf{B}_{e0})^{-1} \mathbf{B}_{e0}^T$).

Для компонентов уравнений (4.107) и (4.108) выполняется легко проверяемое свойство.

Свойство 4.3. Для любых $\bar{\mathbf{x}} \in \bar{X}$ таких, что $\text{rang } \mathbf{B}_{e0} = h_e$

$$[\mathbf{I} - \mathbf{B}_{e0} \mathbf{B}_{e0}^+] [\mathbf{a}_0 + \mathbf{B}_{x0} \bar{\mathbf{U}}] = 0.$$

Для произвольных векторов \mathbf{x} из множества X , удовлетворяющих условию $\|\mathbf{e}\| \leq \delta$, воспользуемся комбинированным алгоритмом стабилизации движения по многообразию

$$\mathbf{u}_e = \mathbf{V}(\bar{\mathbf{x}}) - \mathbf{K}(\bar{\mathbf{x}}) \mathbf{e}, \quad (4.111)$$

где $\mathbf{V}(\bar{\mathbf{x}})$ определяется выражением (4.110) и обеспечивает компенсацию возмущающих факторов уравнения относительного движения (4.107), $\mathbf{K}(\bar{\mathbf{x}})$ — матрица обратных связей по отклонению от многообразия S .

Уравнение замкнутой системы (4.107) с учетом свойства 4.3 можно записать в виде

$$\dot{\mathbf{e}} = \mathbf{A}_c(\bar{\mathbf{x}}) \mathbf{e}, \quad (4.112)$$

где

$$\mathbf{A}_c(\bar{\mathbf{x}}) = \mathbf{A}_1 + \sum_{j=1}^{h_x} \mathbf{B}_{0j} \bar{\mathbf{U}}_j + \sum_{j=1}^{h_e} \mathbf{B}_{ej} V_j - \mathbf{B}_{e0} \mathbf{K}; \quad \{V_j\} = \mathbf{V}(\bar{\mathbf{x}}).$$

Таким образом, задача стабилизации пространственного движения сводится к выбору матрицы обратных связей $\mathbf{K}(\bar{\mathbf{x}})$, обеспечивающей устойчивость линейной нестационарной системы (4.112), параметры которой зависят от вектора состояния редуцированной системы (4.109). Специфика задачи устойчивости требует особого рассмотрения (см. п. 4.6).

Перейдем к анализу случая, когда в \bar{X} найдутся множество значений $\bar{\mathbf{x}}$, для которых $\text{rang } \mathbf{B}_{e0}(\bar{\mathbf{x}}) < h_e$. При этом схема синтеза управления изменяется. Здесь возможны две ситуации:

1) условие $\text{rang } \mathbf{B}_{e0}(\bar{\mathbf{x}}) < h_e$ выполняется на многообразии $S' \in \bar{X}$;

2) условие $\text{rang } \mathbf{B}_{e0}(\bar{\mathbf{x}}) < h_e$ выполняется в окрестности \bar{X}' некоторой точки $\bar{\mathbf{x}}' \in \bar{X}$.

В ситуации 1 из уравнения (4.108) должно быть найдено уравнение многообразия S'

$$\Phi'(\bar{\mathbf{x}}) = 0,$$

пересечение которого с S дает в X многообразие

$$\left[\frac{\Phi(\mathbf{x})}{\Phi'(\Psi(\mathbf{x}))} \right] = 0$$

(задача стабилизации на пересечении решается по рассмотренной уже общей схеме).

В ситуации 2 при $\bar{\mathbf{x}} \in \bar{X}'$ уравнение (4.108) определяет в R^h многообразие S_u допустимых значений вектора u . Тогда задача 4.4 решается неоднозначно и при необходимости может быть доопределена в целях получения дополнительных свойств проектируемой системы.

4.6. Стабилизация нестационарных моделей

Решение задачи 4.4 пространственного управления для нелинейных объектов сводится к стабилизации линейных нестационарных моделей (4.100), (4.112) или их дискретных аналогов. В общем случае коэффициенты обратных связей могут быть синтезированы точными методами, например, в результате оптимизации интегральных квадратичных (1.29) или суммарных критериев (1.30) [4, 6, 32]. Методы предусматривают интегрирование дифференциальных (или разностных) нестационарных уравнений Риккати с краевыми условиями в конечной точке. Даже в предположении существования устойчивого решения они требуют больших затрат машинного времени. Поэтому во многих прикладных задачах нестационарной стабилизации применяются эвристические алгоритмы управления, основанные на упрощенных решениях оптимальной задачи [57, 71, 101]. Использование таких алгоритмов предполагает проведение предварительной проверки замкнутой системы на устойчивость и качество функционирования.

Постановка задачи стабилизации. Рассмотрим общую задачу стабилизации нестационарной модели

$$\dot{\mathbf{e}} = \mathbf{A}(\mathbf{x})\mathbf{e} + \mathbf{B}(\mathbf{x})\mathbf{u}. \quad (4.113)$$

Здесь $t \geq 0$; $\mathbf{e} \in R^v$; $\mathbf{e}(0) = \mathbf{e}_0$; $\mathbf{u} \in R^h$; $\mathbf{A}(\mathbf{x})$, $\mathbf{B}(\mathbf{x})$ — гладкие функциональные матрицы; \mathbf{x} — вектор переменных параметров, динамика которого подчинена уравнению

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad (4.114)$$

где $\mathbf{x} \in X \subset R^{n-v}$, X — замкнутая и ограниченная область пространства R^{n-v} .

Для данной задачи будем использовать линейные алгоритмы стабилизации

$$\mathbf{u} = -\mathbf{K}(\mathbf{x})\mathbf{e}, \quad (4.115)$$

где $K(x)$ — нестационарная матрица коэффициентов обратной связи по отклонению e .

Тогда поведение замкнутой модели будет соответствовать уравнению

$$\dot{e} = A_c(x) e, \quad (4.116)$$

где $A_c(x) = A - BK$ — матрица замкнутой модели.

Структура рассматриваемой нестационарной системы с регулятором (4.115) приведена на рис. 4.26.

Задачу стабилизации сформулируем как задачу достижения экспоненциальной устойчивости модели (4.116).

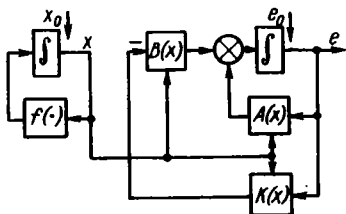


Рис. 4.26. Структура нестационарной системы

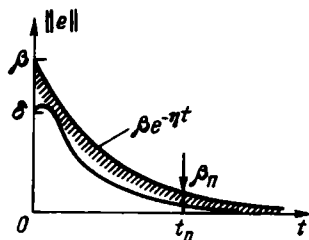


Рис. 4.27. Мажоранта и параметры нормированных переходных процессов

Задача 4.5. Найти нестационарную матрицу $K(x)$ такую, что движение замкнутой модели (4.116) при начальных значениях $\|e_0\| \leq \delta$ и любых $x \in X$, удовлетворяющих уравнению (4.114), экспоненциально устойчиво с оценкой

$$\|e\| \leq \beta \exp(-\eta t), \quad (4.117)$$

где $t \geq 0$; $\eta > 0$; $\beta > 0$.

Решение этой задачи гарантирует сходимость переходного процесса (по норме), и, следовательно, ликвидацию начальной ошибки программного движения или начального отклонения от многообразия в соответствии с условиями задачи 4.4. Более того, значения η и β назначают *мажоранту* ансамбля переходных процессов и задают качество стабилизации.

Под временем переходного процесса будем понимать время t_n , за которое достигаются значения ошибки $\|e\| \leq \beta_n > 0$ (рис. 4.27). Из (4.117) найдем

$$t_n \leq \frac{1}{\eta} \ln \frac{\beta}{\beta_n}.$$

В дальнейшем будем полагать, что x — скалярная величина ($x \in X \subset R^1$). Это позволит избежать при рассмотрении задачи излишних сложностей в обозначениях и преобразованиях. Распространение полученных результатов, на случай модели с несколькими варьируемыми параметрами, затруднений качественного характера не вызывает.

Введем понятие номинального значения параметра $x = x_0$, которое часто отождествляется с начальным значением $x_0 = x(0)$, и определим номинальные матрицы:

$$A_0 = A(x_0); \quad B_0 = B(x_0); \quad K_0 = K(x_0).$$

Поведение номинальной системы (4.116) характеризуется уравнением

$$\dot{e}^0 = A_{c0}e^0, \quad (4.118)$$

где $A_{c0} = A_0 - B_0K_0$.

Возмущенное движение системы (4.116), т. е. движение при $x \neq x_0$, можно оценить с помощью полинома [82, 99]

$$e = e^0 + w_1\Delta x + \frac{1}{2!}w_2\Delta x^2 + \dots, \quad (4.119)$$

где $\Delta x = x - x_0$, $w_i = w_i(t)$, $i = 1, 2, \dots$ — номинальные вектор-функции чувствительности

$$w_1 = \left. \frac{\partial e}{\partial x} \right|_{x=x_0}, \quad w_2 = \left. \frac{\partial w_1}{\partial x} \right|_{x=x_0}, \quad \dots$$

Динамика $w_i(t)$ задается моделями чувствительности, которые нетрудно получить, дифференцируя по x выражение (4.116):

$$\dot{w}_1 = A_{c0}w_1 + G_1e^0; \quad (4.120)$$

$$\dot{w}_2 = A_{c0}w_2 + 2G_1w_1 + G_2e^0; \quad \dots$$

где

$$G_1 = \left. \frac{\partial A_c}{\partial x} \right|_{x=x_0} = \left. \frac{\partial A}{\partial x} \right|_{x=x_0} - \left. \frac{\partial B}{\partial x} \right|_{x=x_0} K_0 - B_0 \left. \frac{\partial K}{\partial x} \right|_{x=x_0}; \quad G_2 = \left. \frac{\partial^2 A_c}{\partial x^2} \right|_{x=x_0}; \quad \dots$$

Использование стационарных моделей (4.118), (4.120) и т. д. совместно с полиномиальным представлением возмущенного движения (4.119) иногда позволяет избежать трудоемкого анализа нестационарной системы (4.116).

Обратим внимание на частные случаи модели (4.113):

$$1) \quad A(x) = \begin{bmatrix} 0 & I \\ -\omega^\top(x) & \end{bmatrix}; \quad B(x) = \begin{bmatrix} 0 \\ \Omega(x) \end{bmatrix},$$

($\omega^\top(x)$ — матрица размерности $h \times v$, $\Omega(x)$ — матрица $h \times h$);

$$2) \quad A(x) = A_0 + B_0\omega^\top(x); \quad B(x) = B_0\Omega(x);$$

3) существует матрица преобразования $T(x)$ ($\|T(x)\| \leq \tau > 0$), с помощью которой (4.113) приводится к модели со структурой матриц, соответствующей частному случаю 2.

Отметим, что частный случай 1 с помощью стационарного преобразования T_0 всегда можно привести к частному случаю 2,

а обратное приведение возможно, по крайней мере, для одноканальной модели (4.113), т. е. когда $h = 1$.

Стационарная стабилизация. В простейших случаях устойчивость нестационарной системы удастся обеспечить с помощью стационарной матрицы обратной связи $K(x) = K_0 = \text{const}$. Воспользовавшись известным подходом (см. [92]), выберем

$$K_0 = R^{-1} B_0^T P_0. \quad (4.121)$$

Здесь P_0 — симметрическая положительно определенная матрица ($P_0 = P_0^T > 0$), являющаяся решением алгебраического уравнения Риккати

$$A_0^T P_0 + P_0 A_0 + P_0 B_0 R^{-1} B_0^T P_0 = -2\eta P_0, \quad (4.122)$$

где η — положительное вещественное число; $R = R^T > 0$.

Уравнение (4.122) интересно прежде всего тем, что его решение может быть найдено без привлечения рекуррентных процедур [92]. Определим обратную матрицу

$$M_0 = P_0^{-1}.$$

Умножая (4.122) слева и справа на M_0 и производя необходимые преобразования, получим

$$M_0 (A_0 + \eta I)^T + (A_0 + \eta I) M_0 = B_0 R^{-1} B_0^T. \quad (4.123)$$

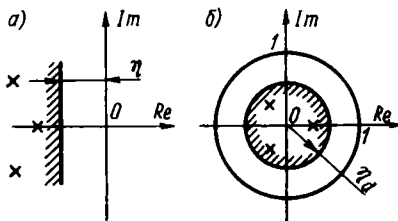


Рис. 4.28. Области расположения корней системы: а — непрерывной; б — дискретной

Итак, найдено линейное уравнение или уравнение Ляпунова, алгебраические методы решения которого известны [29]. Процедура синтеза матрицы обратной связи сводится к линейным операциям: решения уравнения Ляпунова, обращения полученной матрицы M_0 , расчета P_0 по формуле (4.121) (см. рис. 4.30). Эта особенность рассматриваемого метода синтеза становится

очень важной при расчете нестационарных обратных связей.

Оценить свойства полученной замкнутой системы поможет следующее положение.

Теорема 4.8. Если $\eta > \max\{0, \text{Re } \lambda_i [A_0]\}$, $i = \overline{1, n}$, $R > 0$, пара $A_0 + \eta I$, B_0 полностью управляема, то:

а) существует положительно определенное решение P_0 системы (4.122);

б)
$$\text{Re } \lambda_i [A_0 - B_0 K_0] \leq -\eta,$$

где $\lambda_i [\cdot]$ — собственные числа соответствующей матрицы (доказательство теоремы 4.8 приведено в приложении 2).

В теореме 4.8 сформулированы условия, при которых матрица P_0 существует и обратима, а, следовательно, применим указанный выше метод построения обратных связей. Более того, в соответствии с условием б при тех же условиях гарантируется асимптотическая устойчивость номинальной модели (4.118) со степенью устойчивости не менее η (рис. 4.28), а поэтому и экспоненциальная устойчивость с оценкой

$$\|e^0\|_{P_0} \leq \delta \exp(-\eta t) \quad (4.124)$$

для любых $\|e_0\|_{P_0} \leq \delta$ (см. приложение 2); нормы векторов определены как $(e^T P_0 e)^{1/2}$.

Отметим, что метод синтеза обеспечивает для номинальной модели оптимальное решение в смысле квадратичного критерия качества (1.29), где $Q = 2\eta P_0$.

В ходе переходного процесса параметры основной системы (4.116) претерпевают изменения и поэтому ее свойства, вообще говоря, будут отличаться от свойств номинальной модели. Остановимся на некоторых практически важных квазистационарных случаях, для которых возмущенное движение модели на интервале времени $[0, t_n]$ близко к движению стационарной системы (1.118) и, следовательно, справедливо (4.117), где $\beta \cong \delta$. Условие квазистационарности нетрудно получить из (4.118). Пренебрегая членами высоких порядков малости, запишем

$$|\Delta x| \|w_1\|_{P_0} \cong 0 \quad \text{при } t \in [0, t_n], \quad (4.125)$$

где w_1 — номинальная функция чувствительности с моделью (4.120), в которой для рассматриваемого случая

$$G_1 = \frac{\partial A}{\partial x} \Big|_{x=x_0} + \frac{\partial B}{\partial x} \Big|_{x=x_0} K_0.$$

На указанном временном интервале

$$|\Delta x| \leq f_m t_n,$$

где $f_m = \max |f(x)|$.

Принимая во внимание, что $t_n \leq \frac{1}{\eta} \ln(\delta/\beta_n)$ и δ, β_n — фиксированы, перепишем условие (4.125) в виде

$$f_m \|w_1\|_{P_0} / \eta \cong 0. \quad (4.126)$$

Таким образом, к квазистационарным системам можно отнести системы с малыми траекторной чувствительностью $w_1 \cong 0$ и скоростью изменения параметров $\dot{x} = f(x) \cong 0$ или значительной степенью устойчивости номинальной модели η , определяющей быстрое протекание переходного процесса. Как показывает более детальное рассмотрение условия квазистационарности, его выполнение для частных случаев 1, 2 обеспечивается при увеличении числа η как параметра процедуры синтеза, входящего в уравнение (4.123). В общем случае изменение коэффициентов

обратной связи K_0 , связанное с увеличением η и, следовательно, параметров модели чувствительности, приводит к увеличению текущих значений функции w_1 и нарушению условия (4.126).

Нарушение условий квазистационарности еще не исключает возможности эффективного использования стационарных обратных связей. Качественно следует ожидать, что в мажоранте неравенства (4.117) получим $\beta \neq \delta$ и η , отличающееся от заданного значения. Необходимые количественные оценки возмущенного движения можно найти, анализируя методами моделей сравнения [14] динамические свойства вектора e как выходной переменной расширенной стационарной системы (4.118)—(4.120). Для систем с достаточно широким диапазоном изменения параметров матриц $A(x)$ и $B(x)$ устойчивость, достигнутая на интервале $[0, t_n]$, может не сохраняться в дальнейшем. Отсюда вытекает необходимость периодической подстройки коэффициентов обратной связи (см. рис. 4.30), что сближает класс стационарных регуляторов с нестационарными регуляторами, рассматриваемыми далее.

Нестационарная стабилизация. В частных случаях 1—3 задача стабилизации системы (4.113) может быть сведена к стационарной задаче. В частных случаях 1, 2 при условии, что $\det \Omega(x) \neq 0$, для любых $x \in X$ выбираем

$$u = -\Omega^{-1}(x) \omega^T(x) e + \Omega^{-1}(x) u_1.$$

Система (4.113) принимает вид

$$\dot{e} = A_0 e + B_0 u_1$$

и становится возможным использование стационарных обратных связей

$$u_1 = -K_0 e,$$

где K_0 определяется выражениями (4.121), (4.122); параметр η задает степень устойчивости, полученной в результате замкнутой системы.

Таким образом, нестационарный алгоритм управления приобретает вид

$$u = -\Omega^{-1}(x) [\omega^T(x) + K_0] e. \quad (4.127)$$

Для частного случая 3 осуществим преобразование вектора состояния e системы (4.113) к

$$e_t = T(x) e, \quad (4.128)$$

где $\det T(x) \neq 0$; $\|T\| \leq \tau_t > 0$.

После этого система приводится к частному случаю 2, и управление можно выбрать в форме, аналогичной (4.127), т. е.

$$u = -\Omega^{-1}(x) [\omega^T(x) + K_t] e,$$

где матрица $K_t = \text{const}$ определяется из выражения $K_t = R^{-1} B_0^T P_t$; P_t — решение уравнения (4.122) для преобразованной системы.

По теореме 4.8 для преобразованного вектора состояния e_t справедлива экспоненциальная оценка

$$\|e_t\|_{P_t} \leq \|e_{t_0}\|_{P_t} \exp(-\eta t),$$

где $e_{t_0} = T(x_0) e_0$.

Для того чтобы оценить поведение вектора состояния e основной системы, осуществим обратное преобразование

$$\|e_t\|_{P_t}^2 = e_t^T P_t e_t = e^T P(x) e = \|e\|_{P(x)}^2.$$

Здесь $P(x) = T^T(x) P_t T(x)$. Следовательно,

$$\|e\|_{P(x)} \leq \|e_0\|_{P_0} \exp(-\eta t), \quad (4.129)$$

где $P_0 = T^T(x_0) P_t T(x_0)$.

Неравенство (4.129) задает мажоранту *нестационарной нормы* вектора e и поэтому не позволяет сделать выводов об устойчивости и качестве движения. Вопросы использования нестационарных норм рассмотрим ниже. Здесь осуществим переход к стационарной норме $\|e\|_{P_0}$, для которого необходимо найти функцию $\beta_1(x)$ такую, что

$$\|e\|_{P_0} / \beta_1(x) \leq \|e\|_{P(x)}.$$

Тогда, усиливая (4.129), получим

$$\|e\|_{P_0} \leq \beta_1(x) \|e_0\|_{P_0} \exp(-\eta t). \quad (4.130)$$

Точность полученной оценки зависит от выбора $\beta_1(x)$. Наилучшая для каждого $x \in X$ функция $\beta_1(x)$ находится как максимум отношения Релея [42]:

$$\beta_1(x) = \max_{\|e\|_{P(x)}=c} \frac{\|e\|_{P_0}}{\|e\|_{P(x)}} = \max_t \lambda_t [P_0 M(x)],$$

причем, так как $P_0 M^{-1}(x_0) = I$, то $\beta_1(x_0) = 1$.

Если функция $\beta_1(x)$ ограничена сверху, т. е. найдется

$$\beta_{10} = \sup_{x \in X} \beta_1(x)$$

и значения e_0 удовлетворяют условию $\|e_0\|_{P_0} \leq \delta$, то замкнутая система будет экспоненциально устойчивой с оценкой (4.117), где $\beta = \delta \beta_{10}$.

Увеличение параметра η позволяет ускорить переходный процесс преобразованной системы и уменьшить время t_n , поэтому

$$\lim_{\eta \rightarrow \infty} T(x) = T(x_0) \quad \text{при } t \in \{0, t_n\}.$$

и, следовательно, $P(x) \rightarrow P_0$, $\beta_1(x) \rightarrow 1$.

Таким образом, выбором значения η свойства систем, относящихся к частному случаю 3, могут быть максимально приближены к свойствам стационарных систем, или может быть обеспечено существование вещественной функции $\beta_1(x)$ за счет сближения параметров $P(x)$ и P_0 .

Пример 4.10. Рассмотрим нестационарную систему

$$\dot{e} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix} e + \begin{bmatrix} x \\ \exp x - 1 \end{bmatrix} u, \quad (4.131)$$

где e, x, u — функции безразмерного аргумента, причем $\dot{x} = -1, x_0 = 1, x \in (0, 1]$.
С помощью нестационарного преобразования

$$T(x) = \begin{bmatrix} 1 & -x/(\exp x - 1) \\ 0 & (x + \exp x - 1)/(\exp x - 1)^2 \end{bmatrix}$$

система (4.131) приводится к виду

$$\dot{e}_t = \begin{bmatrix} 0 & 1 \\ 0 & \omega(x) \end{bmatrix} e_t + \begin{bmatrix} 0 \\ \Omega(x) \end{bmatrix} u,$$

$$\omega(x) = \frac{2}{\exp x - 1} + \frac{x}{x - (\exp x - 1)}; \quad \Omega(x) = 1 - \frac{x}{\exp x - 1},$$

т. е. может быть отнесена к частному случаю 3.

В соответствии с (4.127) сформируем управление

$$u = -\frac{1}{\Omega(x)} [k_1 | \omega(x) + k_2] T(x) e.$$

Решая уравнение (4.122) для $A_0 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, b_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, R = 1$, найдем

$$P_t = 4\eta \begin{bmatrix} 2\eta^2 & \eta \\ \eta & 1 \end{bmatrix},$$

и, следовательно,

$$K_0 = [k_1 k_2] = [0 \quad 1] P_t = [4\eta^2 \quad 4\eta];$$

задавая $\eta = 5$, получим $k_1 = 100, k_2 = 20$.

Для оценки переходного процесса определим

$$P(x) = T^T(x) P_t T(x) =$$

$$= 4\eta \left[\begin{array}{c|c} 2\eta^2 & \eta(1-2\eta)\rho_1 - \eta\rho_2 \\ \hline \eta(1-2\eta)\rho_1 - \eta\rho_2 & (1-2\eta+2\eta^2)\rho_1^2 - 2(\eta-1)\rho_1\rho_2 + \rho_2^2 \end{array} \right],$$

где $\rho_1 = x/(\exp x - 1); \rho_2 = -(\exp x - 1 - x \exp x)/(\exp x - 1)^2$.

Для $\eta = 5$:

$$P(x) = 20 \left[\begin{array}{c|c} 50 & -45\rho_1 - 5\rho_2 \\ \hline -45\rho_1 - 5\rho_2 & 41\rho_1^2 + 8\rho_1\rho_2 + \rho_2^2 \end{array} \right];$$

$$P_0 = P(x_0) = 20 \begin{bmatrix} 50,0 & -27,9 \\ -27,9 & 15,6 \end{bmatrix}.$$

Найдем: $\beta_{10}^2 = \beta_1(0) = 1/\min_{i=1,2} \lambda_i [P(0) P_0^{-1}] = 56,2$ и $\beta_{10} = 7,5$. Полагая

$e \|P_0 \leq 0,1$, в соответствии с (4.117), получим

$$\|e\|_{P_0} \leq 0,75 \exp(-5t).$$

Мажоранта переходных процессов приведена на рис. 4.29, кривая 2. Более точно качество процессов можно оценить с помощью норм вращения (см. пример 4.11).

В общем случае будем использовать алгоритм управления (4.115), где матрица обратной связи выбирается как

$$\mathbf{K}(x) = \mathbf{R}^{-1} \mathbf{B}^T(x) \mathbf{P}(x), \quad (4.132)$$

а матрица $\mathbf{P}(x)$ ($\mathbf{P}(x) = \mathbf{P}^T(x) > 0$) — решение алгебраического уравнения Риккати

$$\begin{aligned} & \mathbf{A}^T(x) \mathbf{P}(x) + \mathbf{P}(x) \mathbf{A}(x) - \\ & - \mathbf{P}(x) \mathbf{B}(x) \mathbf{R}^{-1}(x) \mathbf{B}^T(x) \mathbf{P}(x) = -2\eta \mathbf{P}(x) \end{aligned} \quad (4.133)$$

в котором $\mathbf{P}(x_0) = \mathbf{P}_0$; $\eta > 0$; $\mathbf{R}(x) = \mathbf{R}^T(x) > 0$.

Определим обратную матрицу

$$\mathbf{M}(x) = \mathbf{P}^{-1}(x),$$

преобразуем (4.133) к уравнению Ляпунова

$$\mathbf{M}(x) [\mathbf{A}(x) + \eta \mathbf{I}]^T + [\mathbf{A}(x) + \eta \mathbf{I}] \mathbf{M}(x) = -\mathbf{B}(x) \mathbf{R}^{-1}(x) \mathbf{B}^T(x). \quad (4.134)$$

Алгоритм расчета матрицы обратной связи (рис. 4.30) для каждого текущего значения x предусматривает решение уравнения (4.134), обращение матрицы $\mathbf{M}(x)$ и вычисление $\mathbf{K}(x)$ по формуле (4.132). Процедура входит в состав прикладного программного обеспечения цифровой САУ или используется на этапе синтеза для нахождения множества значений $\mathbf{K}(x)$.

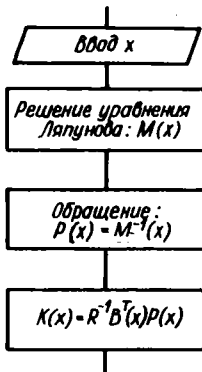


Рис. 4.30. Алгоритм расчета матрицы обратной связи

Рассмотрим динамические свойства замкнутой системы. Следующий результат является обобщением теоремы 4.8 для произвольного фиксированного значения параметра $x \in X$.

Теорема 4.9. Если для заданного значения $x \in X$ имеет место $\eta > \max \{0, \operatorname{Re} \lambda_i [\mathbf{A}(x)]\}$, $i = \overline{1, v}$; $\mathbf{R}(x) > 0$ и пара $\mathbf{A}(x) + \eta \mathbf{I}, \mathbf{B}(x)$ полностью управляема, то: а) существует положительно определенное решение $\mathbf{P}(x)$ уравнения (4.133);

$$б) \operatorname{Re} \lambda_i [\mathbf{A}(x) + \mathbf{B}(x) \mathbf{K}(x)] \leq -\eta, \quad i = \overline{1, v}.$$

Пункт а) теоремы 4.9 устанавливает условия существования и корректности предложенной процедуры синтеза матрицы обратных связей $\mathbf{K}(x)$. Для квазистационарных систем, удовлетворяющих условию (4.126), пункт б) гарантирует асимптотическую устойчивость нулевого решения (4.116) со степенью η , причем

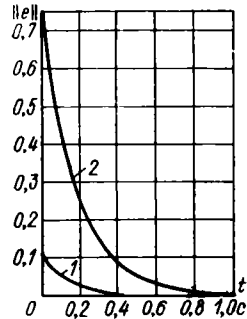


Рис. 4.29. Мажоранты переходных процессов

квазистационарность для частных случаев 1, 2 достигается путем выбора достаточно большого значения параметра η .

В общем случае отрицательность вещественных частей корней нестационарной системы не является критерием ее устойчивости, и требуется дополнительное исследование свойств модели (4.116). В приложении 2 с использованием метода функций Ляпунова [14.92] получены оценки переходных процессов. В данном случае ограничимся обсуждением основных результатов.

Рассмотрим матричную функцию чувствительности $\partial P/\partial x$. Для нахождения $\partial P/\partial x$ воспользуемся уравнением Ляпунова

$$\begin{aligned} & [A_c(x) + \eta I]^T \frac{\partial P}{\partial x} + \frac{\partial P}{\partial x} [A_c(x) + \eta I] = \\ & = P(x) \left[\frac{\partial B}{\partial x} R^{-1}(x) B^T(x) - B(x) R^{-1}(x) \frac{\partial B^T}{\partial x} \right] P(x) - \\ & \quad - \frac{\partial A^T}{\partial x} P(x) - P(x) \frac{\partial A}{\partial x}, \end{aligned} \quad (4.135)$$

полученным дифференцированием по x уравнения Риккати (4.133).

Определим также функцию

$$\beta_2(x) = \exp \left\{ \frac{1}{2} \int_{x_0}^x \max_i \lambda_i \left[\frac{\partial P}{\partial x} M(x) \right] dx \right\} \quad (4.136)$$

и сформулируем следующий результат, доказанный в приложении 2.

Т е о р е м а 4.10. Если для любых $x \in X$ выполняются условия теоремы 4.9, то для переходных процессов системы (4.116) с начальными значениями $\|e_0\|_{P_0} \leq \delta$ справедлива оценка

$$\|e\|_{P(x)} \leq \beta(x) \exp(-\eta t), \quad (4.137)$$

где $\beta(x) = \delta \beta_2(x)$.

Теорема устанавливает мажоранту $\beta(x) \exp(-\eta t)$, ограничивающую переходные процессы по нестационарной норме $\|e\|_{P(x)}$. Так как в состав мажоранты входит функция $\beta(x)$, то ее возможное возрастание будет препятствовать сходимости $\|e\|_{P(x)}$. Поэтому необходимо оговорить существование верхней границы β_{20} функции $\beta_2(x)$. Тогда рассматриваемая норма отклонения будет сходиться к нулю с оценкой

$$\|e\|_{P(x)} \leq \delta \beta_{20} \exp(-\eta t). \quad (4.138)$$

Условие (4.138) ввиду нестационарности используемой нормы еще не обеспечивает приближения состояния системы (4.116) к положению равновесия $e = 0$. Рассмотрим поведение поверхности постоянного уровня нормы (оценочного гиперэллипсоида)

$$\|e\|_{P(x)} = c, \quad (4.139)$$

где $c = \delta \beta_{20} = \text{const}$.

С течением времени и с изменением x поверхность (4.139) деформируется. Если ее «растяжение» происходит быстрее экспоненциального затухания мажоранты $\exp(-\eta t)$, то переходные процессы могут удаляться от положения равновесия (рис. 4.31).

Выше было показано, что оценка по нестационарной норме сводится к оценке нормы $\|e\|_{P(x)}$ (4.130). Геометрически такая процедура соответствует вписыванию нестационарного гиперэллипсоида (4.139) в начальный гиперэллипсоид $\|e\|_{P_0} = c$ (рис. 4.32). Однако для систем с переменными параметрами этот метод оказывается грубым. При изменении x поверхность (4.139) претерпевает два вида деформаций: 1) изменяются параметры эллипсоида (длины осей); 2) эллипсоид разворачивается относительно начала координат, что связано соответственно с изменением собственных чисел и собственных векторов матрицы $P(x)$. При отсутствии поворота параметр $\beta_1(x)$ характеризует отношение длин осей начального и текущего гиперэллипсоидов и достаточно полно отражает характер деформаций. Поворот оценочного гиперэллипсоида даже при сохранении его размеров приводит к необходимости увеличения $\beta_1(x)$ (рис. 4.32) и ухудшению оценки по стационарной норме (4.130).

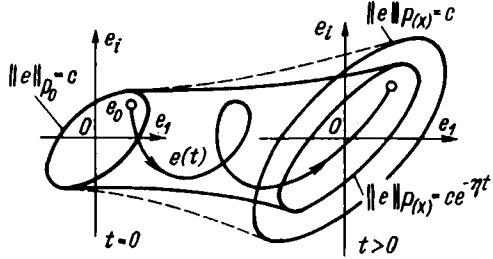


Рис. 4.31. Деформация поверхности постоянного уровня (расходящийся переходный процесс)

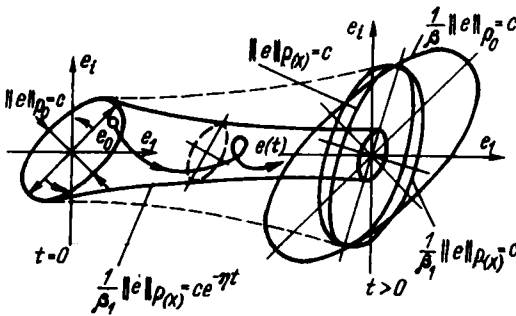


Рис. 4.32. Поверхности постоянного уровня (сходящийся переходный процесс)

является естественным свойством систем с переменными параметрами. Если вращение не сопровождается растяжением или сжатием, то оно сохраняет длины (единичные нормы) векторов и, при условии экспоненциальной сходимости нормы $\|e\|_{P(x)}$, не препятствует затуханию переходных процессов. Этими факторами и обусловлена необходимость использования нестационарных норм, нечувствительных к поворотам векторов, но в то же время учитывающих изменения расстояний.

Введем в рассмотрение *нестационарную норму* (вращения)

$$\|e\|_{\tilde{P}(x)}^2 = e^T \tilde{P}(x) e, \quad (4.140)$$

где $\tilde{P}(x) = T^T(x) P_0 T(x)$; $T(x)$ — ортогональная матрица ($T^T T = T T^T = I$); $T(x_0) = I$.

Такой выбор $\tilde{P}(x)$ позволяет сохранить значение нормы при вращении вектора e , вызванном изменением параметров системы. Поверхность постоянного уровня $\|e\|_{\tilde{P}(x)} = c$ получается поворотом осей начального эллипсоида $\|e\|_{P_0} = c$ (рис. 4.32), а условие устойчивости (4.117) интерпретируется как требование экспоненциального уменьшения параметров вращающегося эллипсоида. Для перехода к оценке по нормам вращения необходимо найти функцию $\tilde{\beta}_1(x)$ такую, что

$$\|e\|_{\tilde{P}(x)} / \tilde{\beta}_1(x) \leq \|e\|_{P(x)};$$

тогда, усиливая (4.137), получим

$$\|e\|_{\tilde{P}(x)} \leq \tilde{\beta}_1(x) \beta_2(x) \|e_0\|_{P_0} \exp(-\eta t). \quad (4.141)$$

Правая часть полученного неравенства является мажорантой переходных процессов по нормам вращения, сходимость которой к нулю обеспечивает устойчивость нестационарной системы. Если для всех $x \in X$ найдется верхняя граница $\beta = \sup_{x \in X} [\tilde{\beta}_1(x) \beta_2(x)]$, то система (4.116) будет экспоненциально

устойчивой по норме вращения с оценкой (4.117), где $\beta = \delta \tilde{\beta}$. Тем самым решается общая задача стабилизации 4.5. При большом диапазоне изменения величины $\tilde{\beta}_1(x) \beta_2(x)$ такая оценка переходных процессов может оказаться очень грубой. Поэтому из условия

$$\tilde{\beta}_1(x) \beta_2(x) \exp(-\eta t) \leq \tilde{\beta}' \exp(-\eta' t)$$

следует попытаться найти значения $\tilde{\beta}'$ и $\eta' < \eta$, задающие более точную мажоранту и лучшую оценку вида

$$\|e\|_{\tilde{P}(x)} \leq \beta' \exp(-\eta' t),$$

где $\beta' = \delta \tilde{\beta}'$.

Отметим, что в любом случае качество переходных процессов, определяемое значениями β и η (β' и η'), регулируется подбором параметра η , задающего скорость затухания оценочной мажоранты.

Матрицы $T(x)$ и функция $\tilde{\beta}_1(x)$ определяются неоднозначно, и от их выбора в значительной мере зависит точность оценки (4.141).

Введем в рассмотрение множество Ξ ортогональных матриц $T(x)$ таких, что $T(x_0) = I$. Тогда наилучшая оценка (4.141) для любого $x \in X$ может быть получена при выборе $\tilde{\beta}_1(x)$ как

$$\tilde{\beta}_1(x) = \min_{\Xi} \max_{\|e\|_{P(x)}=c} \frac{\|e\|_{\tilde{P}(x)}}{\|e\|_{P(x)}}. \quad (4.142)$$

и матрицы $T(x) \in \mathbb{E}$, доставляющей искомое минимальное решение.

Возможность выбора матрицы вращения $T(x)$ позволяет значительно уменьшить $\beta_1(x)$ по сравнению со значениями $\beta_1(x)$, полученными при использовании стационарных норм и $T(x) = I$. Решение задачи дает следующее положение, доказательство которого приведено в приложении 2.

Т е о р е м а 4.11. Пусть $T_1(x)$ — ортогональная матрица преобразования

$$P(x) = T_1^T(x) \Lambda(x) T_1(x), \quad (4.143)$$

где $\Lambda(x) = \text{diag} \{ \lambda_i [P(x)] \}$; $T_{10} = T_1(x_0)$, $\Lambda_0 = \Lambda(x_0)$, причем для любых $x \in X$ собственные числа матриц Λ и Λ_0 упорядочены по убыванию значений с увеличением $i = \overline{1, n}$.

Тогда

$$\tilde{\beta}_1(x) = \left\{ \max_i \frac{\lambda_i [P_0]}{\lambda_i [P(x)]} \right\}^{1/2} \quad (4.144)$$

при $T(x) = T_{10}^T T_1(x)$.

Выбор матрицы $T(x)$ осуществляется из условия совпадения собственных векторов матриц $\tilde{P}(x)$ и $P(x)$, т. е. совмещения осей соответствующих эллипсоидов. Тогда функция $\tilde{\beta}_1^2$ становится равной отношению собственных чисел \tilde{P}_0 (или P_0) и $P(x)$ и не зависит от поворота эллипсоида (4.139) (см. рис. 4.32).

Таким образом, процедура оценки устойчивости и качества переходного процесса нестационарной системы в общем случае требует нахождения функций $\tilde{\beta}_1(x)$ (4.144) и $\beta_2(x)$ (4.136), а затем верхней границы $\tilde{\beta}$ их произведения.

П р и м е р 4.11. Продолжая рассмотрение системы (4.131), проанализируем деформацию линий постоянного уровня (эллипса)

$$\|e\|_{P(x)} = 0,1. \quad (4.145)$$

Эллипсы для значений $x = x_0 = 1$ и $x = x(1) = 0$ приведены на рис. 4.33. Отметим, что с течением времени и измерением x оценочный эллипс разворачивается, а длины его осей уменьшаются. Штриховой линией на рис. 4.33 показан эллипс стационарной нормы

$$\|e\|_{P_0} / \beta_{10} = 0,1. \quad (4.146)$$

Величина $\beta_{10} = 7,5$ (см. пример 4.10) найдена из условия

$$\|e\|_{P_0} / \beta_{10} \leq \|e\|_{P(x)},$$

т. е. соответствует значению β_{10} , при котором в стационарный эллипс (4.146) вписываются все частные реализации эллипса (4.145). Из рисунка видно, что необходимость выбора большего значения обусловлена исключительно вращением линии (4.145).

Рассмотрим оценку по нормам вращения. Анализируя поведение функции $\tilde{\beta}_1(x) = \max_{i=1,2} \{ \lambda_i [P_0] / \lambda_i [P(x)] \}$, отметим, что она изменяется от 1 (при $t = 0$, $x = 1$) до 0,69 (при $t = 1$, $x = 0$), а ее верхняя граница $\tilde{\beta}_{10} = \sup_{(0,1]} \tilde{\beta}_1(x) = 1$. Изменение $\tilde{\beta}_1(x)$ соответствует уменьшению осей оценочного эллипса на рис. 4.33.

Таким образом, переходный процесс нестационарной замкнутой системы по норме вращения экспоненциально устойчив с оценкой

$$\|e\| \leq 0,1 \exp(-5t)$$

для $\|e_0\| \leq 0,1$ (см. рис. 4.29, кривая 1).

Пример 4.12. Рассмотрим общий способ стабилизации системы (4.131). Назначая $\eta = 5$ и $R(x) = x^2$, решим уравнение (4.134):

$$M(x) = \begin{bmatrix} 0,1 & 0,11/p_1(x) \\ 0,11/p_1(x) & 0,13/p_1^2(x) \end{bmatrix}$$

и, следовательно,

$$P(x) = M^{-1}(x) = \begin{bmatrix} 810 & -720p_1(x) \\ -720p_1(x) & 648p_1^2(x) \end{bmatrix}.$$

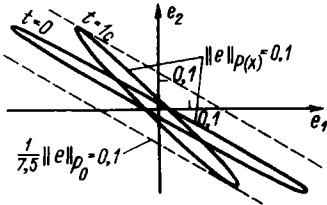


Рис. 4.33. Линии постоянного уровня

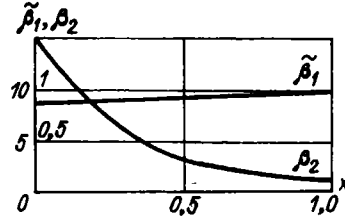


Рис. 4.34. Вариации параметров мажоранты

По формуле (4.132) найдем

$$K(x) = \frac{1}{x^2} [x | \exp x - 1] P(x) = [90/x | -72/(\exp x - 1)].$$

Отметим, что матрица замкнутой системы

$$A_c(x) = \begin{bmatrix} -90 & 72p_1(x) \\ -90/p_1(x) & 71 \end{bmatrix}$$

имеет фиксированные собственные числа

$$\lambda_i [A_c(x)] = \{-10, -9\} < -\eta$$

и нестационарные собственные векторы

$$\begin{bmatrix} 1 \\ 1,11/p_1(x) \end{bmatrix}, \quad \begin{bmatrix} 1 \\ 1,13/p_1(x) \end{bmatrix}.$$

Из выражения (4.136) найдем $\beta_2(x)$. Предварительно, решая уравнение Ляпунова (4.135), получим

$$\frac{\partial P}{\partial x} = \begin{bmatrix} 0 & 810p_2(x) \\ 810p_2(x) & -1296p_2(x)/p_1(x) \end{bmatrix};$$

тогда

$$\max_{i=1,2} \lambda_i \left[-\frac{\partial P}{\partial x} M(x) \right] = -10p_2(x)/p_1(x)$$

и, следовательно,

$$\beta_2(x) = \exp \left\{ -\frac{1}{2} \int_1^x 10 \frac{p_2(x)}{p_1(x)} dx \right\} = \left(\frac{p_1(0)}{p_1(x)} \right)^5.$$

Зависимость $\beta_2(x)$ приведена на рис. 4.34. Нестационарная оценка замкнутой системы принимает вид

$$\|e\|_{P(x)} \leq \beta_2(x) \exp(-5t).$$

Проанализируем поведение оценочного эллипса. Его реализации при $x(0) = 1$ и $x(1) = 0$ представлены на рис. 4.35. Вращение эллипса приводит к загромождению оценки по стационарной норме $\|e\|_{P_s}$. Поэтому воспользуемся нормой вращения. По формуле (4.144) определим функцию $\beta_1(x)$ (см. рис. 4.34). Тогда мажоранта процессов по норме вращения принимает вид $0,1\beta_1(x)\beta_2(x)\exp(-5t)$ (рис. 4.36, кривая 3). Переходя к оценке вида (4.117), найдем $\bar{\beta} = \sup_{x \in (0,1]} \beta_1\beta_2 = 12,6$

и, следовательно, $\|e\|_{\tilde{P}(x)} \leq 1,26 \exp(-5t)$. Оценка подтверждает экспоненциальную устойчивость, но, как видно из рис. 4.36 (кривая 1), очень груба.

Нетрудно подобрать стационарную мажоранту с меньшим затуханием η' . Из условия

$$\bar{\beta}_1(x)\beta_2(x)\exp(-5t) \leq \beta' \exp(-\eta't)$$

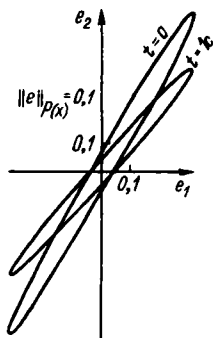


Рис. 4.35. Линии постоянного уровня (общий способ стабилизации)

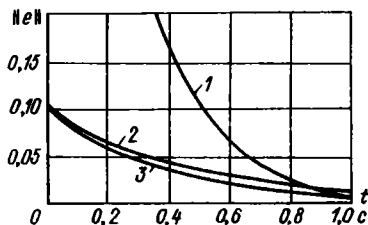


Рис. 4.36. Мажоранты переходных процессов (общий способ стабилизации)

(см. рис. 4.36, кривая 2) найдем $\beta' = 1$, $\eta' = 2$. Следовательно, справедлива оценка

$$\|e\|_{\tilde{P}(x)} \leq 0,1 \exp(-2t).$$

Алгоритмы стабилизации дискретных систем. Специфика микропроцессорного управления наиболее полно учитывается при использовании рекуррентных моделей вида

$$e_{m+1} = A_d(x_m) e_m + B_d(x_m) u_m; \quad m = 0, 1, 2, \dots, \quad (4.147)$$

полученных дискретизацией непрерывных уравнений ошибки (4.113) (см. п. 4.5).

Рассмотренные здесь методы синтеза обратных связей практически без изменений могут быть применены к дискретной задаче стабилизации.

В общем случае воспользуемся алгоритмом (4.115), где $x = x_m$, а матрица обратной связи выбирается как [34, 37]

$$K(x_m) = [R + B_d^T(x_m) P(x_m) B_d(x_m)]^{-1} B_d^T(x_m) P(x_m) A_d(x_m). \quad (4.148)$$

Положительно определенная симметрическая матрица $\mathbf{P}(x_m)$ ищется как решение дискретного аналога алгебраического уравнения Риккати

$$\begin{aligned} \mathbf{A}_d^\top(x_m) \mathbf{P}(x_m) \mathbf{A}_d(x_m) - \mathbf{P}(x_m) - \mathbf{A}_d^\top(x_m) \mathbf{P}(x_m) \mathbf{B}_d(x_m) \times \\ \times [\mathbf{R} + \mathbf{B}_d^\top(x_m) \mathbf{P}(x_m) \mathbf{B}_d(x_m)]^{-1} \mathbf{B}_d^\top(x_m) \mathbf{P}(x_m) \mathbf{A}_d(x_m) = \\ = -(1 - \eta_d^2) \mathbf{P}(x_m), \end{aligned} \quad (4.149)$$

где $0 < \eta_d < 1$.

Определим матрицу $\mathbf{M}(x_m) = \mathbf{P}^{-1}(x_m)$, после чего преобразуем уравнение (4.149) к виду:

$$\frac{\mathbf{A}_d(x_m)}{\eta_d} \mathbf{M}(x_m) \frac{\mathbf{A}_d^\top(x_m)}{\eta_d} - \mathbf{M}(x_m) = \mathbf{B}_d(x_m) \mathbf{R} \mathbf{B}_d^\top(x_m). \quad (4.150)$$

Линейное уравнение (4.150) называется дискретным уравнением Ляпунова, и методы его решения хорошо известны [29].

Таким образом, процедура синтеза дискретного регулятора аналогична процедуре, представленной на рис. 4.30, и на каждом шаге включает решение уравнения (4.150), обращение матрицы $\mathbf{P}(x_m)$ и расчет матрицы обратных связей $\mathbf{K}(x_m)$, по формуле (4.148).

Метод синтеза при некоторых дополнительных ограничениях на параметры модели (4.147), гарантирует для любых $x \in X$ расположение собственных чисел матрицы замкнутой системы $\mathbf{A}_{cd}(x_m) = \mathbf{A}_d - \mathbf{B}_d \mathbf{K}$ в круге радиуса η_d (см. рис. 4.28). Выбор η_d определяет степень сходимости дискретной системы. Для квазистационарных моделей обеспечивается экспоненциальная устойчивость переходного процесса с оценкой

$$\|\mathbf{e}_m\| \leq \delta \eta_d^m$$

для любых начальных значений $\|\mathbf{e}_0\| \leq \delta$.

С использованием дискретных функций Ляпунова [14] трудно получить свойства нестационарных моделей общего вида, и в частности условия экспоненциальной устойчивости.

Процедуры расчета коэффициентов обратной связи непрерывных и дискретных нестационарных систем (см. рис. 4.30) в большинстве практических случаев не позволяют непосредственно получить аналитический результат, т. е. функциональную зависимость $\mathbf{K} = \mathbf{K}(x)$. Поэтому выше предполагалось, что основные операции процедуры производятся управляющей микроЭВМ в реальном масштабе времени. Реализация операций линейной алгебры требует использования специальных программных или аппаратных средств и в любом случае связана с существенными затратами машинного времени и увеличением интервала квантования T .

В связи с этим представляет интерес подход, который предусматривает получение и использование в прикладных программах САУ аппроксимированных выражений для расчета коэффициентов

обратной связи. Процедура синтеза аппроксимированных алгоритмов (см. п. 4.7) включает расчет многомерного массива $МК$ коэффициентов обратных связей (для дискретного множества значений вектора $x \in X$) и аппроксимацию полученного массива полиномиальным выражением. Для одноканального случая, т. е. $h = 1$, последнее имеет вид:

$$K(x) \cong N^T z(x), \quad (4.151)$$

где $z(x)$ — вектор базисных функций; N^T — матрица коэффициентов аппроксимации. Этап аппроксимации проводится с привлечением методов, рассмотренных в п. 4.4.

Расчет управляющего воздействия, выполняемый управляющей ЭВМ на каждом такте ее работы, сводится к вычислению базисных функций (обычно степенных), определению матрицы $K(x)$ по формуле (4.151) и вектора u по формуле (4.115), т. е. к последовательности простейших алгебраических операций.

4.7. Синтез системы управления

Оптимальное функционирование САУ поддерживается локальными алгоритмами управления и алгоритмом переключения структуры, осуществляющим их последовательную коммутацию. Переключение локальных алгоритмов производится при изменении вектора состояния объекта управления, а точнее — по мере прохождения фазовой траекторией областей пространства R^n . При пересечении границ окрестностей оптимальных многообразий и переходе из одной области знакопостоянства в другую требуется изменение знаков оптимизирующих компонент U (или \bar{U}), переключение воздействий V и функциональных коэффициентов обратной связи K .

После того как получена необходимая аналитическая информация, задача синтеза САУ сводится к построению наиболее рациональной, с точки зрения качества работы и сложности цифровой реализации, структуры системы и ее элементов (алгоритмов). В состав функциональных элементов многорежимной цифровой системы (см. рис. 1.10) входит планирующая программа, реализующая алгоритм переключения структуры, и подпрограммы локальных алгоритмов управления. Сложность разрабатываемого прикладного программного обеспечения, а также возможности его быстрого приспособления для решения различных задач управления, в значительной степени определяются степенью унификации алгоритмических модулей. Рациональная организация цифровой САУ предполагает использование минимального набора локальных алгоритмов и наименьшего числа коммутируемых элементов. Так как алгоритмическое обеспечение, полученное аналитическими методами этим требованиям обычно не удовлетворяет, то на завершающем этапе синтеза САУ возникает задача приведения локальных алгоритмов к унифицированной форме.

Алгоритм переключения структуры. Различие требований, предъявляемых к САУ на разных стадиях ее функционирования, определяет необходимость выделения режимов работы, коммутируемых алгоритмом переключения структуры. Определим режим r как дискретную переменную, принимающую квантованные значения из области Ω .

В традиционных системах программного управления [34] переключение режимов часто осуществляется по жестко заданным или пересчитываемым временным программам, т. е. по разомкнутому алгоритму, не учитывающему текущего состояния САУ,

$$r = \rho_{\Gamma}^{\prime}(t),$$

где ρ_{Γ}^{\prime} — оператор квантования.

Не исключая возможности использования временного подхода в простых случаях, далее остановимся на замкнутых реализациях переключения структуры, в которых моменты коммутации режимов определяются текущим состоянием объекта управления, т. е.

$$r = \rho_{\Gamma}(x),$$

где оператор ρ_{Γ} устанавливает соответствие непрерывных значений $x \in R^n$ и дискретных значений $r \in \Omega$. Для уточнения множества режимов Ω примем во внимание, что оптимальному поведению ОУ отвечает определенная последовательность прохождения областей знакопостоянства (подмножеств пространства R^n размерности n) и многообразий оптимального движения размерности от $n - 1$ (на гиперповерхностях) до 0 (в конечной точке движения x_f). Поэтому определим Ω следующим образом:

$$\Omega = \{r_1^n, r_2^n, \dots, r_1^{n-1}, r_2^{n-1}, \dots, r_s^i, \dots, r_1^0\},$$

где r_s^i — символы; $i = n - \nu$ — размерность текущего многообразия; s — индекс, присваиваемый данному многообразию в ряду множеств равной размерности.

Тогда при $r = r_s^i$ имеет место режим движения по s -му многообразию размерности i .

Найдем условия, при которых вектор состояния принадлежит характерной области пространства R^n , и установим связь этих условий с режимами работы САУ. Для формирования условий переключения используется процедура *упорядоченной дискретизации* R^n . Она предусматривает последовательное разбиение пространства на области знакопостоянства и окрестности поверхностей оптимального движения, затем выделение в каждой окрестности собственных областей знакопостоянства и многообразий оптимального движения и т. д. вплоть до δ -окрестности конечной точки движения.

Дискретизация пространства. Исходными данными для этой процедуры служит аналитическое описание следующих множеств:

1) гиперповерхностей переключения пространства R^n (многообразий размерности $n - 1$)

$$\Theta_q^n(\mathbf{x}) = 0; \quad q = \overline{1, \pi_n}, \quad (4.152)$$

и гиперповерхностей оптимального движения

$$\Phi_s(\mathbf{x}) = 0; \quad s = \overline{1, \sigma}, \quad (4.153)$$

уравнения которых в частных случаях совпадают с (4.152);

2) многообразий переключения размерности $n - 2$, заданных как пересечения оптимальной поверхности (4.152) с поверхностями переключения

$$\Theta_q^{n-1}(\mathbf{x}) = 0; \quad q = \overline{1, \pi_{n-1}}, \quad (4.154)$$

и многообразий оптимального движения размерности $n - 2$, заданных как взаимопересечение двух поверхностей (4.153) или пересечение каждой поверхности (4.153) с поверхностью переключения (4.154);

3) многообразий переключения и оптимальных многообразий размерности $n - 2$, $n - 3$ и т. д.; линий переключения и оптимального движения;

л) конечной точки движения \mathbf{x}_j , заданной как пересечение линий оптимального движения с поверхностью переключения

$$\Theta_q^1(\mathbf{x}) = 0; \quad q = 1. \quad (4.155)$$

Последовательная дискретизация приведена на рис. 4.37 и включает следующие этапы.

Э т а п 1. В пространстве R^n выделяются δ -окрестности X_s оптимальных поверхностей (4.153):

$$X_s : |e_s| \leq \delta_s; \quad s = \overline{1, \sigma}. \quad (4.156)$$

Здесь $e_s = \Phi(\mathbf{x})$; $\delta_s > 0$ и области знакопостоянства R_s^n , задаваемые системами неравенств:

$$R_s^n : \{ \Delta_q^n \leq 0 \}_s; \quad s = \overline{1, \pi_n}, \quad (4.157)$$

где $\Delta_q^n = \Theta_q^n(\mathbf{x})$.

Э т а п 2. В каждом множестве X_s выделяются собственные характерные области по следующей схеме (для поверхности X_1): находят окрестности оптимальных многообразий как пересечение X_1 с X_s , $s = \overline{2, \sigma}$ и X_1 с окрестностями X_s , $s = \overline{\sigma + 1, \sigma + \pi_{n-1}}$ поверхностей переключения (4.154), где

$$X_s : | \Delta_q^{n-1} | \leq \delta_s^{n-1}; \quad s = \overline{\sigma + 1, \sigma + \pi_{n-1}},$$

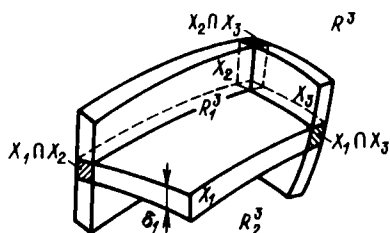


Рис. 4.37. Дискретизация пространства состояний

$\Delta_s^{n-1} = \Theta_s^{n-1}(\mathbf{x})$; $\delta_s^{n-1} > 0$ определяют области знакопостоянства, как пересечения X_1 с областями знакопостоянства R_s^{n-1} , заданными системами неравенств,

$$R_s^{n-1} : \{ \Delta_s^{n-1} \leq 0 \}_s, \quad s = \overline{1, \pi_{n-1}}$$

и т. д. во всех X_s , $s = \overline{2, \sigma}$.

Э т а п 3 ... n - 1. В окрестностях всех оптимальных многообразий размерности от $n - 2$ до 1 выделяются собственные характерные множества.

Э т а п n. В окрестностях линий оптимального движения выделяют окрестность конечной точки x_j , как пересечение окрестности линии с окрестностью X_{σ_1} поверхности переключения (4.155), где $X_{\sigma_1} : |\Delta_1^1| \leq \delta_1^1$; $\Delta_1^1 = \Theta_1^1(\mathbf{x})$; $\delta_1^1 > 0$, и находят окрестности отрезков знакопостоянства как пересечения окрестностей линий с областями знакопостоянства R_s^1 , заданными системами неравенств,

$$R_s^1 : \{ \Delta_s^1 \leq 0 \}_s; \quad s = \overline{1, \pi_1}.$$

Отметим, что в ходе дискретизации одни и те же множества могут быть получены различными путями. Так, окрестность $X_1 \cap X_2$ пересечения двух гиперповерхностей находится при анализе структуры окрестностей как первой (X_1), так и второй (X_2) поверхности (рис. 4.37). Этот факт учитывается при построении алгоритма переключения.

Синтез комбинационного автомата. Алгоритм переключения строится как комбинационный автомат (без памяти), реализующий проверку условий, полученных при дискретизации. Процедура дискретизации пространства предопределяет структуру алгоритма как комбинационного автомата, содержащего группу *локальных автоматов* A_s^i , $i = \overline{1, n}$, $s = 1, 2, \dots$. Локальные автоматы последовательно оценивают состояние ОУ по отношению к характерным областям, выявленным в ходе процедуры, т. е. автомат A^n осуществляет анализ областей в пространстве R^n , автомат A_s^{n-1} — в окрестностях X_s поверхностей оптимального движения, и т. д., автомат A_s^1 — в окрестностях линий оптимального движения.

Определим *входной вектор* автомата A^n как

$$G^n = [X_1 X_2 \dots X_\sigma R_1^n R_2^n \dots R_{\pi_n}^n]^T.$$

Здесь X_s , R_s^n — дискретные переменные, принимающие значения из алфавита $\{0, 1\}$ по правилам:

$$\begin{cases} X_s = 1 & \text{при } \mathbf{x} \in X_s; \\ X_s = 0 & \text{при } \mathbf{x} \notin X_s; \end{cases} \quad \begin{cases} R_s^n = 1 & \text{при } \mathbf{x} \in R_s^n; \\ R_s^n = 0 & \text{при } \mathbf{x} \notin R_s^n. \end{cases} \quad (4.158)$$

Определим *выходную переменную* автомата A^n как D^n с выходным алфавитом:

$$\Sigma^n = \{d_1, d_2, \dots, d_\sigma, r_1, r_2, \dots, r_{\pi_n}\}.$$

Работа автомата характеризуется переключательной функцией, связывающей входную и выходную переменные по следующему правилу:

$$\begin{cases} D^n = d_s & \text{при } X_j = 0, \quad j = \overline{1, s-1}; \quad X_s = 1; \\ D^n = r_s & \text{при } X_j = 0, \quad j = \overline{1, \sigma}; \quad R_s^n = 1, \end{cases} \quad (4.159)$$

т. е. автомат предписывает выходной переменной D^n значение в соответствии с тем, какой окрестности или оптимальной поверхности X_s (первой по счету) или какой области знакопостоянства принадлежит текущее значение вектора x .

Алгоритм работы автомата A^n приведен на рис. 4.38, он состоит из последовательно выполняемых логических операций и операторов присвоения.

Каждый из автоматов A_v^{n-1} , $v = 1, 2, \dots$, в окрестности X_s определяет множество, которому принадлежит текущее значение вектора состояния ОУ. Входной вектор автомата задается как

$$\mathbf{G}_v^{n-1} = [X_v X_{v+1} \dots X_{\sigma+\pi_{n-1}} R_1^{n-1} \dots R_{\pi_{n-1}}^{n-1}]^T,$$

где его компоненты X_s, R_s^{n-1} принимают значения из алфавита $\{0, 1\}$ по правилу (4.158), а выходная переменная D_v^{n-1} определена на множестве

$$\Sigma_v^{n-1} = \{d_v, d_{v+1}, \dots, d_{\sigma+\pi_{n-1}}, r_1, r_2, \dots, r_{\pi_{n-1}}\}.$$

Переключательная функция, связывающая выходную переменную с входным вектором \mathbf{G}_v^{n-1} , задается следующим правилом:

$$\begin{cases} D_v^{n-1} = d_s & \text{при } X_j = 0, \quad j = \overline{v, s-1}; \quad X_s = 1; \\ D_v^{n-1} = r_s & \text{при } X_j = 0, \quad j = \overline{v, \sigma + \pi_{n-1}}; \quad R_s^{n-1} = 1. \end{cases} \quad (4.160)$$

Аналогично строятся и другие локальные автоматы вплоть до A_1^n . Последние определяют принадлежность вектора состояниям подмножеств окрестностей линий оптимального движения, окрестности конечной точки и множеств знакопостоянства. Для входного вектора

$$\mathbf{G}^1 = [X_{\sigma_1} R_1^1 \dots R_{\pi_1}^1]^T$$

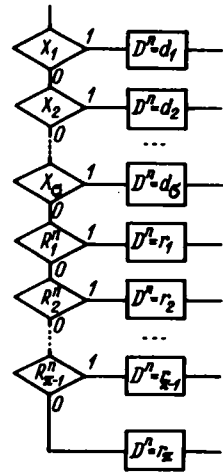


Рис. 4.38. Локальный автомат

втомат A_v^1 определяет выходную переменную $D^1 \in \Sigma^1$, где

$$\Sigma^1 = \{d_{\sigma_1}^1, r_1^1, r_2^1, \dots, r_{\pi_1}^1\},$$

в соответствии с общим алгоритмом

$$\begin{cases} D^1 = d_{\sigma_1} & \text{при } X_{\sigma_1} = 1; \\ D^1 = r_s & \text{при } X_{\sigma_1} = 0, R_s^1 = 1. \end{cases} \quad (4.161)$$

Взаимодействие локальных автоматов в рамках всего комбинационного автомата обеспечивает выбор режима $r \in \Omega$ по принадлежности вектора состояния ОУ определенной области пространства R^n . Для описания полной структуры автомата определим входной вектор $D = [D^1 D^2 \dots D^n]^T$ с областями значений переменных из пересечения множеств $\Sigma^1, \dots, \Sigma_v^{n-1}, \Sigma^n$ и выходную переменную

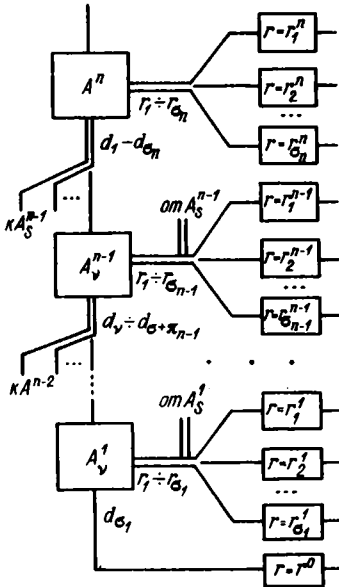


Рис. 4.39. Алгоритм переключения структуры

входной вектор $D = [D^1 D^2 \dots D^n]^T$ с областями значений переменных из пересечения множеств $\Sigma^1, \dots, \Sigma_v^{n-1}, \Sigma^n$ и выходную переменную

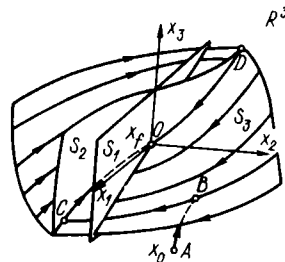


Рис. 4.40. Структура пространства состояний

$r \in \Omega$. Работа автомата определяется переключательной функцией

$$r = \rho(D),$$

которая связывает D и r по следующему правилу:

$$\begin{aligned} r &= r_s^n & \text{при } D^n &= r_s; \\ r &= r_s^{n-1} & \text{при } D^n &= d_v; \quad D^{n-1} = r_s; \quad \dots; \\ r &= r_s^1 & \text{при } D^2 &= d_v; \quad D^1 = r_s; \\ r &= r^0 & \text{при } D^1 &= d_{\sigma_1}. \end{aligned}$$

Алгоритм работы комбинационного автомата, реализующего алгоритм переключения структуры, приведен на рис. 4.39. Он

имеет разветвленную структуру, каждая ветвь которой представлена последовательно выполняемыми логическими модулями локальных автоматов A_i^f . Решение обладает высокой степенью универсальности и может использоваться для построения широкого класса многорежимных систем. Универсальность является предпосылкой автоматизации процедуры синтеза алгоритма переключения, унификации планирующей программы и других элементов программного обеспечения цифровой САУ.

Пример 4.13. Синтезируем алгоритм переключения структуры для линейного объекта управления [6]:

$$\begin{cases} \dot{x}_1 = x_2 - u; \\ \dot{x}_2 = u; \\ \dot{x}_3 = -x_3 + u, \end{cases} \quad (4.162)$$

где x_1, x_2, x_3, u — функции безразмерного времени.

Задача управления формулируется как задача оптимального по быстродействию перевода ОУ из начального состояния x_0 в состояние $x_f = 0$. Оптимальная фазовая траектория системы (см. [6]) содержит отрезки (рис. 4.40): AB — движения в R^3 (разгон); BC — движения вдоль оптимальной поверхности и CO — движения вдоль оптимальной линии переключения (торможение). Траектория завершается в окрестности x_f (режим стабилизации).

В [6] получены уравнения:

поверхности оптимального движения (поверхности переключения)

$$S_3 : \Theta_3^0(x) = \varphi_1(x) = x_3 + z - z \exp \left\{ zx_2 + \left[\frac{x_2^2}{2} + z[x_1 + x_2] \right]^{1/2} \right\} \times \\ \times \left[2 - \exp \left\{ \left[\frac{x_2^2}{2} + z[x_1 + x_2] \right]^{1/2} \right\} \right] = 0; \quad (4.163)$$

поверхности переключения

$$S_2 : \Theta_2^0(x) = x_1 + \frac{x_2^2}{2} \operatorname{sign} x_2 + x_2 = 0, \quad (4.164)$$

пересечение которой с (4.163) дает линию оптимального движения (линию переключения CO);

поверхности переключения

$$S_1 : \Theta_1^0(x) = x_2 = 0, \quad (4.165)$$

пересечение которой с линией оптимального движения образует конечную точку $x_f = 0$; где

$$z = z(x_1, x_2) = \operatorname{sign} \left(x_1 + \frac{x_2^2}{2} \operatorname{sign} x_2 + x_2 \right).$$

Существим дискретизацию пространства R^3 .

Э т а п 1. Выделим окрестности поверхности (4.163) $X_1 : |e_1| \leq \delta_1$, а также области знакопостоянства управления:

$$\begin{aligned} R_1^0 : \Delta_1 > 0 & \quad (u = -1); \\ R_2^0 : \Delta_1 < 0 & \quad (u = 1), \end{aligned}$$

где $e_1 = \Delta_1 = \Theta_3^0(x)$.

Этап 2. В окрестности X_1 выделим окрестность оптимальной линии как пересечение X_1 и X_2 (X_2 — окрестность поверхности переключения (4.164), т. е. $X_2: |e_2| \leq \delta_2$), а также области знакопостоянства:

$$R_1^2: \Delta_2 > 0 \quad (u = 1);$$

$$R_2^2: \Delta_2 < 0 \quad (u = -1),$$

где $e_2 = \Delta_2 = \Theta_2^1(x)$.

Этап 3. В окрестности линии оптимального движения выделим окрестность конечной точки, как пересечение X_1 , X_2, \dots и окрестности поверхности (4.165) $X_3: |e_3| \leq \delta_3$, а также области знакопостоянства:

$$R_1^3: \Delta_3 > 0 \quad (u = -1);$$

$$R_2^3: \Delta_3 < 0 \quad (u = 1),$$

где $e_3 = \Delta_3 = \Theta_3^1(x)$.

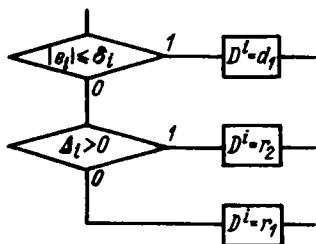


Рис. 4.41. Локальный автомат для ОУ (4.162)

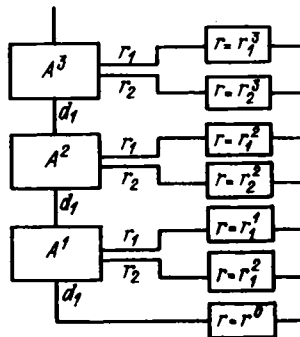


Рис. 4.42. Алгоритм переключения структуры для ОУ (4.162)

Определим режимы работы системы r_1^i и r_2^i , $i = 1, 2, 3$, соответствующие различным знакам управления u при движении в R_1^i и R_2^i , а также режим стабилизации r^0 в точке $x = 0$.

Алгоритм переключения структуры содержит три автомата:

- 1) A^3 — с входом $G^3 = [X_1 \mid R_1^3 \mid R_2^3]^T$ и выходом D^3 ;
- 2) A^2 — с входом $G^2 = [X_2 \mid R_1^2 \mid R_2^2]^T$ и выходом D^2 ;
- 3) A^1 — с входом $G^1 = [X_3 \mid R_1^1 \mid R_2^1]^T$ и выходом D^1 , причем выходной алфавит $A^1 - A^3$ представлен множеством

$$\Sigma = \{d_1, r_1, r_2\}.$$

Структура локального автомата A^i , $i = 1, 2, 3$, приведена на рис. 4.41. Вектор $[D^1 \mid D^2 \mid D^3]^T$ рассматривается как вход общего комбинационного автомата алгоритма переключения структуры, представленного на рис. 4.42. Алгоритм осуществляет последовательный анализ состояния объекта управления (4.162), выявляет его принадлежность к одному из множеств, определенных процедурой дискретизации, и присваивает переменной режима r одно из значений r_s^i .

Управляющая программа цифровой САУ. Проанализируем типовые алгоритмы расчета управляющего воздействия, подключаемые в различных режимах.

Алгоритмы пространственного управления описываются общим выражением

$$P_s^i: u = E_1 \bar{U} + E_2 [V(\bar{x}) - K(\bar{x}) e], \quad (4.166)$$

где $[E_1 | E_2]$ — матрица перестановок; \bar{U} — оптимизирующая компонента; $V + Ke$ — координированный алгоритм стабилизации движения по многообразию (4.11); e — вектор относительного движения (4.18).

Рассмотрим частные реализации выражения (4.166) в различных режимах работы системы, ограничивая анализ широко распространенным случаем двухканального управления (см. п. 1.1), т. е. $h = 2$.

На участке разгона ($r = r_s^n$) движение системы происходит вне окрестностей оптимальных многообразий, и обе координаты вектора управления выполняют оптимизирующие функции, т. е. можно принять $E_1 = I$, $E_2 = 0$ и, следовательно,

$$u = \bar{U} \in \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right\}.$$

На участке движения по гиперповерхности ($r = r_s^{n-1}$) одна из компонент управления осуществляет оптимизацию продольного движения, а другая — стабилизацию относительного движения. Выбор стабилизирующей и оптимизирующей компонент производится за счет назначения матриц E_1 и E_2 :

$$E_1 \in \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}, \quad E_2 \in \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\};$$

в формуле (4.166) $\bar{U} \in \{1, -1\}$; $V(\bar{x})$ — скалярные функции координат редуцированного вектора состояния \bar{x} ; e — скалярная функция относительного движения (см. п. 4.5). Функции K , V и e задаются индивидуальными формулами в зависимости от индекса s .

На участках движения по оптимальным многообразиям размерности $n - v$, т. е. при $r = r_s^i$, $i = 0, n - 2$, обе координаты вектора управления выполняют стабилизирующие функции, $E_1 = 0$, $E_2 = I$ и алгоритм (4.165) принимает вид:

$$u = V(\bar{x}) - K(\bar{x})e,$$

где $V(\bar{x}) = \{V_j\}$; $j = 1, 2$; e — v -мерный вектор, определяющийся общим выражением (4.18); $K(\bar{x})$ — матрица $v \times 2$, причем V_j и коэффициенты матрицы K являются функциями вектора продольного движения \bar{x} и вычисляются по индивидуальным формулам для каждого из режимов.

Типовой алгоритм расчета управления в режиме r_s^i (локальный регулятор P_s^i) представлен на рис. 4.43. Алгоритм содержит модуль формирования векторов e и \bar{x} , блок вычисления функций $V(\bar{x})$, $K(\bar{x})$ (в соответствии с выражениями из пп. 4.5, 4.6) и модуль вычисления управления. Каждый из таких алгоритмов имеет индивидуальные векторные компоненты E_1 , E_2 , \bar{U} и соб-

ственные формулы расчета коэффициентов вектор-функций $V(\bar{x})$ и $K(\bar{x})$.

Укрупненный алгоритм расчета управления (управляющая программа) оптимальной системы приведен на рис. 4.44. Алгоритм представлен модулями ввода—вывода, алгоритмом переключения структуры (планирующая программа ПП) и группой локальных регуляторов P_i^l .

Пример 4.14. Локальные алгоритмы САУ оптимального по быстродействию управления объектом (4.162) имеют вид:

$$\begin{aligned} P_1^3: u &= -1; & P_2^3: u &= 1; \\ P_1^2: u &= 1_{\Delta} - k_1 e_3; & P_2^2: u &= -1_{\Delta} - k_2 e_3; \\ P_1^1: u &= -1_{\Delta} - k_1 e_3 - k_2 e_2; & P_2^1: u &= 1_{\Delta} - k_1 e_3 - k_2 e_2; \\ P_0: u &= -k_{10} e_3 - k_{20} e_3 - k_{30} e_1, \end{aligned}$$

где $k = k(x_1, x_2)$; $0 < 1_{\Delta} < 1$; $k_1 = k_1(\bar{x})$; $k_2 = k_2(\bar{x})$; $\bar{x} = x_2$.

Управляющая программа, представленная планирующей программой, которая реализует алгоритм переключения структуры (см. рис. 4.42), и локальными регуляторами P_i^l , приведена на рис. 4.45.

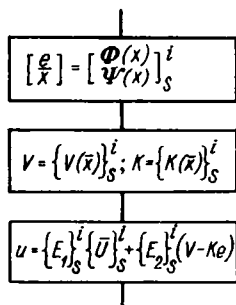


Рис. 4.43. Регулятор пространственного движения

Унифицированная управляющая программа. Сложности расчета и индивидуальные особенности аналитических выражений для компонент уравнения (4.166) являются основным недостатком рассмотренного класса локальных алгоритмов. Многообразие расчетных формул приводит к усложнению программы, значительному увеличению времени разработки, программирования и отладки прикладного обеспечения цифровых САУ, а также настройки спроектированной системы.

Рассмотрим одно из возможных направлений унификации алгоритмических модулей (А. с. № 1270854) [60, 96], основанное на методах аппроксимации (см. п. 4.6). Аналитические выражения компонентов уравнения (4.166) будем искать в виде:

$$\begin{cases} V_a(\bar{x}) = G^T z(\bar{x}); \\ K_{aj}(\bar{x}) = H_j^T z(\bar{x}) \end{cases} \quad (4.167)$$

(G^T и H_j^T матрицы коэффициентов аппроксимации; $\{K_{aj}\} = K_a(\bar{x})$, $j = \overline{1, h}$; $z(\bar{x})$ — вектор базисных функций). В частном случае, при использовании степенной аппроксимации, в качестве компонент z_k вектора $z(\bar{x})$ выбираются произведения степеней координат вектора состояния \bar{x} , т. е.

$$z_k = \bar{x}_1^{p_1} \bar{x}_2^{p_2} \bar{x}_3^{p_3} \dots,$$

где p_1, p_2, p_3, \dots — целые показатели степени.

Значения матриц C^T и H_j^T для каждого режима работы $r = r_s^i$ находят в результате процедуры аппроксимации (см. п. 4.4) из условия минимизации ошибок $\|V(\bar{x}) - V_a(\bar{x})\|$ и $\|K_j(\bar{x}) - K_{aj}(\bar{x})\|$ для заданного множества дискретных значений $\bar{x} \in \bar{X}$. Причем величины $V(\bar{x})$ определяются из выражения (4.97) или (4.110), а значения $K(\bar{x})$ — с привлечением процедуры решения уравнения (4.134) или (4.150). Для использующихся здесь замк-

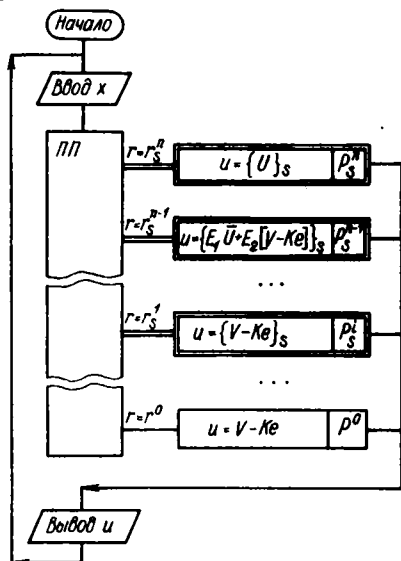


Рис. 4.44. Алгоритм управления оптимальной системы

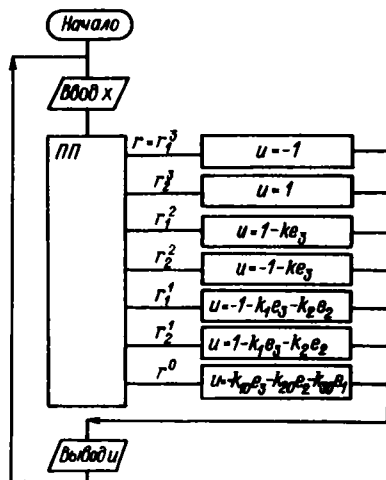


Рис. 4.45. Алгоритм управления для ОУ (4.162)

нутых алгоритмов управления, содержащих обратные связи по отклонению, излишне жесткие требования к точности аппроксимации можно не предъявлять.

В результате процедуры аппроксимации получим массивы коэффициентов $\{C\}_s^i$ и $\{H\}_s^i$, которые вместе с уже известными массивами $\{E_1\}_s^i$, $\{E_2\}_s^i$ и $\{\bar{U}\}_s^i$ упорядоченным образом размещаются в памяти управляющей микроЭВМ.

Управляющая программа САУ (с использованием унифицированных выражений (4.167) для расчета V и K) строится на базе единого для всех режимов комплекта алгоритмических модулей, осуществляющих расчет управления с привлечением массивов коэффициентов, которые требуются в данном режиме $r = r_s^i$. Структура программы приведена на рис. 4.46.

Пример 4.15. На рис. 4.47 приведен начальный участок фазовой траектории оптимального по быстродействию разгона двухканального электропривода

(1.2), (1.4), (1.5) (см. (А. с. № 955475, А. с. № 957399, А. с. № 964938, А. с. № 1228207, А. с. № 1270854) [22, 60]):

$$\dot{x}_1 = 0,23i_{я}x_3 - \mu_c(x_1);$$

$$\dot{x}_2 = -0,62x_2 + 0,93u_1;$$

$$\dot{x}_3 = -0,3x_3 + 0,45u_2,$$

где x_1 — угловая скорость двигателя; x_2 — выходное напряжение усилителя; x_3 — ток обмотки возбуждения; u_1 — напряжение на входе усилителя; u_2 — напряжение возбуждения; $i_{я}$ — ток якоря, определяемый выражением (1.3) при $T_{\theta} = 0$: $i_{я} = 2,4x_2 - 1,4x_1x_3$; μ_c — момент сопротивления ($\mu_c = |x_1| |x_2|$).

Ограничения на переменные состояния ОУ задаются как $|i_{я}| \leq 2$; $|x_3| \leq 1$, а на координаты вектора управления u_j определяются выражением (4.2), $j = 1, 2$.

Оптимальная траектория содержит три отрезка:

1) OA — разгон без ограничений на переменные состояния (режим $r = r^3$);

2) AB — движение по граничной поверхности

$$S_1: i_{я} = 2 \quad (4.168)$$

(режим $r = r_2$);

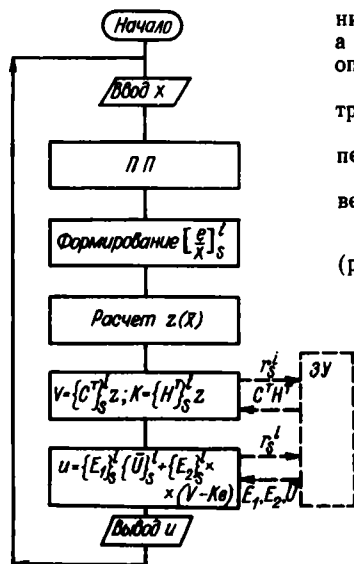


Рис. 4.46. Унифицированный алгоритм управления

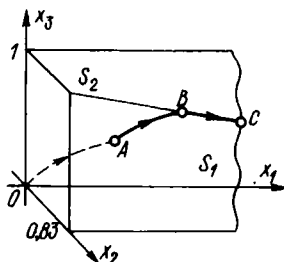


Рис. 4.47. Фазовая траектория оптимального разгона

3) BC — движение по линии, т. е. по пересечению граничных поверхностей (4.168) и

$$S_2: x_3 = 1 \quad (4.169)$$

(режим $r = r_2$).

Для рассматриваемых режимов методами из пп. 4.5, 4.6 получены следующие локальные алгоритмы управления:

$$P^3: u_1 = 1, u_2 = 1;$$

$$P^2: u_1 = V(x_1, x_2) - k(x_2)e_2, u_2 = 1;$$

$$P^1: u_1 = V_1(x_1) - k_1e_2, u_2 = V_2 - k_2e_1,$$

где $V(x_1, x_2) = 0,56 + 0,28x_1 + 0,20x_1x_2 + 0,28x_2^2 - 0,14x_1^2x_2$; $k(x_2) = 0,56 - 0,14x_2^2$; $e_2 = i_{я} - 2$; $V_1(x_1) = 0,84 + 0,39x_1 - 0,14x_1^2$; $k_1 = 0,42$; $V_2 = 0,66$; $k_2 = 1,33$; $e_1 = x_3 - 1$.

При синтезе обратных связей выбирались степени устойчивости $\eta = 0,9 \text{ с}^{-1}$ на поверхности (4.168) и $\eta = 1,8 \text{ с}^{-1}$ на поверхности (4.169).

Таблица 4.2

Коэффициенты унифицированной системы

Режим	C_1^T	C_2^T	H_1^T	H_2^T
r^2	[1 : 0 : 0]	[1 : 0 : 0]	[0 : 0 : 0]	[0 : 0 : 0]
r^3	[0,75 : 0,01 : 0,39]	[1 : 0 : 0]	[0,56 : 0 : -0,15]	[0 : 0 : 0]
r^3	[0,86 : 0,11 : 0]	[0,66 : 0 : 0]	[0,42 : 0 : 0]	[1,33 : 0 : 0]

Так как выражения для V и K найдены в виде отрезков степенных рядов, то они могут быть непосредственно использованы для построения унифицированного регулятора. С помощью процедуры аппроксимации их можно только упростить. Выбирая $z(x) = [z_1 z_2 z_3]^T = [1x_1 x_2]^T$ для заданного диапазона вариаций переменных $x_1 \in [0,1]$, $x_2 \in [0,1]$, найдем аппроксимирующие функции:

$$V_a(x_1, x_2) = 0,75 + 0,01x_1 + 0,39x_2;$$

$$V_{1a}(x_1) = 0,86 + 0,11x_1; k_a(x_2) = 0,56 - 0,15x_2.$$

Тогда уравнения регулятора можно записать в виде:

$$p^i : \begin{cases} u_1 = \{C_1^T\}^i z + [\{H_1^T\}^i z] e_1; \\ u_2 = \{C_2^T\}^i z + [\{H_2^T\}^i z] e_2, \end{cases}$$

где векторы C_1 , C_2 , H_1 , H_2 для каждого режима выбираются из табл. 4.2.

Использование регуляторов унифицированной структуры позволяет снизить вычислительную сложность управляющих программ, упростить процедуру синтеза и программирования. В результате создаются предпосылки унификации программного обеспечения для однотипных объектов управления; настройка унифицированной системы сводится к подбору матричных коэффициентов C^T и H^T , значения которых рассчитываются при аппроксимации и корректируются при испытаниях САУ ручными или адаптивными методами.

Глава 5

ТЕХНИЧЕСКИЕ СРЕДСТВА МИКРОПРОЦЕССОРНЫХ СИСТЕМ УПРАВЛЕНИЯ

5.1. Принципы организации вычислительных средств на основе секционированных микропроцессорных наборов

Основной элементной базой при разработке и изготовлении цифровых систем управления являются *микропроцессорные наборы*. В состав наборов входят БИС, позволяющие реализовать на одном или нескольких элементах функции вычислительного устройства, связанные с вводом, и обработкой данных. *Микропроцессорные системы*, построенные на основе этих наборов, в зависимости от параметров используемых БИС, объема основной памяти и состава внешних устройств охватывают широкий класс средств обработки данных — от простейших микроконтроллеров до сложных многомашинных управляющих комплексов.

Встроенные цифровые регуляторы, выполненные на основе микропроцессорных наборов, вытесняют такие традиционные локальные средства управления в технологических процессах, как промышленные контроллеры на релейных и аналоговых элементах [61]. Соответствующие микропроцессорные средства, а также любые специализированные микропроцессорные системы, предназначенные для локального управления, называют *микроконтроллерами*. К ним относятся, в частности, локальные регуляторы, реализующие специальные алгоритмы управления при жестких временных ограничениях, выполненные на основе *секционированных микропроцессорных наборов* с микропрограммным управлением.

Элементы набора секционированных микропроцессорных БИС используют при проектировании высокопроизводительных средств управления и специализированных цифровых устройств различного назначения. Универсальные микропроцессорные системы с программным управлением, выполненные на этих наборах, могут превосходить по параметрам существующие серийные микро-ЭВМ и относятся к классу мини-, супермини-ЭВМ. В то же время специализированные микропроцессорные средства с микропрограммным управлением, выполняющие функции локального цифрового регулятора или формирования команд управления стартовыми устройствами, относятся к классу микроконтроллеров.

Проектирование цифровых САУ на секционированных микропроцессорных наборах включает разработку полного комплекта схем системы и микропрограммирование.

Структура микропрограммируемого микроконтроллера приведена на рис. 5.1. *Микропроцессор* выполнен на унифицированных элементах секционированного микропрограммируемого микропроцессорного набора, в который входят следующие блоки: арифметико-логический (АЛБ), адресуемых регистров общего назначения (РОН), микропрограммного управления (БМУ) и генератор синхросигналов (ГС).

Арифметико-логический блок предназначен для выполнения элементарных одноктактных микроопераций сложения, вычитания, сдвига, логических сложения и умножения и др.; блок адресуемых регистров общего назначения — для хранения данных, оперативно используемых в микрооперациях АЛБ; блок микропрограммного управления — для формирования одноктактных микрокоманд функциональными элементами в соответствии с микропрограммой; ГС — для синхронного переключения регистров. Генератор позволяет сформировать серию синхросигналов (СС) в автоматическом режиме работы и одиночные синхросигналы в пошаговом режиме, необходимым при отладке микропрограмм (внешний переключатель А—Ш). Запуск и останов ГС осуществляется внешним переключателем П—С.

Основная память физически разделяется на два блока: ОЗУ и ПЗУ. ОЗУ используется при вводе данных и для временного хранения данных в процессе микропрограммной обработки; ПЗУ — для хранения констант, необходимых в алгоритмах управления. Функциональные элементы соединяются общей ШИ для передачи данных и ША — для передачи адреса при обращении к ОП.

Существует широкая номенклатура секционированных микропроцессорных наборов, позволяющих реализовать микроконтроллер (рис. 5.1) на одной печатной плате. Характеристики наборов (табл. 5.1) удовлетворяют самым разнообразным требованиям к параметрам локальных цифровых регуляторов. Длительность такта микрокоманды (периодичность синхросигнала) характеризует быстродействие микроконтроллера и может быть выбрана от 0,05 до 1 мкс. Потребляемая мощность при этом изменяется от нескольких ватт до микроватт. Каждый микропроцессорный набор

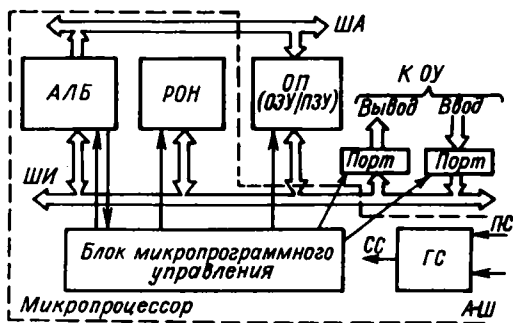


Рис. 5.1. Структура микропрограммируемого микроконтроллера

ввода I_1, I_2 , предназначенные для приема данных с преобразователей информации и пульта управления; регистры O_1, O_2 — для вывода цифрового кода управляющего воздействия в ОУ. Универсальный набор операций в АЛБ, последовательная по шагам синхронизация и управление преобразованиями, обеспечиваемые БМУ, гарантируют возможность реализации алгоритма на микропрограммном уровне.

Совокупность элементарных операций, исполняемых в различных элементах контроллера в течение такта синхронизации, объединяются в *микрокоманду*, которая однозначно задается кодом микрокоманды. Структура кода микрокоманды:

[АЛБ] [АДР] [ОП] [В/В],

где [АЛБ] — поле кода управления арифметико-логическим блоком; [АДР] — код управления схемой адресации; [ОП] — код управления основной памятью; [В/В] — код управления вводом (I_1, I_2) и выводом (O_1, O_2).

При этом в одном такте по микрокоманде микроконтроллера выполняются операции в АЛБ, в схеме формирования адреса микрокоманды, чтения или записи в ОП и ввода или вывода. Результаты операций фиксируются в соответствующих регистрах АЛБ, БМУ, БР, O_1, O_2 в конце такта по положительному перепаду синхросигнала СС. Таким образом, для микрокоманд контроллера характерен внутренний микропараллелизм, который может быть использован для ускорения обработки при реализации алгоритмов управления САУ. В частности, структура микроконтроллера может включать несколько параллельных АЛБ, а многократно исполняемая последовательность операций с различными данными может быть реализована последовательным соединением АЛБ в виде *конвейера*.

Наиболее длительная микрокоманда включает операцию в АЛБ, формирование признаков результата (Z — нуль результата, C — переполнение или заём), формирование адреса следующей микрокоманды, чтение ее из ПЗУ и фиксацию по перепаду СС на регистре микрокоманд (РМК). Период синхронизации определяется максимальной задержкой $t_{СС} \approx t_{АДР} + t_{ПЗУ} + t_{АЛБ}$, где $t_{АЛБ}$ — время выполнения наиболее длительной операции в АЛБ; $t_{АДР}$ — время формирования адреса микрокоманды, $t_{ПЗУ}$ — время выборки из ПЗУ.

Если адрес следующей микрокоманды не зависит от признаков Z и C , то выполняемая в АЛБ текущая операция и безусловная выборка из ПЗУ следующей микрокоманды совместимы и могут быть выполнены параллельно. Длительность такта синхронизации сокращается в 1,5—2 раза и определяется как

$$t_{СС} \approx \max \{t_{АЛБ}, t_{АДР} + t_{ПЗУ}\}.$$

Выбор управления функциональными элементами схемы с учетом синхронизации, формирование и кодирование управляющего

слова, выбор длительности синхротакта являются задачами *микропрограммирования*.

Микроконтроллер (рис. 5.2) включает блок управления *прерываниями* (БУП), который выполнен на элементах К1804ВН1 и обеспечивает связь с объектом управления в реальном времени. Отсчет реального времени осуществляется счетчиком-таймером T , окончание отсчета сигнализируется запросом прерывания, поступающим в блок управления прерываниями. Запросы прерывания в БУП поступают также с пульта управления при оперативном вмешательстве, изменении режима работы микроконтроллера и с объекта управления при возникновении критических ситуаций.

Схема БИС КР1804ВН1 приведена в [53, 100] и включает элементы, позволяющие в одном такте по микрокоманде выполнить следующие элементарные операции: фиксацию во входном буфере сигналов прерывания; выбор наиболее приоритетного сигнала, сравнение его по приоритету с текущим прерыванием; формирование на выходе кода прерывания ($KП$), сигнала запроса прерывания ($ЗП$). Один из вариантов быстрого входа в прерывание предусматривает подмену текущего адреса выборки микрокоманды новым адресом, соответствующим началу микропрограммы обработки прерывания и считываемого с регистра адреса прерывания ($РАП$).

Вход в прерывание осуществляется за один такт (0,2 мкс), что достаточно для управления быстрыми процессами, происходящими в реальном времени; обработка прерывания, т. е. принятие решения, может быть выполнена за несколько тактов (1—2 мкс). Приведем некоторые примеры прерываний, которые могут быть вызваны: 1) таймером, регистрирующим события реального времени с точностью отсчета временного интервала 0,2 мкс; 2) готовностью данных или устройств ввода—вывода, завершением процесса управления по внешним признакам окончания механического перемещения в ОУ (в этом случае за 1—2 мкс данные будут переданы в основную память и приостановлена микропрограмма формирования управления); 3) сигналом сбоя питания, при этом состояние регистров контроллера сохраняется в энергонезависимой памяти. Для ввода данных формируются сигналы чтения регистров I_1 и I_2 , а при выводе — сигналы записи в регистры O_1 и O_2 . Таким образом, операции ввода—вывода выполняются за один такт (0,2 мкс) и могут быть совмещены с записью — чтением данных из ОЗУ и операциями в АЛБ.

Отметим следующие достоинства контроллеров с микропрограммным управлением выполненных на секционированных микропроцессорных наборах: 1) высокие быстродействие и точность, которые обеспечиваются возможностью выбора соответствующей разрядности данных, обрабатываемых в АЛБ за один такт; 2) быстрый ввод—вывод с прямым микропрограммным управлением, позволяющий регистрировать достаточно быстрые процессы в объектах управления; 3) быстрый вход в прерывание в минимальный

Сравнительные характеристики микропроцессоров в цифровых САУ

Характеристика	K1804	KP580	K581	KP1810	K1801 BM2
Время выполнения операций с 16-разрядными числами, мкс:					
сложения	0,2	15	4	1	1,2
умножения	5	300	80	20	25
Задержка, мкс:					
входа в прерывание	0,2	10	2,5	1	2,5
ввода—вывода	0,2	4	6	2	4
Интервал внешних прерываний, мкс	2	50	50	10	20
Рассеиваемая мощность	8	0,8	1	1,8	1
Число модулей в контроллере, шт.	30	8—15	25	15—20	15—20
Разрядность процессора, бит	16	8	16	16	16

доступный интервал между прерываниями в микропроцессорных системах; 4) возможность специализации структуры для эффективной параллельной или конвейерной реализации алгоритмов управления.

Основные технические характеристики микропрограммируемого микроконтроллера, реализованного на наборе K1804, приведен в табл. 5.2. По быстродействию контроллер превосходит однокристалльные микропроцессоры, однако характеризуется высокой трудоемкостью проектирования, изготовления и отладки. Значительное снижение трудоемкости и продолжительности этапов микропрограммирования микроконтроллеров на секционированных БИС обеспечивают средства автоматизации. Инструментальные системы [53, 54] автоматизации микропрограммирования содержат микроассемблеры, позволяющие использовать символическую запись микрокоманд и обеспечивающие кодирование и размещение микропрограмм в ПЗУ. Для отладки в инструментальные системы включены схемы, эмулирующие БМУ и позволяющие выполнять отладку микропрограмм в ОЗУ.

5.2. Принципы организации вычислительных средств на основе однокристалльных микропроцессоров

Наиболее существенное снижение трудоемкости реализации алгоритмов, повышение уровня алгоритмизации, универсальность и гибкость позволяют получить программное управление, в котором используются многотактные команды.

Выполнение команд на уровне функциональных элементов обеспечивается микропрограммным управлением, а прикладные

задачи программируются в системе команд и выполняются на более высоком уровне — программном. Для цифрового управления может быть разработана специальная система команд и специализированная микроЭВМ. Однако предпочтение отдается стандартным системам команд, которые поддерживаются микропроцессорными наборами, серийными микроЭВМ, обширным программным обеспечением и развитыми инструментальными системами. Широкое распространение получили программируемые микропроцессорные наборы, включающие однокристалльные микропроцессоры, которые в одном элементе реализуют функции управления выборкой команд, декодированием и исполнением. Структура программи-

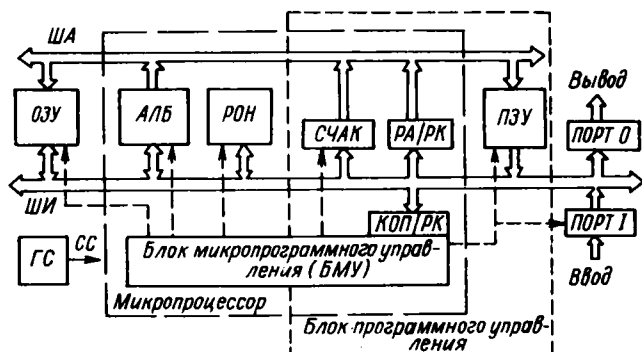


Рис. 5.3. Структура микроЭВМ с программным управлением

руемого контроллера (рис. 5.3) в отличие от микропрограммируемого контроллера (см. рис. 5.1) содержит дополнительные элементы, обеспечивающие программный принцип управления: счетчик адреса команд (*СЧАК*); регистр команд (*РК*) и *ПЗУ*, в котором хранится программа.

Микропроцессор включает: рассмотренные в контроллере (см. рис. 5.1 и 5.2) средства преобразования данных (*АЛБ*, *РОН*, *БМУ*); *СЧАК*, предназначенный для хранения и формирования адреса команды и *РК* для хранения команды, выбираемой из *ПЗУ*. В данном случае *РК* разделяется с учетом принятого кодирования команд на два регистра: адреса (*РА*) и кода операции (*КОП*). Регистр адреса выполняет также роль буферного регистра и может быть использован для хранения и формирования адреса при обращении к *ОЗУ* и *ПЗУ*. *РКОП* содержит часть кода команды, которая декодируется блоком микропрограммного управления и определяет выбор микропрограммы для выполнения операции. Программно управляемая структура содержит также основную память (*ОП*), которая разделяется на *ОЗУ* (для хранения оперативных данных) и *ПЗУ* (для хранения констант и программы).

Блок микропрограммного управления обеспечивает управление выборкой команды из *ПЗУ*, декодирование *КОП*, управление

Характеристики однокристалльных микропроцессоров

Тип микропроцессора	Разрядность, бит	Объем адресного пространства, Кбайт	Тактовая частота, МГц	Среднее время сложения, мкс	Среднее время умножения, мкс	Статическая мощность рассеивания, Вт
KP580ИК80	8	64	2	2,5	—	0,8
KP1810BM86	16	1000	5	1	20	1,8
K1801BM2	16	64	5	1,2	25	1,0

формированием адреса следующей команды в *СЧАК* и исполнением команды. БИС микропроцессора определяет структуру цифровой системы, принципы организации и управления запоминающими устройствами и организацию ввода—вывода. Фиксированная система команд микропроцессора содержит все необходимые операции с данными (достаточные для реализации алгоритмов прикладных задач) и определяет доступ к ним в запоминающих устройствах.

Память программ может быть выполнена в *ПЗУ* и *ОЗУ*, которые связаны между собой общими шинами адреса (*ША*) и данных (*ШД*). Благодаря этому программы могут быть введены, выведены и обработаны как данные, при этом легко изменить режим управления в цифровой *САУ*, вызывая новую программу из внешней памяти.

Выпускаемые серийно однокристалльные микропроцессоры (табл. 5.3) отличаются системой команд, организацией внутренней регистровой памяти *РОН*, объемом адресного пространства *ОП*, организацией управления *ОП* и вводом—выводом. Все особенности структуры микропроцессора, функциональные возможности и условия применения в цифровых системах управления отражаются в системе команд. Команда включает функционально законченную операцию преобразования данных или управления ветвлениями и переходами в программе и представлена двоично-кодированным словом, которое можно хранить в основной памяти.

В одной команде микропроцессора могут быть выполнены обычные операции с простейшими типами данных: целыми и дробными числами, кодами символов. Выделяют основной тип данных, на эффективную обработку которого ориентирована структура и система команд микропроцессора. Для 8-разрядных микропроцессоров основной тип данных — 8-разрядный двоичный код; для 16-разрядных микропроцессоров — 16-разрядное двоичное число с фиксированной запятой. Команды 16-разрядных микропроцессоров позволяют организовать эффективную обработку данных в цифровых *САУ*, связанных с объектом управления 10-, 12-разрядными *ЦАП* и *АЦП*. Вместе с тем универсаль-

ность любой системы команд позволяет использовать и 8-разрядные микропроцессоры для аналогичных цифровых САУ. Несмотря на то, что в этом случае снижается быстродействие САУ, другие характеристики улучшаются. В частности, снижаются стоимость, габаритные размеры и потребляемая мощность системы. В связи с этим 8-разрядные микропроцессоры применяются, главным образом, в контроллерах локальных регуляторов и инерционных цифровых САУ.

Восьмиразрядный микропроцессор КР580ИК80 [3, 72] содержит регистровую память *РОН*, которая может быть использована, в зависимости от команды, в виде шести 8-разрядных регистров (6×8) или 16-разрядных регистров (3×16), и универсальный регистр-аккумулятор (*АС*). Основными типами данных в микропроцессоре являются 8-разрядное двоичное число и двоичный код. Система команд включает 8-разрядные арифметико-логические операции $\{+, -, \&, \vee, \oplus, \text{сравнение}, \uparrow, \pm 1, \text{циклические сдвиги}\}$, 16-разрядные операции $\{\pm 1, +\}$, команды управления ветвлениями и переходами в программе.

Специальные команды ввода (порт (*j*) \rightarrow *АС*) и вывода (*АС* \rightarrow порт (*j*)), где *j* — прямой 8-разрядный адрес регистра (порта) ввода или вывода. По командам ввода формируется специальный сигнал чтения порта (*ior*), а в команде вывода сигнала записи в порт (*iow*).

16-разрядная шина адреса, 8-разрядная двунаправленная шина данных и управляющие линии микропроцессора КР580 образуют интерфейс (*микрошину*), легко сопрягаемый с элементами *ОЗУ*, *ПЗУ* и регистрами ввода-вывода (рис. 5.3).

Доступ к данным в *ОП* обеспечивается различными способами адресации: абсолютная, косвенная регистровая, непосредственная, стековая.

16-разрядный микропроцессор К1801ВМ2 [48] отличается высокой степенью интеграции и реализует систему команд, совместимую с системой команд распространенных 16-разрядных мини-ЭВМ *СМ-4*, *СМ-1420* и др. Микропроцессор содержит восемь 16-разрядных *РОН* (r_0, r_1, \dots, r_7), причем r_7 является *СЧАК*. 16-разрядная совмещенная шина *ША—ШИ* позволяет адресовать 64 Кбайт; по этой шине также передаются 16-разрядные слова *ОП* и байты данных при вводе-выводе.

Основной тип данных в микропроцессоре — 15-разрядные двоичные числа со знаком (числа представлены в 16-разрядном формате). Отрицательное число *A* хранится в дополнительном коде ($A = 2^{16} - |A|$ — для целых чисел или $A = 2 - |A|$ для дробных). Таким образом, диапазон целых чисел $-2^{16} \dots 2^{16} - 1$, а диапазон дробных чисел $-1/2 \dots 1/2 - 2^{-16}$. Положительные числа отличаются от отрицательных старшим битом (для отрицательных чисел бит равен единице, для положительных — нулю).

Система команд включает 16-разрядные арифметико-логические операции с двоичными числами и кодами $\{+, -, \times, /, \vee,$

&, ± 1 , $\pm C$, \uparrow , сдвиги циклические и арифметические}. Все операции, кроме $\{+, -, \times, /\}$, могут быть выполнены с 8-разрядными двоичными кодами (байтами). Система команд отличается большим разнообразием способов адресации данных в ОП, включает необходимые команды ветвления и обращения к подпрограммам. Порты ввода—вывода адресуются как ячейки основной памяти. При этом в команде можно совмещать ввод с арифметико-логическими операциями. Устройство ввода—вывода идентифицирует свой адрес и осуществляет обмен данными с микропроцессором через стандартный интерфейс МПИ, включающий совмещенную информационно-адресную шину (ША—ШИ) и линии управления.

По сравнению с 8-разрядным микропроцессором КР580ИК80 производительность вычислений средней точности (до 16 разрядов) возрастает в 4—5 раз, а объем памяти программ (благодаря более эффективной системе команд) сокращается в 3—4 раза. Микропроцессор К1801ВМ2 используется в серийных одноплатных микроЭВМ различного назначения, которые могут входить в состав технических средств быстродействующих многорежимных цифровых САУ.

Высокая производительность 16-разрядного микропроцессора КР1810ВМ86 [25] на уровне процессора мини-ЭВМ обеспечивается не только схмотехникой и качественной технологией, но и существенным усовершенствованием структуры по сравнению с ранее рассмотренной традиционной структурой (см. рис. 5.3). В микропроцессоре предусмотрена совместимость с широко распространенным 8-разрядным микропроцессором КР580ИК80. Совместимость сохраняется на функциональном уровне «снизу—вверх», благодаря этому возможна замена команд микропроцессора КР580ИК80 командами микропроцессора КР1810ВМ86. Совместимость позволяет сохранить существующий задел программного обеспечения 8-разрядных микроЭВМ и использовать его в 16-разрядных микроЭВМ с более высокой производительностью.

Микропроцессор КР1810ВМ86 формирует 20-разрядный адрес ОП, что соответствует адресному пространству объемом 1 Мбайт и имеет совмещенную шину для передачи адреса и данных, которые разделяются внешними схемами на две шины ША и ШИ, образующие вместе с линиями управления интерфейс микроЭВМ — *микрощину*.

Восемь 16-разрядных РОН организованы так, что обеспечивается доступ к 16-разрядным словам или байтам. Разнообразные способы адресации и переменный формат команд, который может занимать от одного до шести байтов, обеспечивают компактное кодирование программ.

Арифметико-логические операции выполняются с 8-разрядными и 16-разрядными кодами и 15-разрядными числами со знаком. Отрицательные числа представлены в дополнительном коде. При выполнении операций формируются признаки результата Z,

S, *C* и *V* (переполнение). Система команд включает операции со строками и сдвига на заданное число разрядов. Для управления ветвлениями и переходами в программе используются: разнообразные безусловные и условные переходы по признакам, инверсиям признаков и различным их комбинациям; команды управления циклом по счетчику и обращение к подпрограмме.

Ввод—вывод организуется так же, как в микропроцессоре КР580ИК80, т. е. с помощью специальных команд ввода (порт (*j*) → *AC*) и вывода (*AC* → порт (*j*)). Адрес регистра-порта ввода—вывода *j* может быть задан непосредственно в команде (короткий адрес) и косвенно — в регистре *DX* (длинный 16-разрядный адрес). Обмен при вводе—выводе осуществляется словами или байтами с аккумулятором *AH*.

Отметим преимущества программного управления, реализуемого однокристалльными микропроцессорами, по сравнению с микропрограммным (на секционированных наборах): 1) компактное кодирование управляющей программы; 2) возможность хранения программы в ОЗУ, что позволяет выполнить перезагрузку программ в многорежимных системах управления и обеспечивает универсальность применения; 3) сравнительно большой объем доступной оперативной памяти, разнообразные и гибкие средства доступа к данным, определяемые системой команд; 4) доступность средств автоматизации программирования на основе алгоритмических языков, позволяющих создавать многофункциональные управляющие комплексы, развивать системы управления (главным образом расширяя программное обеспечение). Разработчики обычно подготовлены к программированию с использованием алгоритмических языков, поддерживаемых серийными инструментальными системами.

На основе однокристалльных микропроцессоров промышленность выпускает разнообразные серийные микроЭВМ, которые отличаются объемом основной памяти, внешним интерфейсом и номенклатурой плат расширения, набором внешних устройств и программным обеспечением. В тех случаях, когда технические характеристики серийных микроЭВМ удовлетворяют требованиям, предъявляемым к цифровым САУ, их применение предпочтительно по сравнению с разработкой и изготовлением собственных специализированных контроллеров. Конструктивное исполнение микроЭВМ может быть ориентировано на использование в виде встроенной системы управления (одноплатные микроЭВМ) или в виде автономной цифровой управляющей системы (многоплатные конфигурации).

Однокристалльный микропроцессор определяет систему команд, структуру, принципы организации памяти и управления вводом—выводом, внутренний интерфейс микроЭВМ.

Основными достоинствами использования серийных микроЭВМ являются существенное снижение трудоемкости проектирования, изготовления и отладки цифровых систем управления, что свя-

Таблица 5.4

**Характеристики
специализированных БИС умножителей**

Тип умножителя	Время умножения 16×16 , нс	Технология БИС	Статическая рассеиваемая мощность, Вт
588BP2	300	КМОП	10^{-6}
1800	17	ЭСЛ	10
1802	170	ТТЛШ	6

зано с унификацией технических средств и программного обеспечения. Серийные микроЭВМ изготавливаются на автоматизированных технологических линиях, что обуславливает их высокую надежность.

Недостатком применения серийных микроЭВМ как унифицированных изделий является избыточность оборудования, необходимость разработки дополнительных УСО для связи объекта со стандартным интерфейсом микроЭВМ.

Проектирование цифровых САУ на серийных микроЭВМ включает выбор типа микроЭВМ, комплектацию ее платами из серийного набора, проектирование специализированных УСО для сопряжения объекта управления с унифицированными интерфейсами микроЭВМ, программирование алгоритмов управления. При выборе микроЭВМ учитываются разрядность, быстродействие, система команд, объем адресуемой памяти, внешний интерфейс, внутреннее (резидентное) и внешнее ПО, наличие инструментальных систем автоматизации программирования и отладки.

Специальные функции в микроЭВМ и микроконтроллерах, выполненных на основе однокристальных микропроцессоров, могут быть реализованы программно, включая операции повышенной точности с фиксированной и с плавающей точкой. Программная реализация вносит существенные задержки и может оказаться критической при формировании управляющих воздействий. Для быстрой вычислений применяются *арифметические расширители*, которые подключаются к микроЭВМ как специальные внешние устройства или *сопроцессоры*. Например, могут быть включены (как расширители) быстрые умножители для вычисления 16×16 -разрядного произведения.

Параметры быстродействующих серийных 16 -разрядных умножителей (16×16) приведены в табл. 5.4 (16 -разрядные вычисления в большинстве случаев удовлетворяют требованиям точности в цифровых САУ). Время выполнения основных арифметических операций типа регистр—регистр с 16 -разрядными данными для трех типов серийных микроЭВМ приведено в табл. 5.5. В арифметическом расширителе МСУВТ В7 микропрограммно реализованы операции умножения и деления 16 - и 32 -разрядных чисел со знаком и без знака, а также арифметика с плавающей точкой в 32 -разрядном формате. В таблице 5.6 приведены приближенные параметры программной и аппаратной реализации операций с плавающей точкой в 32 -разрядном формате. Аппаратно операции выполняются на порядок быстрее, чем в случае программной

Таблица 5.5

Время выполнения арифметических операций с 16-разрядными двоичными числами в серийных микроЭВМ, мкс

Операция	Тип микроЭВМ (микропроцессора)			
	«Электроника-60» (МП К581)	МСУВТ В7		СМ-1810 (МП КР1810ВМ86)
		программа для МП КР580ИК80	команда в расширителе (МПК К589)	
+	4	15	2	1
*	80	300	40	20
/	100	400	60	30

реализации этих операций. Модуль арифметического сопроцессора, реализующего операции с 32-разрядными числами, арифметику с плавающей точкой, тригонометрические и экспоненциальные функции входит в микропроцессорный комплект К1810.

С учетом приведенных табличных данных для каждой серийной микроЭВМ можно рассчитать среднее время выполнения алгоритма управления (t_p), если рассчитано среднее число основных операций, выполняемых при вычислениях,

$$t_p \approx \bar{n}_c \tau_c + \bar{n}_y \tau_y + \bar{n}_d \tau_d,$$

где \bar{n}_c — число сложений; τ_c — время выполнения сложения; \bar{n}_y — среднее число операций умножения; \bar{n}_d — среднее число операций деления.

Если m — число ограничивающих условий, проверяемых в планирующей программе, то время ее выполнения определяется как $t_{\text{пл}} \approx m(\tau_c + \tau_v)$, где τ_v — время выполнения команды ветвления.

Кодирование команд оказывает влияние на объем требуемой памяти и время выполнения программ. Если z — длина слова ОП,

Таблица 5.6

Время выполнения арифметических операций с плавающей запятой в 32-разрядных форматах в серийных микроЭВМ, мс

Операция	Тип управляющей микроЭВМ (микропроцессора)					
	«Электроника-60»		МСУВТ В7		СМ-1810	
	программа МП К581	команда расширителя	программа МП КР580	команда расширителя	программа МП КР1810ВМ86	команда сопроцессора КР1810ВМ87
+	0,15	0,08	0,8	0,1	0,18	0,014
*	0,6	0,15	1,5	0,15	0,3	0,02
/	1,2	0,3	3,6	0,2	0,5	0,04

совпадающая, как правило, с разрядностью микропроцессора, l_i — длина команд, то необходимый объем ПО равен $\sum_i n_i l_i / z$, а среднее время выборки команд — $t \sum_i p_i l_i / z$, где p_i — вероятность обращения к команде с форматом l_i ; n_i — число i -х команд в программе.

Таким образом, с увеличением длины слова ОП или сокращением длины команды сокращается время выполнения программы. Эффективная с точки зрения кодирования система команд отличается большим разнообразием форматов и преимущественным использованием коротких однобайтовых команд. Из рассмотренных выше микропроцессоров наиболее эффективной с этой точки зрения является система команд МП К1810ВМ86.

Возможности серийных микроЭВМ зависят также от комплекса плат расширения. Платы имеют унифицированный для конкретного типа микроЭВМ системный интерфейс и совместимы конструктивно. Их можно разделить на несколько типов по назначению.

1. Плата процессора или одноплатная микроЭВМ является центральной в комплекте, включает унифицированный или специальный микропроцессор (МП) и адаптер системной шины. На плате размещаются контроллеры прерывания (ПР), прямого доступа к памяти (ПДП), таймеры (Т), ОЗУ, ПЗУ, ИРПР и ИРПС-адаптеры.

2. Плата памяти и ввода—вывода включает ОЗУ, ППЗУ, СППЗУ. Кроме того на плате могут располагаться таймеры и интерфейсы ввода—вывода ИРПР и ИРПС.

3. Платы специального ввода—вывода имеют разнообразные устройства связи с объектом, числоимпульсные преобразователи (ЧИА), АЦП, ЦАП, компараторы (КМП), усилители (У), выходы типа «сухой контакт» (СХК), ИРПР с оптической развязкой (ОР), ИРПС.

4. Платы стандартного ввода—вывода обеспечивают связь с внешними устройствами: фотосчитывателем (ФС), перфоратором (ПЛ), накопителем на магнитном диске (НГМД), кассетным накопителем на магнитной ленте (КНМЛ), алфавитно-цифровым дисплеем (АЦД), графическим дисплеем (ГД), алфавитно-цифровой печатью (АЦПУ).

Выбор комплекта плат с учетом требований к ресурсам цифровой САУ может быть представлен в общем виде задачей целочисленного линейного программирования.

Требования к ресурсам рассматриваются на начальных этапах проектирования цифровой САУ, при выборе структуры САУ и расчете числа преобразователей информации, измерительных устройств, их разрядности и выборе принципов подключения. В дальнейшем на этапах синтеза алгоритмов управления, проектирования УСО и реализации алгоритмов требования уточняются.

Таблица 5.7

Распределение ресурсов в платах расширения
управляющей микроЭВМ МСУВТ В7

Тип платы	МП	ОЗУ, Кбайт	СППЗУ, Кбайт	ПДП вход	ИРПС канал	Таймер	ПР вход	ИРПР канал	АЦП вход	ЦАП	У
ПМВ01П	1	16	16	—	4	6	15	—	—	—	—
МВ01П	—	16	16	—	4	3	8	6	—	—	—
АВВ01П	—	—	—	—	—	—	—	—	32	2	—
ЦВВ01П	—	—	—	—	—	—	4	—	—	—	48
ПМВ03П	1	16	8	4	4	3	8	—	—	—	—

Пусть для реализации цифровой системы управления необходимо: ОЗУ емкостью b_1 , СППЗУ емкостью b_2 , число последовательных каналов b_3 , число таймеров b_4 , число сигналов прерываний b_5 , число линий цифрового ввода—вывода b_6 . Существует набор стандартных плат $i = \overline{1, m}$, каждая из которых покрывает некоторое число a_{ij} ресурса b_j и оценивается стоимостью C_i . Распределение ресурсов в платах расширения серийной управляющей микроЭВМ МСУВТ В7 приведено в табл. 5.7. Необходимо выбрать минимальный по стоимости C набор плат $\{x_i\}$, где $C = \sum_i C_i x_i$ при ограничениях вида $\sum_j a_{ij} x_i \geq b_j$, $j = \overline{1, m}$ (x_i — целое положительное число плат i -го типа).

Часто используются платы, содержащие однотипный ресурс, т. е. $a_{ij} \neq 0$ только для одного значения $j \in \{1, 2, \dots, m\}$. В этом случае задача упрощается и решается последовательно покрытием требуемых ресурсов.

Номенклатура серийных микроЭВМ постоянно обновляется [48, 55, 94]. В настоящее время в САУ используются управляющие микроЭВМ «Электроника С5-21» (200 тыс. оп./с) и МСУВТ В7 (МПК 580); микроЭВМ общего назначения «Электроника-60», СМ-1800 (МПК 580), СМ-1810 (МПК 1810). Одноплатная микроЭВМ МС-1201 (МПК 1801) является одной из перспективных в САУ благодаря значительным внутренним ресурсам (ОЗУ 28К × 16 бит, ИРПС, ИРПР, интерфейс НГМД, выход на печать, 4К × 16 бит ПЗУ с системным монитором).

Кроме перечисленных, промышленность выпускает ряд отраслевых, специализированных управляющих микроЭВМ: МС 11100 (МПК 1801), КТС ЛИУС (МПК 580), РЕМИКОНТ (МПК 580) и др.

Серийные микроЭВМ обеспечиваются системными программами, библиотеками стандартных программ, пакетами прикладных программ и инструментальными системами. Совместимость серийных микроЭВМ по системе команд с наиболее распространенными, имеющими большой задел программного обеспечения

ЭВМ (в данном случае СМ-1420, СМ-1800), является одним из условий их широкого распространения, быстрого освоения и внедрения.

5.3. Стандартные внешние интерфейсы серийных микроЭВМ

Структура цифровой САУ включает одноплатную микроЭВМ, дополнительную плату памяти ОП₀ (расширение резидентной памяти), плату адаптера внешнего интерфейса для связи с УСО (рис. 5.4).

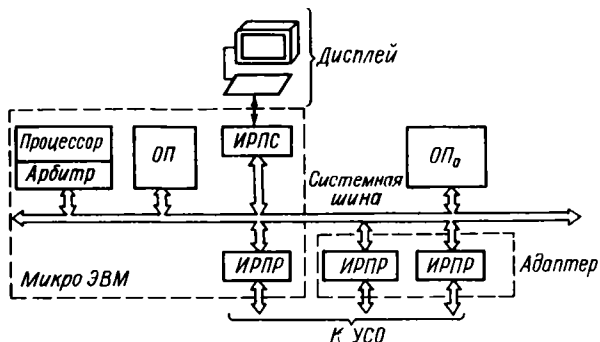


Рис. 5.4. Структура серийной микроЭВМ в цифровой САУ

В серийных управляющих микроЭВМ используются три типа внешних стандартных интерфейсов: системный, параллельный ИРПР и последовательный ИРПС.

Структура, принцип работы и организация интерфейсов в целом определяют принципы организации технических средств в цифровой САУ на основе серийных микроЭВМ.

Интерфейс радиальный параллельный ориентирован на параллельную передачу байта данных. К нему непосредственно подключаются цифровые измерительные и исполнительные устройства объекта управления, некоторые стандартные внешние устройства (например, печатающее и фотосчитывающее устройства) и специальный пульт в цифровой САУ. ИРПР (ОСТ 25778—77) содержит 8-разрядные шины ввода и вывода (возможно включение дополнительного контрольного разряда), линии сопровождения (готовность и строб источника) и линии ответа (запрос приема и готовность приемника). БИС программируемого параллельного интерфейса КР580 ВВ55 реализует три режима обмена байтами в ИРПР:

- 1) синхронную однонаправленную передачу байта данных по команде микропроцессора;
- 2) асинхронную однонаправленную передачу байта по готовности внешнего устройства;
- 3) асинхронный двунаправленный обмен байтами.

Схема имеет, с одной стороны, интерфейс, согласованный с внутренним интерфейсом микроЭВМ, а с другой — программируемые цифровые входы—выходы [3, 72] и включает регистр управления Y , в котором устанавливается управляющее слово, и 8-разрядные регистры данных A , B , C , связанные с двунаправленными программируемыми входами—выходами (цифровыми каналами). Управляющее слово в регистре Y задает режим работы и направление передачи по каналу. Через канал C осуществляется только синхронная передача байта или полубайта. В других режимах регистр C обеспечивает управление асинхронными передачами через каналы B и A .

По каналу B могут быть выполнены обе элементарные передачи байта — синхронная и асинхронная. Асинхронная передача сопровождается сигналом начала передачи СТРОБ и ответным сигналом ГОТОВНОСТЬ, который фиксируется в регистре C . В этом случае ГОТОВНОСТЬ можно проверить, опрашивая состояние регистра C и анализируя бит ГОТОВНОСТИ в программном цикле ожидания (длительность цикла 15 мкс для МПК 580). Кроме того, признак готовности, формируемый в УСО, можно подключить к линии *ready* или *int* внутреннего интерфейса микроЭВМ. При этом значение признака анализируется аппаратом микропроцессором в цикле ожидания соответствующей команды ввода—вывода или вызывает прерывание (задержка ввода—вывода 0,5 мкс для МПК 580).

Через канал A , кроме элементарных однонаправленных передач, может быть выполнена двунаправленная передача, сопровождаемая сигналами, определяющими направление передачи, и соответствующая асинхронному обмену с ожиданием. Медленные процессы ввода—вывода в каналах цифровых систем управления могут быть совмещены с параллельными вычислениями в микроЭВМ, если разрешение передачи сигнализируется прерыванием.

Интерфейс радиальный последовательный обеспечивает последовательный побитовый обмен данными с дистанционными датчиками и исполнительными устройствами, включает линии передачи и приема данных типа токовая петля, а также линию сигнала готовности приемника; скорость передачи данных 9600 бит/с, максимальное расстояние — до 500 м.

Из последовательности битов необходимо выделить коды символов. Для этой цели в начале кода передается стартовый бит (уровень нуля), а в конце кода — стоповый бит (уровень единицы).

В комплект БИС микропроцессорного набора КР580 входит универсальный синхронно-асинхронный передатчик КР580ВВ51. БИС подключается к внутреннему интерфейсу микроЭВМ и обеспечивает преобразование параллельного кода в последовательность битов при выводе и формирование кода при вводе по ИРПС каналу. Управляющее слово задает режимы обмена: ввод или вывод, асинхронный обмен с контролем и формированием стартстоповых битов или синхронный с автоматической передачей

синхрoслова. Программируется скорость приема—передачи последовательных битов, число битов в последовательности, контроль по четности. Окончание ввода—вывода кода символа данных фиксируется по готовности.

В структуре микроЭВМ (рис. 5.5) выделяют *внутренний интерфейс (микрошина) M_0* , включающий шины адреса, информационную и управления. Шины внутреннего интерфейса являются собственными шинами микропроцессора, и расширение осуществляется подключением к ним дополнительных элементов набора КР580, КР1810, которые позволяют, в частности, реализовать внешние интерфейсы типа ИРПР или ИРПС

Кроме внутренней шины M_0 , может быть организована системная *общая шина M_1* и *мультишина M_2* . Для управления этими интерфейсами используются модули системных арбитров общей шины (АОШ) и мультишины (АМШ).

Стандартные системные интерфейсы обеспечивают следующие режимы управления: 1) обращение к основной памяти (чтение и запись по адресу); 2) обращение системному вводу—выводу (ввод или вывод по адресу); 3) вход в общесистемное прерывание; 4) арбитраж при обращении к системным устройствам ввода—вывода или ОП.

Первые две операции являются элементарными и обеспечивают простые режимы обращения к ОП и устройствам ввода — вывода, которые являются пассивными устройствами по отношению к активному процессору. Вход в прерывание включает две элементарные операции в системном интерфейсе: разрешение прерывания по запросу и передачу *вектора прерывания* для перехода к программе обработки прерывания.

Захват шины и арбитраж при обращении к шине со стороны нескольких устройств является основной операцией, которая отличает системную шину от других внешних интерфейсов серийной микроЭВМ. При выполнении этой операции одно из устройств становится активным и ему предоставляется быстрый доступ (прямой) к пассивным системным устройствам, в том числе и ОП. Расширение возможностей цифровой системы управления, в частности, включение параллельных процессоров в мультипроцессорной системе осуществляется также через системную шину.

Магистральный параллельный интерфейс (МПИ) (ОСТ 11305.903—80) используется в микроЭВМ «Электроника-60» и применяется в САУ с одним центральным процессором, что соответствует организации системы на основе общей шины. МПИ

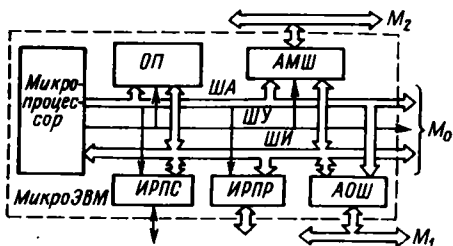


Рис. 5.5. Структура микроЭВМ с тремя системными интерфейсами

включает 16-разрядную совмещенную мультиплексную шину адреса и данных. Передача данных или адреса через шину сопровождается признаками, формируемыми на линиях СИА (признак адреса), ВЫВОД (признак данных, передаваемых в шину активным устройством), ВВОД (запрос данных активным устройством при вводе) и СИП (завершение обмена, формируемое пассивным устройством P). Эти сигналы сопровождают простой обмен дан-

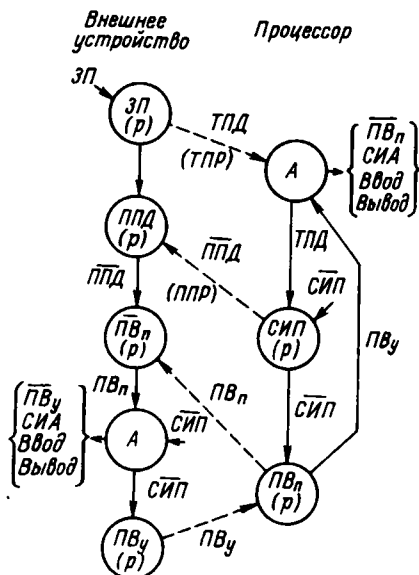


Рис. 5.6. Диаграмма состояний интерфейсных схем при переключении управления общей шиной МПИ

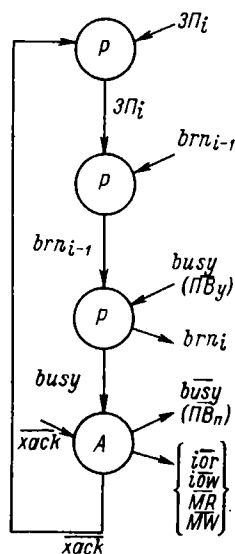


Рис. 5.7. Диаграмма состояний арбитра состояния мультишины

ными через шину по «инициативе» устройства, управляющего шиной, которое считается *активным*.

Общая схема взаимодействия системных устройств через интерфейс при передаче управления шиной и обменах приведена на рис. 5.6.

Передача управления шиной новому активному устройству осуществляется по запросу (линия запроса ТПД). Арбитр шины, анализирующий состояние линии запроса, размещается в центральном процессоре. В ответ на сигнал запроса арбитр формирует сигнал разрешения по линии разрешения ППД. Все устройства подключаются параллельно к линии запроса ТПД и последовательно к линии разрешения ППД. При одновременных запросах со стороны нескольких устройств к шине подключается только одно из них, причем только такое устройство, которое первым воспринимает сигнал разрешения по линии ППД. Таким образом, чем ближе устройство подключено по последовательной

линии ППД к центральному процессору, тем выше его приоритет, который в данном случае называют позиционным.

Признак занятия шины (низкий уровень на линии ПВ) устанавливается активным устройством. Состояния системных устройств могут быть пассивными (*P*) и активными (*A*) и соответствовать задержкам, связанным с ожиданием или формированием сигналов, передаваемым по линиям управления. После захвата шины активное устройство осуществляет элементарную операцию обмена с системным устройством, формирует адрес устройства, сигналы СИА, ВВОД, ВЫВОД.

Приведенная на рис. 5.6 схема взаимодействия является приближенной моделью соответствующих процессов, связанных с использованием системного интерфейса. Полное определение интерфейса, кроме формальных схем и описания работы, включает временные диаграммы для каждого типа обмена и переключения состояния шины. Стандартом устанавливаются допустимые задержки между управляющими и ответными сигналами, причем отсутствие ответного сигнала СИП в течение 10 мкс, вызывает прерывание по неисправности в шине.

Мультимашинный интерфейс (мультишина или И-41 ОСТ 25968—82) используется преимущественно в микроЭВМ, выполненных на наборах КР580, КР1810. Позиционно распределенная схема арбитра позволяет упростить управление обменами и переключение шины, интерфейс включает отдельные 16-разрядные ША и ШИ. Благодаря этому операции обмена выполняются быстрее, чем в мультиплексной шине МПИ. Управляющие сигналы формируются активным устройством и определяют режим работы пассивного устройства. К ним относятся запись ОЗУ *MW*, чтение ОП *MR*, ввода *ior* и вывода *iow*.

Переключение шины осуществляется асинхронно локальными арбитрами. Схема работы локального арбитра в *i*-м устройстве, которое становится активным, приведена на рис. 5.7. Сигнал запроса шины *brn* формируется каждым устройством и передается по схеме позиционного приоритета на вход арбитра соседнего, менее приоритетного устройства, либо в центральную схему анализа приоритета. При позиционном последовательном соединении арбитров более приоритетный арбитр запрещает формирование запросов во всех последующих менее приоритетных арбитрах. В схеме централизованного анализа приоритета можно выбрать другой способ обслуживания запросов (например, циклическое изменение приоритетов или программируемое).

Схема арбитра наиболее приоритетного устройства готова перейти к управлению шиной, но остается пассивной, пока шина не освободится. Соответствующий сигнал *busy* эквивалентен сигналу $\overline{ПВ}$ на рис. 5.6, сигнал окончания обмена *hack* совпадает по смыслу с сигналом СИП. После освобождения шины (высокий уровень сигнала *busy*) устройство становится активным (*A*) и

управляет обменом с пассивным устройством, формируя управляющие сигналы, адрес и данные.

Стандартные внешние интерфейсы позволяют комплексовать серийные микроЭВМ серийными внешними устройствами, платами расширения и обеспечивают физическое сопряжение микроЭВМ с цифровыми входами—выходами объекта управления. Физическое сопряжение с объектом управления поддерживается аппаратными и программными средствами управления обменом данными в реальном времени.

5.4. Сопряжение микроЭВМ с объектом управления в реальном времени

Все процессы управления развиваются в реальном времени, которое связано с динамическими свойствами объекта и учитывается при выборе параметров цифровой САУ. Важнейшим из этих параметров является интервал квантования во времени T , который определяет допустимое запаздывание τ в формировании управления $\tau \leq T$.

Интервал квантования учитывается на этапах синтеза алгоритмов управления при выборе микропроцессорного набора или серийной микроЭВМ и программирования алгоритмов управления. Таким образом обеспечиваются требования формирования управления в реальном времени, определяемом динамическими свойствами ОУ. При сопряжении микроЭВМ с объектом управления требуются специальные технические средства, обеспечивающие согласование процессов формирования управления при исполнении программ в микроЭВМ с изменением состояния ОУ во времени на физическом уровне с учетом ограничения $\tau \leq T$. Поскольку сопряжение осуществляется на уровне процессов ввода—вывода, то характеристики системы ввода—вывода и ее организация являются важнейшими при проектировании технических средств цифровой САУ.

Особенностями всех процессов, происходящих в ОУ и связанных с изменением состояния во времени, являются асинхронность и параллельность по отношению к процессам вычислений в управляющей микроЭВМ.

Параллельное, неконтролируемое изменение состояния ОУ в интервале квантования T может вызывать случайные события — прерывания, требующие принятия срочного решения с задержкой, существенно меньшей T . Вместе с тем, инерционность отдельных процессов изменения состояния ОУ при параллельном развитии процессов может быть использована для выполнения параллельных вычислений и формирования управляющих воздействий для других процессов. Окончание инерционных процессов может быть проконтролировано: 1) запуском таймера, который вызывает прерывание после окончания отсчета заданного интервала времени; 2) прерыванием, формируемым независимо в ОУ.

Процессы в ОУ могут быть завершены при достижении заданной скорости перемещения или положения в пространстве. Такие состояния ОУ являются наиболее вероятными, легко идентифицируются сигналами с порогового датчика и концевого переключателя. При этом после запуска таймера прерывание требуется только в случае отсутствия сигнала с датчика после окончания заданного интервала.

Операция ввода—вывода в ряде случаев также является самостоятельным асинхронным процессом, в котором скорость выполнения команд в микроЭВМ существенно выше скорости формирования, приема или выдачи данных в ОУ или УСО.

Асинхронный обмен начинается по команде процессора и связан с ожиданием сигнала готовности ОУ, который либо контролируется в команде, либо вызывает прерывание. Таким образом, для сопряжения микроЭВМ с объектом управления в реальном времени необходимы следующие технические средства: система прерывания, средства асинхронного ввода—вывода, таймеры.

Таймер представляет собой счетчик времени, накапливающий или вычитающий. Накапливающий счетчик фиксирует абсолютное или относительное время ($T = \Delta t N$), пропорциональное интервалу отсчета Δt и значению отсчета N . Чаще используется вычитающий счетчик, в котором при запуске устанавливается отсчет N , и при переходе его в нулевое состояние формируется сигнал окончания интервала.

Счетчик таймера может быть организован программно. При этом подсчет времени и проверка окончания счета осуществляются командами. Дискретность (точность) отсчета времени характеризуется длительностью цикла программы, который может включать несколько команд. Недостатком программного таймера является ограниченная точность отсчета времени (10—50 мкс).

Аппаратный счетчик-таймер отсчитывает время параллельно с исполнением программ. На вход счетчика подается синхросерия с частотой до 1 МГц и выше, что позволяет выполнять отсчет времени с точностью 1 мкс. Отсчет относительно длительных интервалов времени осуществляется параллельно с вычислениями текущего управления в цифровой САУ (окончание отсчета вызывает прерывание).

Программируемый таймер в БИС КР580ВИ53 [3] может работать в нескольких режимах. Отсчет интервалов может начинаться с момента занесения константы времени или по сигналу разрешения, формируемому в УСО. Отсчет может выполняться периодически с заданным интервалом; возможно перепрограммирование интервала во время отсчета.

В цифровой системе управления можно использовать несколько таймеров для синхронизации нескольких параллельных процессов. При относительно медленных процессах управления в операционных системах реального времени (гл. 6) поддерживаются несколько программных счетчиков. Значения счетчиков

изменяются «одновременно» (насколько это можно осуществить программным опросом) по временной метке, которой соответствует сигнал прерывания с системного таймера (синхронизатора). Точность отсчета в микроЭВМ СМ-1800 — 0,05 с, а в микроЭВМ «Электроника-60» — 0,02 с. Интервал отсчета соответственно можно приближенно определить исходя из того что, как правило, используются 16-разрядные счетчики, т. е. для СМ-1800 он находится в пределах 0,05 ... 1 ч, для «Электроники-60» — от 0,02 с до 0,3 ч.

Система прерывания обеспечивает быструю реакцию на многочисленные события, происходящие в реальном времени в цифровом регуляторе, во внешних устройствах микроЭВМ и объекте управления. Соответственно различают *внутренние* и *внешние события*, вызывающие прерывания. Внутренние события могут быть связаны с использованием программы (обнаружение резервных кодов команды в программе, неправильная адресация в некоторых типах команд микроЭВМ «Электроника-60»). Такие прерывания относятся к процессорным. *Внепроцессорные прерывания* связаны с событиями, которые регистрируются специальными схемами (неисправностями питания и шины МПИ при отсутствии ответного сигнала СИП, прерыванием от таймера, регенерацией динамического ОЗУ).

Синхронизация работы цифрового регулятора с событиями реального времени в объекте управления осуществляется сигналами внешних прерываний, которые формируются схемами УСО и аппаратурой внешних устройств.

Условия процессорных прерываний в микроЭВМ «Электроника-60» обнаруживаются микропрограммой при выполнении соответствующей команды, которая завершается переходом к подпрограмме по косвенному адресу.

Адрес начала подпрограммы принятия решения по прерыванию выбирается из фиксированной ячейки ОП и устанавливается в СЧАК (текущее состояние СЧАК записывается в стек).

Внепроцессорные прерывания в микроЭВМ «Электроника-60» вызывают микропрограмму косвенного обращения к подпрограмме с использованием механизма подмены адреса микрокоманды, рассмотренного в п. 5.1. Внешние прерывания регистрируются микропроцессором в конце цикла текущей команды проверкой входа запроса прерывания. Управление внешними прерываниями осуществляется через системный интерфейс.

В интерфейсе МПИ последовательность операций в шине совпадает при прерываниях с соответствующей последовательностью операций при захвате шины (см. рис. 5.6). Запросы прерывания объединяются в одну линию ТПР и поступают на вход запроса ТПР процессора, который формирует сигнал разрешения по линии ППР. Устройства, запрашивающие прерывания, на линии ППР соединены последовательно в соответствии с позиционным приоритетом. После того, как текущий обмен в шине завер-

шается (признак СИП), процессор освобождает шину (признак ПВ_д) и становится пассивным устройством, ожидающим косвенный адрес-вектор от внешних устройств. Адрес с соответствующими сопровождающими сигналами устанавливается наиболее приоритетным устройством, запрашивающим прерывание, которое и становится активным в этом обмене. По окончании обмена (признак СИП от процессора) устройство сбрасывает признак занятости шины (признак ПВ_д) и становится пассивным, а процессор — снова активным и завершает выполнение микропрограммы обращения к подпрограмме по принятому косвенному адресу-вектору от внешнего устройства.

Таким образом, каждое устройство вызывает собственную подпрограмму ввода—вывода или принятия решения по прерыванию, передавая через шину свой адрес-вектор. Подпрограмма ввода—вывода, вызванная по прерыванию, завершается, например, командой ВОЗВРАТ ИЗ ПЕРЕРЫВАНИЯ. При этом из стека читается в СЧАК адрес возврата в основную программу (программа продолжается с команды, перед которой зафиксировано прерывание). При обращении к подпрограмме по прерыванию в стек записывается СЧАК, состояние регистра признаков результата и бит приоритета, которые объединяются в *слово состояния программы* (ССП).

Приоритеты, закрепляемые аппаратно за каждым событием в системе с МПИ, учитываются как в случае одновременного возникновения соответствующих запросов, так и в случае последовательных запросов. В последнем случае возможны вложенные (многоуровневые) прерывания, при которых следующее событие прерывает программу обработки предыдущего события. Вложенные прерывания в МПИ возможны сразу после перехода к очередной программе обработки прерывания.

В серийных микроЭВМ, выполненных на основе микропроцессорного набора КР580 с системной шиной И-41, также различают внутренние и внешние прерывания. Внешние по отношению к микропроцессору сигналы прерывания можно разделить на локальные, связанные только с конкретным микропроцессором или одноплатным микроконтроллером, и системные, формируемые на линиях системной шины.

Управление внешними прерываниями осуществляется при взаимодействии микропроцессора и локальной программируемой БИС контроллера прерывания КР580ВН59 [3, 7]. К одной БИС подключаются до восьми сигналов прерывания. Возможно параллельное включение m БИС, что позволяет использовать $8 \times m$ сигналов. Каждому входному сигналу прерывания соответствует определенный приоритет, который может изменяться программно.

Диаграмма взаимодействия микропроцессора и контроллера прерываний приведена на рис. 5.8. По сигналу прерывания с внешнего устройства ЗП ПР контроллер формирует запрос прерыва-

ния *INT* к микропроцессору, который отвечает сигналом подтверждения *INTA*. По этому сигналу контроллер формирует на информационной шине команду обращения к подпрограмме *CALL* (адрес); адрес модифицируется кодом причины прерывания.

Запрос прерывания анализируется в конце (для КР580) или в начале (для КР1810) цикла микропрограммы исполнения команды. При обнаружении запроса (*INT*) осуществляется чтение команды обращения к подпрограмме, формируемой контроллером прерывания.

В системной шине И-41 выделено восемь линий для системных сигналов прерываний, которые подключаются к локальным контроллерам.

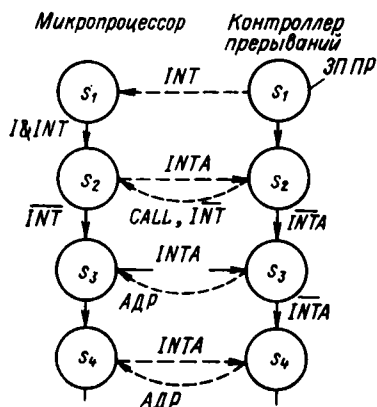


Рис. 5.8. Диаграмма состояний микропроцессора и контроллера при входе в прерывание

Сигнал прерывания в системной линии может вызвать одновременно прерывание в нескольких параллельно работающих устройствах, подключенных к шине.

В начале выполнения подпрограммы обработки прерывания возможно новое прерывание от более приоритетного внешнего события. Анализ и сравнение приоритетов осуществляются автоматически локальным контроллером прерываний, который хранит код текущего прерывания.

Специальный однобитовый регистр (флаг) разрешения прерываний *I* в микропроцессоре устанавливается программно, а сбрасывается автоматически после приема запроса прерывания.

Прикладная программа, вызываемая по этому запросу, может задержать повторные прерывания во время своего исполнения, но по окончании разрешает прерывания.

Асинхронный ввод—вывод предусмотрен в стандартных интерфейсах микроЭВМ (см. п. 5.3): в ИРПР, ИРПС и системных интерфейсах формируется сигнал готовности, который опрашивается командой микропроцессора. В системных интерфейсах после захвата шины выполняется прямой обмен данными между активным устройством и пассивным.

Асинхронные чтение или запись данных в ОЗУ микроЭВМ активным внешним устройством называется *прямым доступом к памяти* (ПДП). Для организации ПДП могут быть использованы специальные контроллеры.

Комплект БИС КР580 включает программируемый контроллер ПДП к основной памяти КР580ВТ57, который управляет прямой передачей данных за один цикл обращения к ОП с приостановкой процессора на время обращения [3].

Контроллер содержит четыре однотипных канала ПДП, между которыми установлен жесткий приоритет для выбора канала в случае одновременного запроса от нескольких УСО. Каждый канал содержит регистр адреса ОП и счетчик данных (СЧД),

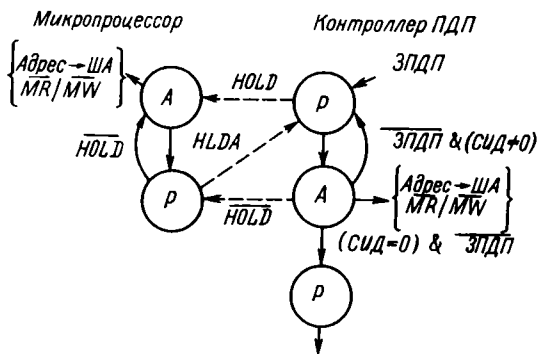


Рис. 5.9. Диаграмма состояний микропроцессора и контроллера ПДП

которые устанавливаются процессором при программировании контроллера ПДП.

Диаграмма взаимодействия системных блоков в ПДП представлена на рис. 5.9, схема подключения контроллера к микрошине — на рис. 5.10.

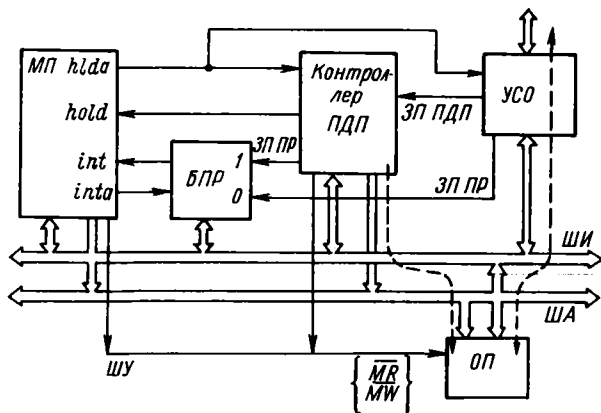


Рис. 5.10. Схема включения контроллеров прерываний и прямого доступа в микроЭВМ

Программирование контроллера ПДП выполняется в обычном порядке: микропроцессор записывает управляющие слова, передает адрес ОП и число данных в счетчик данных (СЧД), обращаясь к контроллеру ПДП как к внешнему устройству. В дальнейшем контроллер готов к анализу запросов ПДП от УСО. По готовности

измерительного или исполнительного устройства в УСО формируется сигнал запроса ПДП (ЗП ПДП), который через контроллер ПДП трансформируется в сигнал запроса приостановки процессора *HOLD*. Процессор отключается от шин (переводит соответствующие выходы в высокоимпедансное состояние) и формирует ответный сигнал *HLDA*, разрешающий передачу данных в режиме ПДП.

Контроллер при этом формирует адрес на ША, сигналы чтения \overline{MR} и записи \overline{MW} в ОЗУ, модифицирует счетчик данных; УСО реализует передачу данных, подключаясь к ШИ. Цикл ПДП завершается сбросом сигналов *HOLD*, *HLDA*, после чего процессор снова подключается к шинам, а УСО и контроллер ПДП отключаются. Таким образом, ПДП обеспечивает быструю передачу данных за один цикл обращения к ОЗУ.

Окончание групповой передачи данных, например, связанной с последовательным опросом аналоговых сигналов на мультиплексоре АЦП, может быть выполнено по прерыванию. Сигнал прерывания ЗП ПР контроллер ПДП вырабатывает автоматически при сбросе счетчика данных в цикле ПДП при СЧД = 0 (см. рис. 5.9). Этот сигнал поступает в БПР (рис. 5.10) и вызывает операцию прерывания в микропроцессоре (см. рис. 5.8).

5.5. Устройства сопряжения с объектом

Устройства обеспечивают физическое согласование внешнего интерфейса микроЭВМ с входами—выходами объекта управления. Входы объекта управления связаны с разнообразными *исполнительными устройствами*. Управляющее воздействие может быть представлено уровнем напряжения, величиной тока, частотой и другими типами сигналов, физические параметры которых пропорциональны цифровому коду, формируемому в микроЭВМ. Преобразование цифрового кода в управляющее воздействие осуществляется УСО. Наиболее распространенными являются преобразователь типа двоичный код — напряжение (цифроаналоговый преобразователь), код — частота, код — электрический ток.

Выходы объекта управления также могут быть представлены разнообразными физическими процессами и явлениями в *измерительных устройствах*.

В цифровых системах, предназначенных для управления механической обработкой, ГАП и в роботизированных системах преобладают измерители, преобразующие линейное перемещение или угол поворота в напряжение, частоту или двоичный код. При управлении двигателями используются датчики скорости и ускорения, преобразующие соответствующие параметры в фазу переменного напряжения, уровень напряжения или частоту. Для ввода соответствующих значений с измерительных устройств в микроЭВМ используются разнообразные преобразователи типа напряжение—код (аналого-цифровой преобразователь), угол—

код, перемещение—код, частота—код, которые входят в состав УСО. Преобразователи могут быть: конструктивно объединены с измерительными устройствами и размещены в ОУ; выделены конструктивно в отдельные унифицированные платы в комплекте плат серийных микроЭВМ; размещены конструктивно в одной платной микроЭВМ или микроконтроллере.

Если в цифровой САУ возможно применение встроенных цифровых регуляторов, то измеряемые входные и управляющие выходные сигналы практически без искажений подключаются к соответствующим преобразователям в ОУ. Однако если используется автономная микроЭВМ, то необходимо обеспечить электрическую развязку и согласование с минимальной потерей точности. При передаче аналоговых сигналов эти условия не всегда можно выполнить.

Цифровые входы и выходы в САУ. Универсальным способом обмена информацией с ОУ является переход к цифровому кодированию и передача цифрового кода. Преобразователи конструктивно объединяются с измерительными и исполнительными устройствами в ОУ, а в УСО входят цифровые каналы со схемами электрической развязки и усиления, которые могут быть выполнены в виде плат расширения к серийной микроЭВМ (МСУВТ В7, КТС ЛИУС и др.) и подключены к ИРПР или ИРПС.

Цифровое кодирование естественно используется для передачи управления к стартстопным механизмам и устройствам, а также для передачи признаков состояния объекта управления.

К стартстопным устройствам относятся электромагнитные муфты и реле-пускатели. Аналоговым значением управляющего сигнала является выходная мощность; мощность более нуля кодируется единицей цифрового кода. Для управления малоомощными стартстопными устройствами могут быть использованы усилители на транзисторах; для передачи сигналов значительной мощности — реле, тиристоры и другие средства промышленной автоматики.

Основные типы реле: герконовые (мощность 10—20 Вт), ртутные герметичные (мощность до 100—150 Вт), сухие реле (до 1000 Вт).

Общая схема подключения цифровых выходов к ОУ приведена на рис. 5.11 и включает несколько каскадов преобразования, электрической развязки и усиления. Конкретные схемы согласования приведены в [61]. Признаки состояния объекта управления формируются: концевыми переключателями; пороговыми датчиками температуры, тока, напряжения; реле.

Цифроаналоговые каналы для связи с ОУ организуются с использованием ЦАП: код — напряжение, код—ток или код—частота (наиболее распространены преобразователи код—напряжение). Для многих применений доступны интегральные ЦАП, например, серии К572 ПА1 (10-разрядные) и ПА2 (12-разрядные), позволяющие формировать на выходе аналоговый сигнал в сим-

метричном или асимметричном диапазоне от 0 до 17 В с временем нарастания в полном диапазоне 5 мкс [100]. Расчет разрядности ЦАП с учетом погрешности управления приведен в п. 6.3.

Структура цифроаналогового канала приведена на рис. 5.9 и включает несколько каскадов преобразования и усиления. Оконечный каскад подключается к обмотке возбуждения, регулирующей скорость вращения двигателя; к тяговому реле, обеспечивающему требуемое механическое усилие; к нагревательному элементу и другим исполнительным устройствам в ОУ.

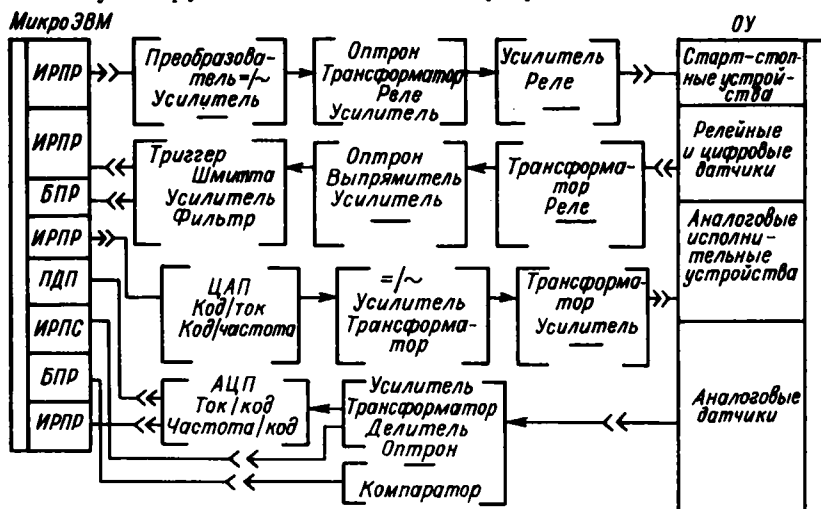


Рис. 5.11. Схема сопряжения микроЭВМ с объектом управления

Аналого-цифровые каналы для передачи сигналов от ОУ к микроЭВМ организуются с использованием АЦП: напряжение—код, ток—код, частота—код (см. рис. 5.9). Преобразователи первого типа являются наиболее распространенными. Параметры некоторых унифицированных интегральных АЦП приведены в табл. 5.8.

Кроме разнообразных измерителей температуры, давления, скорости, на выходе которых сигнал формируется в виде напряжения, в объекте управления устанавливаются электромеханические преобразователи типа угол—код, перемещение—код, скорость—код.

Код измеряемой величины из ОУ или УСО может быть передан в микроЭВМ через цепи электрической развязки на оптронах, ИРПР или последовательно через ИРПС.

Основные параметры АЦП: время преобразования, точность (разрядность выходных двоичного или двоично-десятичного кодов), диапазон входного напряжения. Разрядность АЦП зависит от погрешности управления ϵ_d и погрешности измерительного устройства Δu и рассчитывается в п. 6.3.

Параметры унифицированных и интегральных АЦП

Тип АЦП	Число входов	Разрядность, бит	Быстрота действия, мкс	Входное напряжение, В
572 ПВ1	1	12	110	5 ... 15
1113 ПВ1	1	10	20	5 ... 15
1107 ПВ1	1	6	0,05	-2 ... 0
СМ-1800	8	16	10^4	0 ... ± 5
С5-121	15	12	10^4	-10 ... +10
АВВ01П (МСУВТ)	16	10	120	$\pm 0,6$... ± 5
АВВ03П	16	10	70	$\pm 0,6$... ± 10
С5-2101	32	11	10^4	± 5

Время преобразования варьируется в широких пределах — от 0,1 мкс в параллельных интегральных АЦП до нескольких миллисекунд в высокоточных алгоритмических преобразователях (см. табл. 5.8).

Алгоритмическое аналого-цифровое преобразование реализует метод последовательного приближения и может быть выполнено в интегральном АЦП в виде микропрограммы или программы. При этом преобразование можно выполнить в ускоренном режиме слежения, совместить его с измерением скорости изменения сигнала и зафиксировать достижение порогового значения или переход сигнала через нуль.

Способ последовательного цифрового приближения к входному аналоговому сигналу $u_{вх}$ сводится к суммированию весов разрядов двоичного числа $0, x_1, x_2 \dots x_m$ при условии, что сумма $S = \sum_i 2^{-i} x_i$ остается меньше $u_{вх}$. Для сравнения выполняется обратное преобразование кода S с помощью ЦАП в аналоговый сигнал u_s . Сравнение $u_s > u_{вх}$ осуществляется на аналоговых компараторах типа К521, К597, которые на выходе Q формируют признак сравнения ($Q = 1$, если $u_s > u_{вх}$, иначе $Q = 0$). Чувствительность компаратора должна быть не ниже значения, соответствующего младшему разряду двоичного кода x_m . Последовательное приближение можно представить следующей рекуррентной формулой:

$$S_{i+1} = S_i + x_i 2^{-i},$$

где $x_i = 1$, если $S_i + 2^{-i} < u_{вх}$ и $x_i = 0$, если $S_i + 2^{-i} > u_{вх}$.

Микропрограмма 8-разрядного преобразования в микроконтроллере на МПК 1804 может быть выполнена за 5 мкс, а программа преобразования в микроЭВМ на основе КР580 — за 450 мкс. Для сокращения оборудования аналого-цифровое преобразование нескольких сигналов может быть выполнено одним АЦП с аналоговым коммутатором (мультиплексором). Существует несколько типов мультиплексоров, в которых коммутируе-

мый вход задается двоичным кодом — адресом (например, К590КН1 — мультиплексор на 8 входов, К590КН2 — на 4 входа [100]; (время коммутации 0,5 ... 1 мкс). Мультиплексор К591КН1 позволяет последовательно автоматически коммутировать 16 входов, время переключения 2,5 мкс.

При последовательной коммутации m аналоговых сигналов время преобразования увеличивается в m раз и равно $m(t_m + t_{\text{АЦП}})$, где t_m — время переключения мультиплексора и $t_{\text{АЦП}}$ — время одного преобразования.

Программа ввода данных, поступающих по аналого-цифровому каналу с мультиплексора сигналов, включает команды подсчета

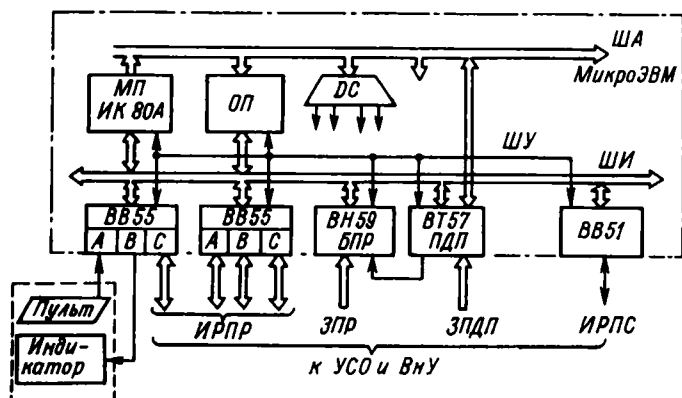


Рис. 5.12. Схема включения интерфейсных и системных БИС микропроцессорного набора КР580

количества данных, формирования адреса основной памяти, ввода данных и передачи их в ОП. Для МП КР580 выполнение такой программы превышает 25 мкс и вносит дополнительную задержку $t_b m$, где t_b — время передачи одного байта данных, m — число байт. Сократить эту задержку в программах цифрового управления позволяет организация прямого доступа к ОП в интерфейсе между УСО и микроЭВМ. При этом в аналого-цифровом канале необходимо обеспечить аппаратное переключение мультиплексора аналоговых сигналов, формирование адреса ОП и управление прямым обменом с ОП. Медленные процессы преобразования в канале могут быть совмещены по времени с выполнением программ в микроЭВМ.

Операция прямого доступа к памяти предусмотрена в системной шине микроЭВМ при активизации УСО (п. 5.3). В этом случае в состав оборудования УСО входит собственный арбитр шины, который совмещает функции контроллера ПДП. Система прерывания микроЭВМ обеспечивает быструю и автоматическую реакцию на критические и предельные значения сигналов, характеризующих состояние ОУ. Эти сигналы формируются концевыми

переключателями в ОУ или компараторами в УСО и поступают на входы запросов ЗПР (см. рис. 5.8) микроЭВМ.

Схема внешнего интерфейса и структура управляющей микроЭВМ (микроконтроллера), в которой используются программируемые БИС интерфейса, приведены на рис. 5.12.

МикроЭВМ включает МП, блок ОП, адресный декордер (DC). Линии внутреннего интерфейса разделяются на ША, ШИ и ШУ, которые включают линии чтения \overline{MR} , записи \overline{MW} , ввода IOR и вывода IOW. Декордер DC разрешает выборку ОП и интерфейсных БИС по адресу.

БИС цифровых входов—выходов КР580ВВ55 подключается к пульту управления, содержащему клавиатуру и светодиодные индикаторы. Часть цифровых входов—выходов используется для организации ИРПР интерфейса.

На входы КР580ВН59 подаются запросы прерываний (ЗПР), а на входы КР580ВТ57 — запросы прямого доступа (ЗПДП). БИС КР580ВВ51 используется для связи с ОУ по последовательному ИРПС интерфейсу. Количество линий внешнего интерфейса наращивается подключением дополнительных БИС.

На основании рассчитанных для цифровой САУ параметров внешнего интерфейса выбираются количество БИС и адресация элементов интерфейса, в том числе адресация ИРПР, ИРПС и ПДП каналов.

5.6. Организация запоминающих устройств

Характеристики и организация памяти существенно влияют на технические параметры цифровой системы управления и реализацию алгоритмов цифрового управления.

Память в цифровой САУ организуется на базе разнообразных запоминающих устройств (ЗУ), существенно различающихся по техническим параметрам, способу хранения и доступу к информации.

ЗУ могут быть выполнены в виде интегральной полупроводниковой памяти, на основе накопителей на магнитных дисках, на магнитной ленте, на цилиндрических магнитных доменах (ЦМД) и т. д.

Основными параметрами ЗУ являются стоимость хранения бита информации (отношение стоимости ЗУ к его емкости), время доступа к данным, объем памяти, потребляемая мощность, надежность, габаритные размеры.

Некоторые типы ЗУ имеют свойства, необходимые в условиях промышленного использования цифровой САУ. Одним из них является энергонезависимость, которая необходима для обеспечения высокой надежности работы системы при нестабильной питающей сети.

Энергозависимыми являются КМОП интегральные ЗУ, способные хранить информацию при пониженном питании и с минималь-

ным потреблением, а также все типы ЗУ на магнитных носителях.

Существенное различие в технических параметрах различных типов ЗУ обуславливает многоуровневую организацию памяти, характерную для всех классов ЭВМ. В микроЭВМ можно выделить четыре уровня памяти: быструю (регистровую); основную; промежуточную и вторичную.

Регистры общего назначения (РОН) являются одним из блоков микропроцессора, хранят оперативные данные. Время доступа минимальное.

Основная память разделяется на ОЗУ и ПЗУ, запоминающие устройства, выполненные на БИС памяти [100]. Время обращения к ОП обычно в несколько раз больше, чем к РОН, а объем ОП превышает объем РОН на 3÷5 порядков.

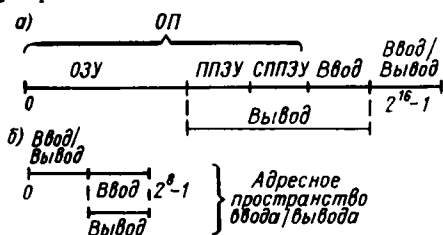


Рис. 5.13. Распределение адресного пространства: а — основного; б — ввода—вывода

Промежуточная память в микроЭВМ может быть выполнена на магнитных доменах, реализована в виде полупроводниковой памяти большого объема, организована в НГМД.

Вторичная память может быть выполнена на основе НМД и кассетного накопителя на магнитной ленте (КНМЛ).

Длина исполнительного адреса формируемого различными способами в командах микропроцессора (п. 5.2), определяет адресное пространство $0 \dots 2^n - 1$ адресов, где n — число разрядов в адресе.

Адресное пространство разделяется на несколько областей адресов, соответствующих различным по назначению и исполнению запоминающим устройствам. В частности, выделяется область оперативного запоминающего устройства для хранения оперативных данных и перезагружаемых программ (ОЗУ). Область ПЗУ или ППЗУ используется для хранения резидентного программного обеспечения цифровой САУ — монитора, библиотеки прикладных подпрограмм.

Область СПЗУ или ЭППЗУ используется для хранения периодически изменяемых констант, резидентных, системных и прикладных программ. Кроме того может быть выделена часть адресов для каналов ввода—вывода, связанных с УСО и внешними устройствами цифровой САУ.

На рис. 5.13 приведена карта распределения областей адресного пространства микропроцессора КР580. В области ввода—вывода адресуются внешние устройства с двунаправленным обменом, управляемые общими с ОП сигналами чтения — записи. ППЗУ, СПЗУ и ЭСПЗУ допускают только чтение, поэтому

те же адреса можно использовать при обращении к устройствам вывода, в которых выполняется только операция записи. Аналогичным образом совместимы по адресам устройства ввода и устройства вывода. Все устройства из основного адресного пространства доступны при чтении и записи в командах, использующих абсолютную, косвенную, индексную и другие способы адресации ОП. В специальных командах ввода и вывода микропроцессора КР580 формируются специальные сигналы чтения (*ior*) и записи (*iow*). Устройства, адресуемые этими командами, занимают отдельное адресное пространство 00 ... FF₁₆ 8-разрядных адресов.

МП КР1810 позволяет организовать основное адресное пространство объемом 1 Мбайт и отдельное адресное пространство ввода—вывода — 64 Кбайт. В микроЭВМ «Электроника-60» доступно только основное адресное пространство 64 Кбайт.

Границы различных областей адресного пространства, в том числе ОЗУ, ПЗУ, устройств ввода—вывода, уточняются в процессе структурного проектирования и программной реализации алгоритмов управления в цифровой САУ. Размеры выделенных областей округляются до ближайшей целой степени 2^m с учетом параметров элементов памяти.

Во внутреннем интерфейсе микроЭВМ физическое разделение адресного пространства осуществляется общим адресным декордером, который имеет одноступенчатую структуру при разделении ОЗУ, ППЗУ, СПЗУ и двухступенчатую — при разделении адресов ввода—вывода.

В системном интерфейсе предполагается, что каждое системное устройство идентифицирует свой адрес или область адресов встроением декордером.

В мультисистемах используется двухпортовая (двухканальная) память, допускающая параллельное обращение по двум каналам. Один из каналов подключается к системной шине, а другой — к локальной шине микроЭВМ.

Для МП КР580 и КР1801 может возникнуть необходимость расширения адресного пространства, ограниченного 16-разрядным исполнительным адресом, формируемым в одной команде микропроцессора. Для этой цели используется программно-доступный регистр банка памяти. Один банк соответствует полному адресному пространству. Программа и относящиеся к ней данные должны располагаться в одном банке памяти. Для переключения программ из различных банков необходимо на внешний регистр занести номер соответствующего банка. Разработаны и другие программно-аппаратные средства расширения адресного пространства, некоторые из которых приведены в работе [98].

В микроЭВМ используется *программируемая конфигурация основной памяти*. В отличие от жесткой конфигурации, в которой разделение областей адресного пространства осуществляется адресным декордером, в программируемой конфигурации используется ППЗУ, через которое осуществляется переадресация.

Адресное пространство разделяется на логические страницы по 0,5, 1, 2 Кслов. Каждой логической странице ставится в соответствие физический адрес страницы, который преобразуется в сигналы выборки конкретных элементов памяти. На адресный вход подаются старшие разряды адреса (адрес логической страницы), а с выхода поступает адрес физической страницы. Программируемая конфигурация памяти используется в процессе проектирования микропроцессорной системы и отладки программного обеспечения. С использованием программируемого адресного преобразователя организуется «теневое» ПЗУ, в котором хранятся относительно редко исполняемые системные программы. Благодаря этому также увеличивается объем доступной памяти, возможно быстрое переключение режимов системы управления. ОЗУ может быть выполнено на элементах статической и динамической памяти. Для хранения бита информации в статическом ОЗУ используется бистабильный статический элемент, а в динамическом ОЗУ используется емкость. Для сохранения информации в динамическом ОЗУ требуется периодическая регенерация, что приводит к усложнению управления памятью и задержкам при выполнении программ. Вместе с тем интеграция в элементах динамической памяти обычно в несколько раз выше, чем в элементах статической памяти. Вследствие этого применение динамических элементов памяти позволяет сократить габаритные размеры и рассеиваемую мощность запоминающих устройств в микроЭВМ.

Регенерация может быть реализована специальной схемой, унифицированной БИС, программно или микропрограммно. В микропроцессоре КР1801 предусмотрена микропрограммная регенерация по прерыванию раз в 2 мс, цикл регенерации ОЗУ объемом 64 Кбайт занимает около 120 мкс и снижает производительность микроЭВМ на 6 %. Внешняя регенерация с использованием специальных схем и БИС управления памятью позволяет в течение 2 мс выполнить последовательно регенерацию ОЗУ в промежутках между обращениями к ОЗУ со стороны микропроцессора при выполнении программы. Унифицированная БИС управления динамической памятью КР1810ИК03 используется с микропроцессорами КР580 и КР1810.

Внешняя память в цифровой системе управления позволяет снизить требования к ресурсам микроЭВМ, в частности, сократить объем ОП, снизить стоимость и энергопотребление. Внешняя память всегда энергонезависимая, позволяет хранить и загружать программы, реализующие различные режимы управления в САУ, накапливать информацию о работе системы.

Кассетные накопители на магнитной ленте (КНМЛ) [3, 98] выполняются на основе бытовых или специальных цифровых лентопротяжных механизмов. Кроме механической части, бытовой магнитофон содержит схемы записи/воспроизведения и средства ручного управления, а цифровой — дополнительно схемы управления приводом движения ленты. Объем памяти КНМЛ до

256 Кбайт. КНМЛ является устройством последовательного доступа к информации. Скорость чтения 200 байт/с для бытового КНМЛ и 3 Кбайт/с — для цифрового.

Накопитель на гибком магнитном диске (НГМД) является одним из наиболее распространенных системных внешних устройств, позволяет использовать в цифровых САУ для управления наиболее популярные и развитые операционные системы и связанное с ними стандартное программное обеспечение.

Гибкие магнитные диски имеют размеры 203 мм и объем до 512 Кбайт, 133 мм и объем до 1 Мбайт. Скорость считывания 30 Кбайт/с и 15 Кбайт/с соответственно. Мини-диски диаметром 133 мм чаще используются в цифровых системах управления благодаря более высокой эксплуатационной надежности. Внешняя память на интегральных КМОП схемах пониженного быстродействия с встроенным батарейным питанием, обеспечивает сохранение информации при отключении основного питания. Организуют обращение к данным в памяти в соответствии со стандартом, принятым в НГМД. Элементы данных адресуются секторами и дорожками. Такой тип внешней памяти называют *электронным диском*, его емкость может достигать 256 Кбайт.

Большой интерес при выборе внешней памяти представляют накопители на цилиндрических магнитных доменах. Благодаря отсутствию движущихся частей, накопители имеют высокую надежность, низкое энергопотребление, малые габариты и сравнительно высокую скорость считывания (например, объем памяти более 256 Кбит, а время доступа 10 мс). Память организована в виде сдвигающего регистра, что позволяет представить ее как непрерывную магнитную ленту. Высокая скорость последовательного считывания позволяет при организации данных в ЦМД использовать стандартное размещение, аналогичное НГМД. Параметры серийных ЦМД-микросборок: для К1602 РУ2 объем памяти 256 Кбайт, время доступа 4 мс, скорость передачи данных 0,15 Мбит/с, потребляемая мощность 0,9 Вт, рабочий диапазон температур — 40 ... +85 °С; для К1602 РУ3 соответственно 1 Мбит, 11 мс, 0,1 Мбит/с, 1 Вт, —40 ... +85 °С.

5.7. Мультипроцессорные и мультимашинные цифровые системы управления

Практически любая достаточно сложная цифровая система содержит несколько управляющих устройств, реализуемых на основе микропроцессоров и поэтому является мультипроцессорной. Микропроцессоры используются для расчета задающих воздействий, локального управления, организации ввода—вывода, сигнализации и контроля.

В зависимости от назначения и способа построения различают иерархические мультисистемы с многоуровневой организацией управления;

параллельные мультисистемы, предназначенные для повышения производительности вычислительных средств;
 локальные сети микроЭВМ, образующие распределенные мультисистемы.

Параллельные мультисистемы в свою очередь подразделяются на мультимикропроцессорные системы, допускающие квазипараллельное исполнение программ несколькими процессорами, и мультимашинные системы, в которых возможно одновременное выполнение прикладных программ в нескольких одноплатных микроЭВМ, подключенных к системной шине.

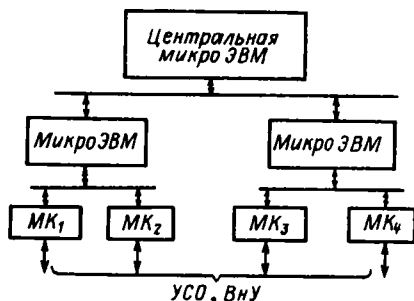


Рис. 5.14. Структура иерархической мультисистемы

В простейшем случае в качестве подчиненных выступают задачи ввода—вывода и управления внешними устройствами, а центральной является задача формирования управляющего воздействия.

Для решения основной задачи выделяется центральная микроЭВМ верхнего уровня, а подчиненные задачи решаются периферийными микроконтроллерами или микроЭВМ (рис. 5.14). Обмен данными на верхних уровнях иерархии с центральной ЭВМ может быть организован по каналам прямого доступа к памяти, а на нижнем уровне — через параллельный или последовательный интерфейс.

В сложных САУ иерархической структуры (см. рис. 1.11) выделяется верхний уровень организации управления производственным комплексом, а также один или несколько уровней локального управления сепаратными объектами. Реализация таких систем предполагает иерархическую организацию вычислительных средств.

Применение иерархических мультисистем позволяет снизить требования к центральному блоку обработки данных, разгрузить его от ряда вспомогательных функций, для выполнения которых могут быть использованы более дешевые или более доступные микропроцессоры или однокристалльные микроЭВМ. Так, например, центральный процессор серийной микроЭВМ СМ-1810 выполнен на основе МП КР1 810ВМ86, а периферийная машина,

Отметим, что приведенная классификация условна и существуют мультисистемы, включающие все перечисленные способы организации.

Иерархические мультисистемы. Возможность распределения функций между отдельными модулями аппаратных и программных средств цифровой САУ и выделение уровней их иерархии позволяет осуществить иерархическую организацию микропроцессорной системы.

предназначенная для управления вводом—выводом, может быть построена на базе МП КР580ИК80. Для управления накопителем на гибком диске также используется контроллер на основе МП КР580ИК80, а для управления клавиатурой — однокристалльная микроЭВМ КР1816ВЕ48.

Параллельные мультисистемы. Такие вычислительные системы обеспечивают параллельную или совмещенную обработку данных на нескольких микропроцессорах или микроЭВМ. Диапазон их применения охватывает широкий класс задач управления, от реализации автономных регуляторов многоканальной САУ до векторно-матричных операций и рекуррентных процедур идентификации, фильтрации и динамического программирования. Производительность параллельных мультисистем и сложность используемых для их построения программных и аппаратных средств зависят от структуры решаемой задачи, т. е. количества независимых алгоритмических модулей и характера обмена информацией между ними. Повышение производительности вычислений позволяет сократить интервал квантования T и запаздывание τ , вносимое микроЭВМ (см. п. 2.3).

Наибольший эффект от использования параллельных мультисистем достигается при реализации алгоритмов автономного управления сепаратными каналами многоканальной системы, например координатного стола непрерывной поточной линии. В этом случае естественно выделение параллельных блоков обработки информации (см. рис. 1.11), построенных на сепаратных микропроцессорах или микроЭВМ. Обмен данными между блоками в процессе работы системы практически отсутствует.

Широкое распространение находят конвейерные методы обработки информации, реализация которых возможна для многих модульных вычислительных алгоритмов. Процедура вычислений разбивается на несколько этапов (фаз), каждый из которых использует в своей работе данные, полученные на предыдущем этапе. Фазы реализуются на собственных вычислительных средствах, передающих данные в одном направлении.

Наибольший эффект совмещения соответствует условию, когда работают все блоки конвейера параллельно. Кроме этого выделяют стадии заполнения и разгрузки конвейера, соответствующие входу в цикл вычислений и выходу из цикла [85]. Если k — количество циклов, n — число фаз в цикле, то ускорение в конвейерной обработке по сравнению с последовательной может быть равно $n \cdot k / (n + k)$ при условии равной длительности всех фаз.

Вместе с тем обеспечить полное совпадение времени исполнения в отдельных фазах практически невозможно на программном уровне конвейерных вычислений. В связи с этим эффект ускорения от совмещения несколько снижается из-за необходимости ожидания окончания процессов в различных фазах.

В конвейерном режиме возможна организация взаимодействия задающего устройства и регулятора, обрабатывающего заданное

входное воздействие (рис. 1.10), решение задач идентификации, фильтрации и динамического программирования, а также реализация модульных алгоритмов управления. Однако следует учитывать, что конвейерная мультисистема, обеспечивая уменьшение интервала дискретности САУ, сохраняет или даже увеличивает время запаздывания t , которое соответствует интервалу времени от начала ввода данных на конвейер до выхода с него. Это ограничивает использование таких мультисистем в замкнутом контуре САУ.

Наиболее распространенные вычислительные процедуры в цифровых системах требуют частого взаимного обмена данными между отдельными модулями общего алгоритма. К ним, в частности, относятся векторно-матричные процедуры, применяющиеся в линейных алгоритмах (см. гл. 2), алгоритмические модули нелинейных многорежимных систем (см. п. 4.7). Для реализации одновременного выполнения таких процедур применяется общая структура параллельных мультисистем, предусматривающая использование общих ресурсов и перекрестный обмен данными внутри каждого цикла обработки информации. Параллельно могут быть выполнены отдельные модули алгоритмов управления, например модуль планирующей программы и унифицированного регулятора управляющей программы, представленной на рис. 4.46. Значительное повышение производительности мультисистемы достигается при параллельном выполнении векторно-матричных операций.

Основные проблемы организации параллельных вычислений связаны с обращением к общим ресурсам памяти и вводу—выводу с обменом информацией между параллельными ветвями вычислительного алгоритма. Для реализации параллельной или совмещенной обработки в мультисистеме используются однотипные платы процессоров или микроЭВМ. Различают аппаратные и программные средства, обеспечивающие мультиобработку. Аппаратные средства включают системный интерфейс и схемы синхронизации параллельных процессов и взаимного исключения на уровне обмена данными в интерфейсе. Программные средства, поддерживаемые многозадачной мультимашинной операционной системой (ММОС) реального времени, обеспечивают синхронизацию параллельных процессов на программном уровне во времени и в пространстве.

Синхронизация во времени учитывает продолжительность процессов, готовность внешних устройств, а также события в объекте управления, вызывающие прерывания. Синхронизация в пространстве учитывает распределение задач между процессорами и микроЭВМ в мультисистеме. Пример такой ММОС приводится в п. 6.1.

Наиболее распространенной аппаратной реализацией высокопроизводительной мультисистемы является магистральная структура, организуемая на основе стандартного системного интерфейса. В *мультимикрпроцессорной системе* к одной стандартной системной шине подключаются несколько одноплатных процессо-

ров (микропроцессоров), системная память, контроллер ввода—вывода (рис. 5.15). В системной памяти хранятся программы и данные для всех задач. Каждая задача решается строго на одном процессоре. Системный интерфейс обеспечивает приоритетный доступ процессоров к системным устройствам, а программные средства операционной системы — последовательное обращение к общим ресурсам (ячейкам памяти, устройствам ввода—вывода).

Приоритетный доступ к системным устройствам реализован, например, в стандартных интерфейсах типа *общая шина* и *мультишина* (см. п. 5.3).

В магистральной мультимикропроцессорной системе вычисления выполняются квазипараллельно, так как обращение к системным устройствам через одну и ту же шину может быть только последовательное. Вместе с тем, для микропроцессоров характерна временная диаграмма обращения к памяти, в которой между двумя последовательными обращениями системная шина свободна в течение значительного интервала времени. Это обстоятельство позволяет использовать шину при обращении к памяти со стороны другого микропроцессора.

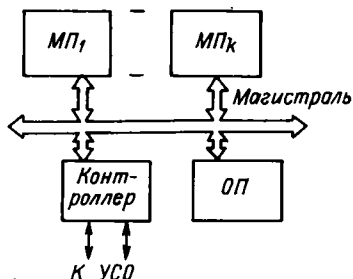


Рис. 5.15. Мультипроцессорная магистральная структура цифровой САУ

Например, на рис. 5.16 приведена временная диаграмма обращения к памяти в микроЭВМ, выполненной на основе МП КР580. Можно приблизительно рассчитать загрузку шины при выполнении различных команд. В частности, в команде с одним обращением к памяти шина занята приблизительно 1/4 времени (рис. 5.16, а); в команде с двумя обращениями — 2/7 (рис. 5.16, б); в команде с тремя обращениями — 3/10 (рис. 5.16, в) и с n обращениями — приблизительно $n/(4 + 3(n - 1))$. Таким образом, если известно распределение вероятностей p_i использования команд i -го типа в программе вычислений, то среднюю загрузку шины можно определить как $\bar{\gamma} = \sum_i p_i t_i$, где t_i — загрузка шины в i -й команде.

Для МП КР580 шина свободна приблизительно 4/5 времени работы микропроцессора и к мультишине можно подключить до 5 одноплатных процессоров. Для системной шины МПИ и процессора М2, используемого в микроЭВМ «Электроника-60», средняя загрузка шины составляет приблизительно 0,3. При этом к системной шине можно подключить до трех процессоров. Один из процессоров — центральный, осуществляет арбитраж и управление шинной. Два других процессора обращаются к системным устройствам в режиме прямого доступа с позиционным приоритетом (п. 5.3).

Загрузка процессоров и шины зависит от множества случайных факторов, в том числе от характера решаемых задач, интенсивности поступления данных для обработки, взаимосвязи между задачами по данным и т. д. Для оценки производительности мультисистемы используются математические модели или макетирование на прототипном оборудовании. Математические модели в ряде случаев дают хорошее приближение, достаточное для количественной оценки мультисистемы. При моделировании вычислительных систем хорошо зарекомендовал себя аппарат теории массового обслуживания [65]. Работу магистральной мультисистемы можно представить *одноканальной системой массового обслуживания* (СМО) с относительными приоритетами (рис. 5.17). Каждый процессор Π_1, \dots, Π_k в мультисистеме является источником заявок на обслуживание к блоку, включающему магистраль и системные устройства. Обмен данными между процессорами и ОП определяет, в основном, производительность мультисистемы. Время обслуживания t_0 включает время переключения шины $t_{ш}$, время передачи данных $t_{д}$ и не меняется для данного интерфейса и типа памяти.

В математической модели запросы шины можно представить потоком заявок на обслуживание, который характеризуется средним временем между двумя последовательными заявками α_0 или ее обратной величиной — интенсивностью λ_0 .

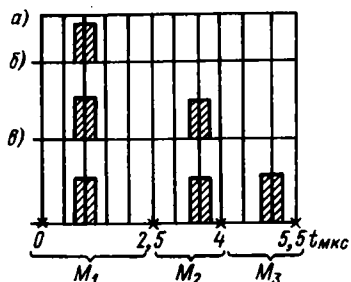


Рис. 5.16. Временная диаграмма загрузки мультишины для МП КР580: а — однобайтовая команда; б — двухбайтовая команда; в — трехбайтовая команда

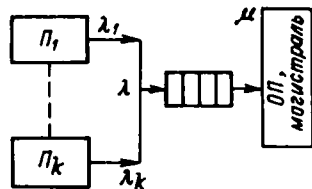


Рис. 5.17. Система массового обслуживания, моделирующая работу магистральной мультисистемы

Среднее время между заявками α_0 связано с временем обслуживания t_0 и средней загрузкой шины как $\alpha_0 = t_0/\bar{\gamma}$.

Потоки заявок от k процессоров при выполнении программ можно принять стационарными (независящими от времени) и одинарными (в каждый момент времени возможна только одна заявка).

В работе [65] показано, что сумма k независимых одинарных стационарных потоков сходится к простейшему потоку с распределением заявок по закону Пуассона с интенсивностью $\lambda = k\lambda_0 = k\bar{\gamma}/t_0$.

Интенсивность обслуживания μ в мультисистеме соответствует количеству обращений к системной ОП в единицу времени или t_0^{-1} в данном случае, где t_0 — время обслуживания. Загрузка системы $\rho = \lambda/\mu = \lambda t_0 = \bar{\gamma}k \leq 1$. Если $\rho = 1$, то интенсивность поступления заявок и интенсивность обслуживания совпадают. Для магистрали И-41 и МП. КР580 $\bar{\gamma} = 0,22$ и $\rho = 1$ означают, что мультисистема может включать не более 5 однотипных процессоров. Эта оценка является верхней границей числа процессоров в системе. Нижняя оценка может быть определена с учетом времени ожидания обслуживания в очереди. Для одноканальной системы массового обслуживания с относительным приоритетом и простейшим входным потоком получены следующие оценки среднего времени пребывания в очереди заявок m -го процессора в мультисистеме, состоящей из k процессоров [65]:

$$\bar{\omega}_m = \frac{k\bar{\gamma}t_0}{2(1 - (m-1)\bar{\gamma})(1 - m\bar{\gamma})}.$$

Задержка в исполнении команд приводит к снижению производительности процессора P_1 на 5 %, процессора P_2 на 8 %, P_3 на 15 %, P_4 на 45 %. Таким образом, производительность мультисистемы, выполненной на основе четырех микропроцессоров КР580, возрастает в 3 раза по сравнению с однопроцессорной. Добавление еще одного процессора практически не изменяет производительности мультисистемы, так как наименее приоритетный пятый процессор большую часть времени будет ожидать в очереди освобождения шины.

Таким образом, системная шина существенно ограничивает производительность магистральной мультимикропроцессорной системы. Более распространенными являются мультимашинные магистральные мультисистемы (рис. 1.14), в которых на основе системной магистрали соединяются одноплатные микроЭВМ, имеющие локальные блоки памяти. Каждая микроЭВМ преимущественно выполняет программу, которая хранится в локальной памяти, не обращаясь к системной шине. Через системную магистраль осуществляется обмен данными между микроЭВМ, обращение к общим данным в системной памяти и обращение к системным устройствам ввода—вывода. При такой организации мультисистем интенсивность загрузки магистрали снижается в несколько раз.

Локальная сеть микроЭВМ. Эта сеть используется в случае децентрализованного, распределенного управления. Сеть соеди-

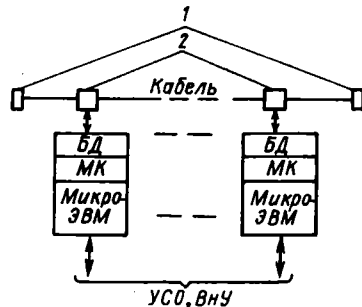


Рис. 5.18. Структура локальной сети распределенной САУ:

1 — элементы физического сопряжения; 2 — элементы электрического согласования

няет в одну систему локальные системы управления распределенного технологического процесса при расстояниях между объектами до 1 км. Каждая локальная САУ обеспечивает управление одним из объектов параллельно и независимо, а их соединение в сеть позволяет координировать работу САУ, корректировать данные, контролировать работу локальных САУ технологического процесса.

Один из распространенных вариантов организации промышленной локальной сети — *моноканал*, который использует коаксиальный или волоконно-оптический кабель для соединения узлов (рис. 5.18). Такой кабель обеспечивает необходимую в производственных условиях помехоустойчивость. В моноканале все узлы одновременно следят за состоянием линии и только один узел может вести последовательную передачу информации.

Узел сети включает блок доступа (*БД*), микроконтроллер (*МК*) и оконечное оборудование, которое может быть локальным регулятором, выполненным на основе микроЭВМ. Блок доступа обнаруживает несущую частоту в линии связи, осуществляет прием и декодирование. Контроллер сети, выполненный на основе микропроцессора, осуществляет связь между блоком доступа и оконечным устройством, формирует информационный пакет, буферизует данные при передаче, распознает адреса, исправляет ошибки. Вопросы организации локальных сетей имеют самостоятельное значение и подробно рассматриваются в специальной литературе [70].

Глава 6

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МИКРОПРОЦЕССОРНЫХ СИСТЕМ УПРАВЛЕНИЯ

6.1. Структура системного программного обеспечения цифровых САУ

Разработку программного обеспечения следует рассматривать как один из наиболее ответственных этапов проектирования цифровой системы управления, от которого зависят ее технические параметры.

Корректность разрабатываемых программ и отсутствие ошибок программирования являются необходимыми условиями безотказного функционирования САУ, исключающими аварийные ситуации, остановки оборудования и т. д.

Качество программирования алгоритмов управления существенно влияет на точность формирования управляющих воздействий, временные задержки и другие погрешности цифровой реализации регуляторов, т. е. в конечном итоге на качество процессов управления и технические показатели производственного оборудования.

Основные задачи ПО цифровой САУ разделяются на два класса: 1) прикладные, связанные с формированием управления по заданным алгоритмам регулирования; 2) системные, связанные с организацией вычислений, управлением техническими средствами цифровой САУ и обслуживанием пульта. Для организации вычислительного процесса в микроЭВМ необходимо учитывать состав и принцип работы технических средств, которые обеспечивают ввод данных для прикладных программ из УСО и вывод результатов вычислений в УСО для преобразования и передачи их в ОУ.

С пульта управления могут быть введены исходные параметры и задан режим работы микроЭВМ (ручной, пошаговый, автоматический, инициализации, контроля, рабочий и т. д.). Все эти режимы будем называть мониторными, в отличие от режимов регулирования, обрабатываемых прикладными программами. Каждому режиму соответствует некоторая частная задача управления или мониторная программа.

Все мониторные и прикладные программы связаны между собой и с техническими средствами в единый аппаратно-программный

комплекс цифровой САУ, который обеспечивает автоматическую работу этой САУ и быструю запланированную «реакцию» на внешние воздействия как со стороны ОУ, так и со стороны внешних устройств и обслуживающего персонала.

Средства связи между собой и с техническими средствами по аналогии с соответствующими средствами в аппаратуре являются интерфейсами. Определение этих интерфейсов является одной из важных задач в проектировании программного обеспечения цифровой САУ.

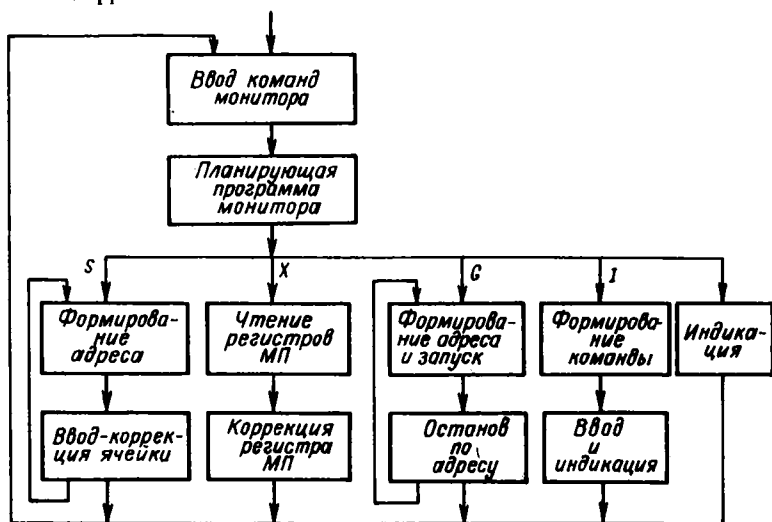


Рис. 6.1. Структура монитора цифровой САУ

Программы ввода—вывода (драйверы), управляющие обменом данными с внешними устройствами, реализуют интерфейс между прикладными программами и техническими средствами или интерфейс между мониторными программами и внешними устройствами.

Интерфейс между мониторными программами может быть определен статически в виде планирующей мониторной программы, в которой анализируются команды с пульта управления и организуются вызовы мониторных программ в определенной последовательности. В частности, в рабочем режиме вызываются прикладные программы, которые между собой могут быть связаны статически прикладной планирующей программой.

Структура монитора цифровой САУ. Структура монитора (рис. 6.1) включает программу ввода команд с клавиатуры пульта, планирующую программу монитора и программы, реализующие мониторные режимы.

В простейших микроконтроллерах вместо клавиатуры на пульте используются переключатели, каждый из которых соот-

ветствует определенному режиму. Ввод команд с пульта и выбор режима реализуют несколько команд микропроцессора.

Использование шестнадцатеричной цифровой клавиатуры, расширенной специальными символами и буквами, позволяет реализовать более сложные команды монитора и выбрать разнообразные мониторные режимы. Такие команды используются в серийных микроЭВМ для перехода в режим оперативного контроля, ввода программ, коррекции данных, запуска и отладки программы с остановкой в заданных точках, ручного контроля устройств ввода—вывода.

Команды монитора имеют простую структуру: идентифицируются первым символом-буквой, затем вводится один или два параметра в виде шестнадцатеричного числа.

Разработка программ ввода с пульта не относится непосредственно к задачам проектирования цифровой САУ. Эта программа, как и многие другие системные программы, является обслуживающей. Необходимо по возможности использовать стандартные пульты и мониторы. Вместе с тем в ряде случаев возникает необходимость создания собственных простейших пультов и мониторных программ, возможности которых могут быть существенно ограничены для конкретного применения.

Программа ввода команд с пульта (рис. 6.2) включает драйвер ввода кода символа с клавиатуры, программу идентификации символа и выбора мониторной подпрограммы. Например, символ <BK> — возврат каретки вызывает окончание ввода команды и выход в планирующую мониторную программу; символ пробела «—» пропускается; остальные символы заносятся в буфер строки команды и индицируются на индикаторе пульта.

Планирующая программа монитора (см. рис. 6.1) идентифицирует первый символ команды; анализирует синтаксис команды; вызывает системную программу, реализующую соответствующий мониторный режим; в случае синтаксической ошибки выводит признак ошибки на индикатор.

Программа режима S выбирает символы кода адреса из буфера команд, формирует адрес ОП, выбирает ячейку по адресу, выводит содержимое ячейки в шестнадцатеричном коде на индикатор пульта. Если допускается коррекция содержимого ячейки памяти, то S-программа обращается к драйверу клавиатуры пульта, читает и записывает новое значение, наращивает адрес ОП и выбирает следующую ячейку до тех пор, пока с пульта не поступит символ-признак окончания режима.

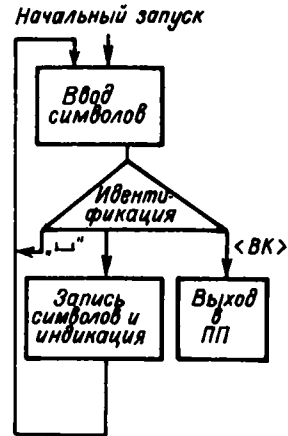


Рис. 6.2. Структура программы ввода команд монитора с пульта

Программа режима *X* читает состояние регистра микропроцессора КР580 по заданному в команде имени {*A, B, C, D, E, F*} и обеспечивает коррекцию содержимого регистра.

Программа режима *G* обеспечивает вызов прикладной программы по адресу, указанному в команде монитора, и останов в контрольной точке по заданному в команде адресу.

Мультипрограммные однопроцессорные операционные системы. Мультипрограммные операционные системы реального времени (ОС РВ) для однопроцессорных микроЭВМ включают стандартные средства управления программами с учетом динамики процессов, происходящих в объекте управления. Существует несколько ОС РВ, ориентированных на применение в различных микропро-

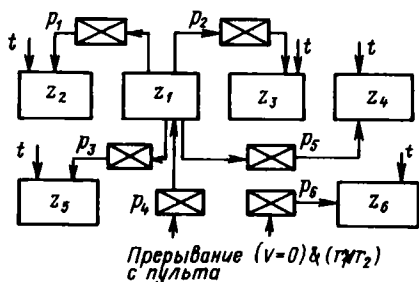


Рис. 6.3. Структура многозадачного комплекса программ цифровой САУ

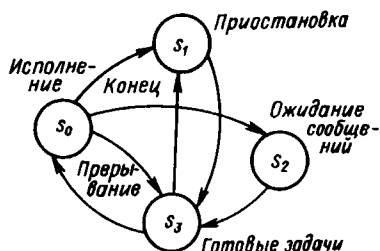


Рис. 6.4. Диаграмма состояний задач под управлением ОС РВ

цессорных системах: 1) МОС РВ для 8-разрядных микроЭВМ, выполненных на основе микропроцессора КР 580, совместимая с системой iRMX-80 [64]; 2) МОС РВ для 16-разрядных микроЭВМ с архитектурой и системой команд микропроцессора КР1801; 3) ОС РВ для высокопроизводительных микропроцессорных систем, выполненных на основе микропроцессора КР1810, и совместимая с iRMX-86.

Проектирование программного обеспечения в ОС РВ начинается с разделения сложной задачи управления на несколько более простых задач. Пример такого разделения рассмотрен в п. 1.3. Задачи оформляются как программные модули, связанные друг с другом сообщениями, а с внешними событиями — временными интервалами и прерываниями. При выполнении каждая задача вызывает самостоятельно развивающийся процесс в системе, синхронизированный с некоторым процессом в ОУ или с процессом, вызываемым другой задачей. В однопроцессорной микроЭВМ программы выполняются последовательно, обмениваясь сообщениями и ожидая сообщений о внешних и внутренних событиях.

Внешние события могут быть связаны с окончанием ввода—вывода, достижением порогового значения сигнала в УСО. Эти

события вызывают прерывания и переход к программе, формирующей сообщения для других программ.

Внутренние события связаны с окончанием временных интервалов, ограничивающих время ожидания событий или продолжительность процессов. Всем событиям и, следовательно, связанным с ними задачам, присваиваются приоритеты, которые определяют преимущества задач при выборе их для исполнения процессором.

Исключительно компактно и вместе с тем эффективно осуществляется синхронизация процессов в малой ОС РВ (для восьмизрядного микропроцессора КР580), совместимой с iRMX-80. При проектировании комплекса программ, работающего под управлением операционной системы, задается структура комплекса в виде связанной многозадачной конфигурации. Для выбора приоритетов задач и анализа эффективности работы ПО в выбранной конфигурации используется модель управления вычислительным процессом в реальном времени.

Структура многозадачного комплекса (см. рис. 1.18) для цифровой САУ представлена на рис. 6.3. Она включает задачи z_1, \dots, z_6 и связывающие их обменники p_1, \dots, p_6 или «почтовые ящики». Обменники используются для передачи сообщений между задачами. Если задача приняла сообщение, то это означает, что она готова к выполнению и может быть запущена, если нет других, более приоритетных, задач.

Каждое сообщение может быть передано только через один обменник и только к одной задаче. Вместе с тем один обменник может быть использован для передачи N сообщений соответственно N задачам-приемникам. Таким образом, одно и то же сообщение можно переслать нескольким задачам повторной передачей через один или несколько обменников. Если через один обменник передаются несколько сообщений разным задачам, то вследствие задержки в передаче и в приеме при обработке организуются очереди сообщений на входе обменника и очереди ждущих задач на его выходе. Передача сообщений из очередей происходит последовательно, в порядке их поступления.

Модель вычислительного процесса представлена на рис. 6.4 в виде графа автомата, где каждая задача может принимать одно из четырех состояний: исполняется в процессоре, ожидает в обменнике, готова к исполнению, приостановлена. При этом она конкурирует с другими готовыми задачами и с задачей, исполняемой в данный момент процессором, в соответствии с установленными для них приоритетами. В результате сравнения приоритетов задача, исполняемая процессором, может быть переведена в состояние готовности, а новая задача передана на выполнение. Множество готовых задач, задерживаемых до выполнения, организуются в очередь.

Таким образом, в состоянии выполнения может быть только одна задача в каждый момент времени; в состоянии ожидания

и приостановки может быть некоторое множество задач, представленных списком; в состоянии готовности образуется очередь задач, упорядоченных по приоритету.

Задачам, ожидающим сообщения, назначается время ожидания t , по окончании которого задача может остаться в состоянии ожидания с новым квантом времени или переходит в состояние готовности и становится в очередь для принятия решения.

Время ожидания отсчитывается программным счетчиком-таймером для каждой задачи по системным меткам времени, формируемым аппаратно каждые 50 мс. Появление сигнала метки времени с внешнего синхронизатора вызывает прерывание, по которому выполняется высокоприоритетная программа, модифицирующая таймеры ждущих задач. Если прерывания, связанные с внешними событиями в системе управления, не требуют срочного решения, то программа операционной системы, обслуживающая прерывания, формирует сообщение в обменнике для соответствующей ждущей задачи.

С учетом рассмотренной модели методами имитационного моделирования оценивается эффективность организации управления задачами в операционной системе. Для повышения эффективности управления могут быть изменены структура комплекса программ (выбрано новое разбиение на задачи), структура связей через обменники и система приоритетов. В качестве критерия эффективности используется минимум задержки принятия решения в системе или минимум времени исполнения наиболее продолжительной прикладной программы.

Конфигурация мультипрограммного обеспечения цифровой САУ (см. рис. 6.3) может быть определена использованием «описателей» задач и обменников. Описание конфигурации вместе с основными операциями управления программами входит в ядро ОС РВ. В инструментальной системе автоматизации программирования (например, СМ-1800 или СО-04) существует комплекс программ, который позволяет сформировать и записать в ПЗУ компактное ядро ОС РВ объемом 2 ... 4 Кбайт.

Ядро обеспечивает выполнение специальных функций: инициализацию при включении цифровой САУ, синхронизацию задач, отсчет времени, управление обменом сообщений. Все эти операции вызываются с использованием системных обращений в прикладных программах (например, используются операции: ПОСЛАТЬ СООБЩЕНИЕ, ОЖИДАТЬ СООБЩЕНИЕ, ПОЛУЧИТЬ СООБЩЕНИЕ БЕЗ ОЖИДАНИЯ и др.). Кроме этих операций, управляющих статической конфигурацией многозадачного комплекса программ, которая определяется в конфигурационной части ядра, в прикладных задачах могут быть использованы операции динамического изменения структуры задач. К таким операциям относятся следующие: СОЗДАТЬ ОБМЕННИК, УДАЛИТЬ ОБМЕННИК, СОЗДАТЬ ЗАДАЧУ, УДАЛИТЬ ЗАДАЧУ.

Операция **ПРИОСТАНОВИТЬ ЗАДАЧУ** позволяет любой выполняемой задаче перейти в состояние временной приостановки до тех пор, пока другая задача не переведет их в состояние готовых к исполнению задач по команде **ВОЗОБНОВИТЬ ЗАДАЧУ**.

Одной из важных проблем, возникающих при управлении многозадачным комплексом программ в ограниченном пространстве ОП микропроцессорной системы, является управление ресурсами свободной памяти в динамике с учетом запросов задач. Некоторые задачи, которым выделена область памяти, включаются и используют эту память достаточно редко. Например, задача инициализации использует часть рабочих областей ОП только в момент запуска и при включении микроЭВМ. Длина сообщений, каждому из которых отводится соответствующая область ОП, может существенно меняться. Например, при вводе—выводе память отводится для максимального сообщения и оказывается избыточной в случае коротких сообщений.

По заданию пользователя при генерации ОС РВ совместно с прикладными программами подключается программа управления, которая следит за состоянием свободной памяти, передает по запросам задач участки свободной памяти и принимает сообщения об освобождении памяти от задач. ОС РВ iRMX-86 для 16-разрядных микроЭВМ, выполненных на основе микропроцессорного комплекса КР1810, включает значительную часть средств управления задачами из операционной системы iRMX-80 для 8-разрядных микроЭВМ.

Задача управления разделяется на N подзадач в соответствии с числом различных событий в САУ (события вызывают соответствующие задачи); для всех задач устанавливаются приоритеты, которые позволяют задачам конкурировать за ресурс процессора так, что всегда выполняется наиболее приоритетная задача. Новым важным качеством, необходимым для мультипрограммной работы, является защита задач от случайного доступа и искажений со стороны других задач в процессе работы. Рассмотрим два типа задач — *активные* и *пассивные*. В отличие от активных, участвующих в состязаниях за ресурсы, пассивные задачи выполняются в свободное время как фоновые, запускаемые активными задачами. Могут быть предусмотрены несколько независимых пассивных задач. Задачи связаны между собой сообщениями, передаваемыми через обменники. При этом выполнение задачи, формирующей сообщение, задерживается до окончания обмена сообщением. Таким образом обеспечивается взаимоисключение чтения и записи в общие области памяти для связанных задач при их выполнении в однопроцессорной микроЭВМ. В операционной системе iRMX-86 предлагается быстрый способ синхронизации и взаимоисключения с использованием *семафоров*, который применяется и в мультипроцессорных системах.

Семафор представляет собой ячейку памяти, в которой устанавливается признак, учитывающий объем доступного ресурса

памяти и ввода — вывода. В частности, признак семафора может принимать только два значения: ресурс занят (признак нуль), ресурс свободен (признак единица). С использованием семафоров возможно прямое обращение к общим областям памяти, в отличие от косвенного обращения в iRMX-80 через дескрипторы, пересылаемые в обменники. При обращении к ресурсу в прикладной программе используется непрерывная *P*-операция с семафором, в которой выбирается и модифицируется ячейка семафора, хранящая соответствующий признак занятия ресурса. Если признак равен единице, то ресурс может быть занят этим процессом: устанавливается значение признака, равное нулю и продолжается решение данной задачи, которая запрашивает ресурс; если признак равен нулю, то ресурс занят и в программе организуется ожидание ресурса с повторным обращением к семафору. Освобождая ресурс, задача выполняет *V*-операцию и в ячейке семафора устанавливается признак, равный единице. Прикладные программы могут использовать определение семафоров для выделенных ресурсов памяти, общедоступных устройств ввода—вывода и включают команды обращения к ресурсам с использованием *P*- и *V*-операций. Непрерывное исполнение *P*- и *V*-операций блокирует обращение к ресурсу и семафору одновременно со стороны нескольких активных процессоров или микроЭВМ в мультисистеме. Непрерывность обеспечивается аппаратно на уровне обмена данными через системную магистраль. Например, в МП КР1810ВМ86 предусмотрена специальная префиксная команда LOCK, при выполнении которой следующая за ней команда выполняется с захватом шины до окончания операции. В операционной системе iRMX-86 используются следующие системные вызовы, которые для синхронизации задач и взаимного исключения могут быть использованы в соответствующих прикладных программах: СОЗДАТЬ СЕМАФОР, УСТРАНИТЬ СЕМАФОР, *P*-операция ПРИНЯТЬ *N* ЕДИНИЦ РЕСУРСА ИЗ СЕМАФОРА, *V*-операция ПЕРЕДАТЬ *N* ЕДИНИЦ РЕСУРСА В СЕМАФОР.

Мультiproграммная мультипроцессорная операционная система. При реализации *мультипрограммного* принципа организации ПО в *мультимашинной цифровой САУ* возникает ряд дополнительных системных задач, связанных с управлением памятью. ОП мультисистемы разделяется на общую (с доступом через системную шину) и распределенную (локализованную на платах микроЭВМ). В свою очередь, локальная память разделяется на общедоступную (с доступом через системную магистраль) и собственную (к которой возможно только монопольное обращение со стороны локального процессора). ПЗУ на плате микроЭВМ обычно является собственной памятью, а ОЗУ может разделяться на две области — общедоступную и собственную.

Общедоступная память на плате микроЭВМ организована на основе двухпортового ОЗУ так, что один порт подключается к ло-

кальному процессору, а другой — к системной магистрали. В адресном пространстве каждой микроЭВМ соответственно выделяются области адресов общедоступной и собственной памяти.

Распределение памяти в мультисистеме, состоящей из трех микроЭВМ ($MЭВМ_1$, $MЭВМ_2$, $MЭВМ_3$) и блока системной па-

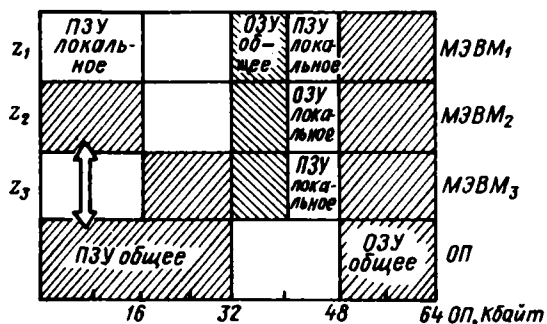


Рис. 6.5. Распределение памяти в мультипроцессорной системе (локальное ОЗУ и ПЗУ, общее ОЗУ в МЭВМ, общее ОЗУ и ПЗУ в ОП)

мяти приведено на рис. 6.5 ($MЭВМ_1$ имеет двухпортовое ОЗУ). Каждая МЭВМ выполняет собственную программу вычислений z_1 , z_2 и z_3 , которые хранятся в собственном и системном ПЗУ.

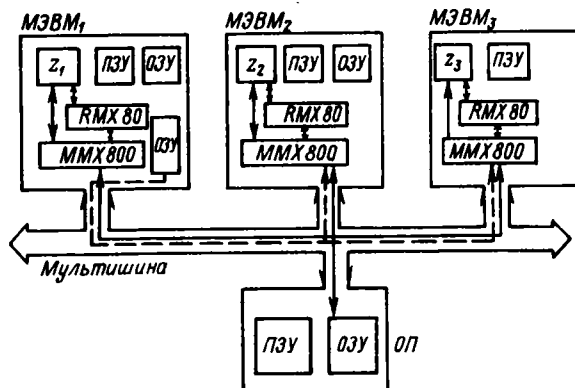


Рис. 6.6. Структура мультипроцессорной ОС РВ

Обмен данными осуществляется через общедоступные области памяти. На рис. 6.5 эти области размещены в одноплатной МЭВМ, и системной памяти (общие области адресов заштрихованы). Для управления обменом данными и синхронизации в мультисистеме, включающей несколько одноплатных микроЭВМ на МП К580, в операционной системе iMMX-800 предлагается виртуальный межпроцессорный протокол обмена сообщениями через мультишину (MIP). В памяти одноплатной микроЭВМ могут

размещаться несколько программ, выполняемых под управлением многозадачной однопроцессорной ОС iRMX-80.

Передача MIP-сообщений отличается от передачи сообщений между задачами в ОС iRMX-80 и обычного ввода—вывода. Для управления MIP-сообщениями вводится расширение ядра ОС iRMX-80 до 6 Кбайт ПЗУ, в котором хранятся также конфигурационные таблицы, учитывающие распределение памяти в мультисистеме. Схема обменов в рассматриваемой мультисистеме приведена на рис. 6.6. Для передачи сообщений организуется системный порт, управляемый iMMX-800, и локальный порт, управляемый iRMX-80; для обращения к портам используются команды: НАЙТИ ПОРТ (задается имя системного порта и возвращается адрес локального порта), АКТИВИЗИРОВАТЬ ПОРТ (задается адрес локального порта и возвращается адрес обменника из iMMX-800), ПЕРЕДАТЬ СООБЩЕНИЕ (задается адрес локального порта приемника и имя сообщения).

Прием сообщения и ожидание окончания обмена в прикладной программе осуществляются по команде ОЖИДАТЬ СООБЩЕНИЕ из iRMX-80.

На окончание обмена указывает ответное сообщение, которое считывается командой ОЖИДАТЬ СООБЩЕНИЕ. После этого системный порт деактивируется командой ДЕЗАКТИВИРОВАТЬ ПОРТ. ОС iMMX-800 следит за состоянием свободной системной памяти, выделяет память для передачи сообщений, организует и удаляет порты, создает очереди запросов к портам и управляет ими.

Возможность подключения унифицированных системных функций мультипроцессорной операционной системы позволяет существенно снизить трудоемкость разработки программного обеспечения мультисистем. Вместе с тем такие средства в настоящее время малодоступны и не решают всех задач, которые предусматриваются, например, при реализации универсальных механизмов управления процессами, предлагаемых в языке АДА [12]

Исключительная важность реализации алгоритмов управления в реальном времени и появление унифицированных операций, которые используют прикладные программы для работы в многопрограммном режиме, обуславливают реализацию ОС РВ в ПЗУ в виде резидентного программного обеспечения («кремниевая» ОС). Примером «кремниевой» операционной системы является модуль i-80130, который подключается к микропроцессору i-8086 и реализует операционную систему, функционально совместимую с iRMX-86. Модуль включает 16 Кбайт ПЗУ, блок программируемых приоритетных прерываний, три таймера и схемы управления интерфейсом. Управляющему ПЗУ отводится фиксированная область адресного пространства, выход в которую идентифицируется внешними схемами, формирующими сигналы разрешения чтения или записи в программируемые блоки прерываний и тай-

меры. Блок прерываний функционально подобен БИС *i-8259* и допускает внешнее расширение — до 57 внешних сигналов прерываний.

Кроме рассмотренных выше функций, в ОС *i-80130* используются обращения, позволяющие выделять системные ресурсы (области ОЗУ для размещения данных, программ и области стека) и создавать необходимые для задач системные типы данных. Соответствующая задача называется *работой* и формируется при обращении СОЗДАТЬ РАБОТУ. В дальнейшем в выделенных областях памяти прикладная программа может создавать связанные задачи. Для синхронизации и взаимного исключения при обращении к общим ресурсам в *i-80130* создаются очереди задач к ресурсу. Из очереди одновременно может быть выбран и обработан только один запрос к ресурсу, который реализуется как непрерывная операция. Соответствующие встроенные функции операционной системы: СОЗДАТЬ ОЧЕРЕДЬ, ПЕРЕДАТЬ УПРАВЛЕНИЕ ресурсом (запрос на ресурс поставить в очередь), ПОЛУЧИТЬ УПРАВЛЕНИЕ с ожиданием, УДАЛИТЬ ОЧЕРЕДЬ.

Алгоритмический язык высокого уровня PL/M-86, расширенный командами обращения к операционной системе, позволяет создавать компактные прикладные программы, использующие все ее возможности для управления объектом в реальном времени.

6.2. Программная реализация алгоритмов управления

Алгоритмическое обеспечение цифровой САУ включает локальные алгоритмы расчета управляющих воздействий и алгоритм переключения структуры. Поэтому основными элементами прикладной программы становятся модули локальных регуляторов R_s , подключаемые в заданных режимах работы САУ $r = r_s$ ($s = 1, 2, \dots$), и модуль ПП, производящий выбор текущего режима работы (см. рис. 4.44). Использование регуляторов унифицированной структуры (см. рис. 4.46) в значительной степени упрощает проектирование программного обеспечения для многорежимной цифровой САУ.

Реализация планирующей программы. Как показано в гл. 4, выбор режима работы САУ r_s и, следовательно, закона управления многорежимной системы, осуществляется в зависимости от принадлежности текущего состояния ОУ, характеризующегося вектором $x = \{x_i\}$ ($i = \overline{1, n}$), тому или иному подмножеству n -мерного пространства состояния R^n (см. рис. 4.37). Для анализа текущего состояния x и выбора режима r планирующая программа осуществляет последовательную проверку выполнения неравенств (4.156), (4.157).

Эти неравенства могут быть записаны в форме *предикатов*:

$$X_j = (\varphi_j(x) \leq \delta_j), \quad (6.1)$$

где X_j — переменная, принимающая значение «ложь» (0) или «истина» (1) в зависимости от конкретных значений x ; $\varphi_j(x)$ — известная функция; δ_j — заданные ограничения ($j = \overline{1, N}$).

Вычисления предикатов могут быть выполнены с использованием арифметических или логических операций. Для вычислений с использованием логических операций форма (6.1) после двоичного кодирования переменных $x = (e_1, \dots, e_n)$ и ограничений $\delta = (q_1, \dots, q_n)$ преобразуется в логическое выражение. При этом предикат (6.1) является булевой функцией, зависящей от аргументов e_1, \dots, e_n и констант (q_1, \dots, q_n) .

Например, предикат $X_j = (x \leq 5)$ в результате кодирования аргумента $x = e_1e_2e_3e_4$, константы $5 = 0101_2$ и последующих преобразований приводится к логической функции

$$X_j(e_1e_2e_3e_4) = \bar{e}_1(\bar{e}_2 \vee \bar{e}_3).$$

Вычисления с использованием логических операций выполняются поразрядно, что усложняет программирование в системе команд микроЭВМ подобного типа преобразований. Вместе с тем, способ вычислений удобен в схемной реализации на основе заказных и полузаказных БИС. Для вычислений с использованием арифметических операций предикаты (6.1) могут быть приведены к виду:

$$X_j = (\varphi_j(x) - \delta_j \leq 0).$$

Арифметическое выражение в левой части программируется в системе команд микроЭВМ. Значение предиката X_j совпадает с признаками результата арифметических операций (см. п. 5.1) (например, с признаком нулевого Z или отрицательного S результата).

Совокупность рассчитанных значений предикатов (6.1) можно представить двоичным набором (вектором)

$$G = [X_1 X_2 \dots X_N]T.$$

Связь переменной режима $r = r(s)$ и вектора G устанавливается функцией

$$r = \rho[G],$$

которая оформляется в виде таблицы [93].

Режим работы $r = r_s$ устанавливается в том случае, если рассчитан заданный в таблице s -й набор G_s значений предикатов X_1, X_2, \dots, X_N . Соответствие текущего набора G_s проверяется бинарными программами, состоящими из команд условного перехода, каждая из которых проверяет один из признаков X_j .

В общем случае каждой команде условного перехода предшествует программа, в которой вычисляется значение соответствующего предиката X_j .

Пример 6.1. При разработке бинарной программы этому значению соответствуют значения, равные нулю и единице.

Последовательный расчет и анализ значений предикатов X_j может быть выполнен в любом порядке (табл. 6.1). При этом порядок влияет на структуру, сложность и время выполнения бинарной программы. На рис. 6.7 приведена одна из таких бинарных программ, включающая минимальное число условных переходов и операторов вычисления предикатов. Можно предположить, что такая программа будет быстрее выполнена при равновероятном выборе режимов r_s . Время выполнения пропорционально числу последовательно проверяемых значений X_j при выборе режима управления.

Задача оптимизации бинарной программы решается направленным перебором на основе эвристических правил [93].

Программная реализация локальных алгоритмов. Основными элементами типовых локальных алгоритмов, рассмотренных в гл. 3 и 4, являются простейшие операции линейной алгебры: сложение векторов и

Таблица 6.1

Таблица решений

Предикат	r_1	r_2	r_3	r_4
X_1	1	0	0	0
X_2	*	1	0	1
X_3	*	1	*	0

* — безразличное значение предиката.

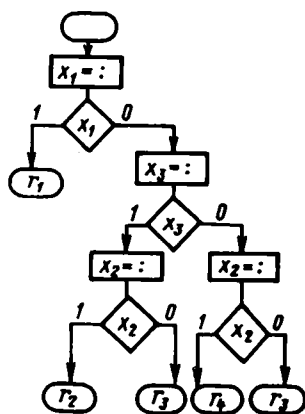


Рис. 6.7. Структура бинарной планирующей программы

умножения вектора на матрицу постоянных коэффициентов. Для нелинейных систем эти операции дополняются процедурой формирования вектора базисных функций (см. п. 4.7), который чаще всего представлен произведением степеней переменных состояния объекта управления.

Операция сложения векторов x и y размерности n сводится к сложению элементов двух одномерных массивов $\{x_i\}$ и $\{y_i\}$; она может быть выполнена как циклическая последовательная процедура в монопроцессорной системе $x_i + y_i, i = \overline{1, n}$ и параллельно — в мультипроцессорной.

Операция умножения вектора z размерности n на матрицу A размерности $m \times n$ включает параллельное или последовательное выполнение операций скалярного умножения векторов

$$a_i^T z; \quad i = \overline{1, n}.$$

Здесь a_i^T — строки матрицы A или расчета многочлена

$$\sum_j a_{ij} z_j, \quad \text{где } \{a_{ij}\} = A, \quad \{z_j\} = z.$$

Расчет многочлена по рекуррентной формуле

$$v_{1j+1} = v_{1j} + a_{1j}z_j; \quad j = \overline{1, m}$$

с начальным значением $v_{10} = 0$, может быть представлен параллельно-последовательными процедурами в различных интервалах индексов j .

Использование параллельных мультисистем (для рассмотренных операций линейной алгебры) позволяет сократить время их выполнения и тем самым ускорить вычисление алгоритма управления, т. е. снизить интервал дискретности цифровой САУ.

Формирование вектора базисных функций вида

$$z = [1 \mid x_1 \mid x_2 \mid \dots \mid x_1 x_2 \mid \dots \mid x_1^2 \mid x_2^2 \mid \dots \mid \dots \mid x_1 x_2^2 \mid \dots \mid x_1^2 x_2^2 \mid \dots]^\top$$

сводится к последовательному выполнению операций умножения и занесения результатов в память микроЭВМ.

Каждый из элементов z_j вектора z может быть получен как произведение предыдущего элемента на одну из координат вектора состояния $x = \{x_i\}$.

Вычисление степени $x_j^{s_j}$, где целое $s_j > 0$, можно представить в виде произведения сомножителей, каждый из которых является степенью с меньшим показателем. Операция умножения бинарная, поэтому оптимальный вычислительный процесс может быть представлен бинарной программой, структура которой для вычисления x^{10} приведена на рис. 6.8.

Оптимальная программа включает минимальное число операций умножения и использует минимальное число ячеек памяти

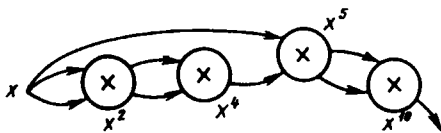


Рис. 6.8. Граф бинарной программы вычисления степени x^{10}

для хранения исходных данных и промежуточных результатов. Структура оптимальной программы для вычисления степени может быть выбрана на основании простого расчета. Число последовательно исполняемых операций

умножения определяется из условия $r = \lceil \log_2 s_j \rceil$; число ячеек памяти, необходимое для хранения операндов, соответствует числу единиц в двоичном представлении показателя s_j .

Погрешности в цифровых системах управления. Включение микроЭВМ в систему управления приводит к появлению различных погрешностей, которые суммируются и трансформируются от входа к выходу каждого блока системы управления. На рис. 6.9 приведены основные типы погрешностей: Δg — ошибка формирования входного воздействия g , поступающего с верхнего уровня управления на вход цифрового регулятора; Δy — ошибка сигнала обратной связи по y ; Δu — ошибка формирования управления в цифровом регуляторе.

Ошибка сигнала обратной связи

$$\Delta y = \Delta y_{\text{из}} + \Delta y_{\text{АЦП}}, \quad (6.2)$$

где $\Delta y_{\text{из}}$ — ошибка измерительного устройства; $\Delta y_{\text{АЦП}}$ — ошибка АЦП (ошибка квантования по уровню).

Одним из важных параметров, характеризующих качество регулирования, является абсолютная погрешность управления e_d , оговариваемая в техническом задании. Учитывая случайный характер погрешности, значение которой зависит от большого числа случайных факторов, в техническом задании может быть задана среднеквадратичная допустимая ошибка регулирования σ_d^2 . С учетом погрешности e_d может быть рассчитана разрядность АЦП.

Расчет разрядности АЦП.

Для обеспечения заданной ошибки управления e_d необходимо, чтобы ошибка задания Δg и ошибка измерения регулируемой величины Δy были значительно меньше допустимой ошибки управления, $\Delta y \ll e_d$, $\Delta g \ll e_d$.

В инженерной практике обычно принимают

$$\Delta y \approx 0,1 e_d. \quad (6.3)$$

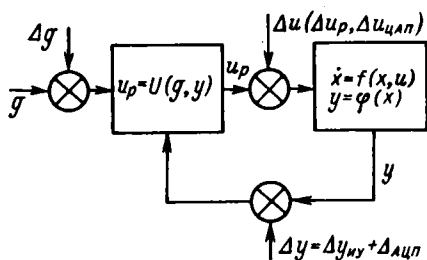


Рис. 6.9. Схема формирования погрешностей в цифровой САУ

Ошибки измерения $\Delta y_{\text{из}}$ можно считать случайными и оценивать среднеквадратичным отклонением $\sigma_{\text{из}}^2$. Значение $\sigma_{\text{из}}^2$ задается в паспорте на измерительное устройство.

Можно предположить, что ошибка АЦП является также случайной величиной, равномерно распределенной в интервале $[-0,5\delta_{\text{АЦП}} \dots 0,5\delta_{\text{АЦП}}]$, где $\delta_{\text{АЦП}} = y_{\text{max}}/d^n$ — цена младшего разряда n -разрядного кода на выходе АЦП (d — основание системы счисления). Тогда дисперсия ошибки квантования по уровню

$$\sigma_{\text{АЦП}} = \frac{\delta_{\text{АЦП}}}{2\sqrt{3}} = \frac{y_{\text{max}}}{2d^n\sqrt{3}}. \quad (6.4)$$

Разрядность аналого-цифрового преобразователя выбирают таким образом, чтобы ошибка квантования по уровню $\sigma_{\text{АЦП}}$ не превышала ошибки измерительного устройства $\sigma_{\text{АЦП}} = \rho\sigma_{\text{из}}$, где $\rho = 0,3 \div 0,5$.

Условие (6.2) преобразуется к виду

$$\sigma_y^2 = (1 + \rho^2)\sigma_{\text{из}}^2, \quad (6.5)$$

где σ_y — среднеквадратичное отклонение случайной погрешности сигнала обратной связи y .

Предполагается, что случайная погрешность измерительного устройства распределена по нормальному закону и $\Delta y \approx 3\sigma_y$.

Тогда с учетом (6.3) и (6.5) для абсолютной погрешности ϵ_d получим: $0,1\epsilon_d = 3\sigma_y$ или $0,1\epsilon_d = 3\sigma_{ny} \sqrt{1 + \rho^2}$, а для среднеквадратичного отклонения $0,1\tau_d = \sigma_{ny} \sqrt{1 + \rho^2}$. Из этих соотношений можно определить среднеквадратичное отклонение погрешности измерительного устройства σ_{ny} и $\sigma_{АЦП} = \rho\sigma_{ny}$. Тогда с учетом (6.4) можно определить разрядность АЦП:

$$n_{АЦП} = \left\lceil \log_d \frac{y_{\max}}{2\sqrt{3}\rho\sigma_{ny}} \right\rceil = \left\lceil \log_d \frac{y_{\max}\sqrt{1+\rho^2}}{0,34\rho\sigma_d} \right\rceil. \quad (6.6)$$

Разрядность ЦАП и расчетная погрешность. Ошибка управляющего воздействия Δu в цифровой САУ включает ошибки расчета Δu_p и преобразования цифрового кода в аналоговый сигнал $\Delta u_{ЦАП}$. Эти ошибки можно рассматривать как независимые, тогда $\Delta u = \Delta u_p + \Delta u_{ЦАП}$. Учитывая, что ошибка Δu_p образуется под воздействием многих случайных факторов, можно считать ее также случайной величиной, распределенной по нормальному закону с дисперсией σ_p .

Ошибка преобразования ЦАП также является случайной величиной, но распределенной равномерно в интервале $[-\delta_{ЦАП} \dots \delta_{ЦАП}]$ со среднеквадратичным отклонением

$$\sigma_{ЦАП} = \delta_{ЦАП}/\sqrt{3}; \quad (6.7)$$

тогда

$$\sigma_u^2 = \sigma_p^2 + \sigma_{ЦАП}^2. \quad (6.8)$$

Значение σ_u может быть найдено при заданных ϵ_d или σ_d в результате решения обратной задачи анализа точности динамической системы. Решение подобной задачи рассматривается, например, в [45]. Таким образом, σ_u рассчитывается исходя из допустимой ошибки ϵ_d (или σ_d) системы управления. Для выбора ЦАП необходимо определить разрядность преобразователя с учетом допустимой погрешности преобразования цифрового кода в аналоговый сигнал $\sigma_{ЦАП}$. Можно принять, что ошибка, соответствующая младшему двоичному разряду преобразователя $\delta_{ЦАП}$, не превосходит ошибки расчета и определяется как $\delta_{ЦАП} = \beta \Delta u_p$, где $\beta < 1$, или, с учетом дисперсии,

$$\delta_{ЦАП} = \beta (3\sigma_p). \quad (6.9)$$

Тогда с учетом (6.8) получим

$$\sigma_u^2 = \sigma_{ЦАП}^2 + (\delta_{ЦАП}/(3\beta))^2$$

и отсюда

$$\sigma_{ЦАП} = 3\sigma_u\beta/\sqrt{1+3\beta^2}.$$

Учитывая, что $\delta_{\text{ЦАП}} = u_{\text{max}}/2^n$ (u_{max} — максимальное значение аналогового сигнала управления на выходе ЦАП; n — разрядность двоичного кода) из (6.7) получим

$$\frac{u_{\text{max}}}{2^n} = \left(\sigma_u \frac{3\beta}{\sqrt{1+3\beta^2}} \right) \sqrt{3};$$

$$n_{\text{ЦАП}} = \left\lceil \log_2 \frac{u_{\text{max}} \sqrt{1+3\beta^2}}{\sigma_u \beta 3 \sqrt{3}} \right\rceil = \left\lceil \log_2 \frac{u_{\text{max}} \sqrt{0,33 + \beta^2}}{3\beta \sigma_u} \right\rceil.$$

Аналогичным образом при заданном значении σ_u можно определить допустимую погрешность расчета σ_p из (6.7)—(6.9)

$$\sigma_u^2 = \sigma_p^2 + 3\beta^2 \sigma_p^2;$$

$$\sigma_p = \sigma_u / \sqrt{1+3\beta^2};$$

$$\Delta u_p = 3\sigma_p = 3\sigma_u / \sqrt{1+3\beta^2} = \Delta u / \sqrt{1+3\beta^2}.$$

Погрешности вычислений. При реализации алгоритмов цифрового управления необходимо выбрать такую схему вычислений и программу, которая дает погрешность не более расчетной Δu_p . Вычислительный процесс вносит целый ряд взаимосвязанных погрешностей, влияние которых можно сократить до требуемого уровня, выбирая метод и оптимальную схему вычислений. Основные погрешности разделяются на два класса: методические Δu_m и инструментальные Δu_n .

Методическая погрешность возникает вследствие использования численных методов для реализации оператора $u = U(g, y)$. Так, если оператор U является интегродифференциальным, то возникает методическая ошибка за счет замены этого оператора приближенным суммарно-разностным, квантования процессов во времени и запаздывания, вносимого ЭВМ в контур управления. В гл. 3 и 4 предлагаются в основном суммарно-разностные операторы, при синтезе которых на этапе составления разностных моделей ОУ и внешних воздействий ошибки квантования во времени и запаздывания в микроЭВМ компенсируются. Методическая ошибка при реализации суммарно-разностных операторов может возникнуть в том случае, если операторы содержат нелинейные функции и для их вычисления используются приближенные численные методы [47]. При этом часто применяют хорошо изученные методы разложения функции в степенные ряды и итерационные методы, для которых разработаны способы оценки ошибки вычислений, и может быть выбрана схема вычислений с заданной методической погрешностью. Необходимо иметь в виду, что аргументы нелинейных зависимостей в управляющих ЭВМ могут быть приведены к диапазону, не превосходящему по модулю единицы. Это обуславливает возможность применения небольшого числа членов ряда разложения или небольшого числа итераций при расчете нелинейных функций.

Методическая ошибка Δu_m в зависимости от реализации F алгоритма управления может суммироваться с другими погрешностями вычислений и трансформироваться в процессе вычислений в новую погрешность.

Инструментальная погрешность Δu_n появляется за счет округления результатов операций, выполненных с числами в форматах с конечной длиной. Исходные данные, вводимые в микроЭВМ с АЦП, могут быть целые и дробные, двоичные и двоично-десятичные числа. При вычислениях целесообразно использовать операции с двоичными числами, а не с двоично-десятичными, так как время вычислений, объем которых при формировании управляющих воздействий может быть значительным, с использованием операций двоичной арифметики существенно сокращается. Вместе с тем, при переходе к двоичным числам необходимо учитывать появление дополнительной задержки в вычислениях, связанной с выполнением программы перевода. Кроме того, при переводе дробных двоично-десятичных чисел в двоичную систему счисления появляется погрешность перевода δ_2 .

Погрешность входных данных для микроЭВМ в этом случае определяется как $\delta_x = \delta_{\text{АЦП}} + \delta_2$. Можно принять $\delta_2 = k\delta_{\text{АЦП}}$ ($k < 1$), тогда число двоичных разрядов n , необходимое для представления входных данных, определяется из соотношения $2^{-n} < k\delta_{\text{АЦП}}$, где 2^{-n} — погрешность усечения в n -разрядном формате.

Все возникающие при вводе данных в микроЭВМ погрешности, а также погрешности промежуточных преобразований по алгоритму F переходят в трансформированные погрешности $\Delta u_t = F(\delta_{\text{АЦП}}, \Delta u_n, \Delta u_m)$, которые вместе с аддитивной составляющей методической погрешности Δu_m не должны превосходить погрешность расчета Δu_p , т. е.

$$\Delta u_m + \Delta u_t < \Delta u_p.$$

Трансформированные погрешности. Значения этих погрешностей можно сохранять на требуемом уровне (с точки зрения окончательной погрешности Δu_p), выбирая форматы данных и алгоритмы вычислений. Для сохранения точности, определяемой через абсолютную погрешность Δu_p , используется формат переменной длины с фиксированной запятой [76]. Данные должны быть приведены к диапазону целых или дробных чисел путем масштабирования. Переменный формат при вычислениях в микроЭВМ принимается кратным длине одного из форматов с фиксированной запятой, поддерживаемых командами. В частности, для микропроцессора КР580ИК80 формат кратен 8, для микроЭВМ «Электроника-60» — 16. Например, 16-разрядный формат с фиксированной запятой имеет вид $[\pm]_{(,)} \overset{0}{\text{---}} \overset{1}{\text{---}} \dots \overset{15}{\text{---}}$, где в старшем бите кодируется знак числа (0 — плюс, 1 — минус), а разряды с 1 по 15 содержат цифры двоичного кода абсолютной

величины. Запятая (,) может быть фиксирована при программировании перед старшим разрядом (дробные числа) или младшим разрядом (целые числа).

Изменение диапазона промежуточных величин в процессе вычислений автоматически учитывается в форматах с плавающей запятой. Например, один из наиболее распространенных форматов в микроЭВМ $[\pm] [X] [M]$ включает бит знака мантиссы, поле характеристики X (8 разрядов) и поле мантиссы M (23 разряда). В формате записывается число, представленное в полулогарифмической форме $Z = \pm 2^P 0, z_1 z_2 \dots z_{24}$. Характеристика числа определяется как $X = P + 2^7$, а мантисса — дробным нормализованным числом в диапазоне $[0,5 \dots 1 - 2^{-24}]$, причем старшая цифра $z_1 = 1$.

В любом диапазоне, определяемом этим форматом [76], сохраняется постоянная относительная погрешность $\delta_z = \Delta z / Z = 2^{-24}$, а по окончании вычислений погрешность $\Delta u_p = 2^{-24} u_{p \max}$ удовлетворяет требованиям к точности расчета управления в САУ. Для выполнения операций с плавающей запятой в микроЭВМ используются специальные арифметические расширители (см. п. 5.3) или стандартные подпрограммы. В последнем случае время вычислений может быть значительным и предпочтительно при отсутствии расширителей использовать форматы с фиксированной запятой и масштабирование.

Масштабирование используется для сохранения точности вычислений с учетом ограниченных форматов с фиксированной запятой. Выполняя масштабирование в процессе вычислений, можно выбирать самые короткие форматы при сохранении точности с учетом диапазона. Масштабирование при вычислениях напоминает работу с порядками в форматах с плавающей точкой, если предположить, что все масштабы, для удобства, — целые степени с основанием два. Таким образом, в худшем случае по трудоемкости эти два способа не отличаются, однако существенно то, что масштабы могут быть учтены и выбраны на этапе реализации алгоритма управления, что позволяет целенаправленно сократить число операций с масштабами и упростить вычисления. В табл. 6.2 приведены абсолютные и относительные погрешности арифметических операций и некоторых функций, которые могут быть использованы для расчета трансформируемых погрешностей, масштабирования и выбора промежуточных форматов данных.

Методика оценки погрешности вычислений с учетом погрешностей исходных данных и их трансформации приведена в [47]. Программа представлена в виде графа вычислительного процесса (рис. 6.10). Для каждой операции известна относительная погрешность с учетом округления. В большинстве случаев для упрощения используется округление усечением (в табл. 6.2 значения погрешностей учитывают также этот способ округления).

Погрешности арифметических операций

Операция	Абсолютная погрешность Δs	Относительная погрешность δs
$(a + \Delta a) \pm (b + \Delta b)$	$\Delta a + \Delta b$	$\frac{\Delta a + \Delta b}{a \pm b}$
$(a + \Delta a) (b + \Delta b)$	$a \Delta b + b \Delta a$	$\frac{\Delta a}{a} + \frac{\Delta b}{b}$
$\frac{a + \Delta a}{b + \Delta b}$	$\frac{a \Delta b + b \Delta a}{b (b + \Delta b)}$	$\frac{\Delta b}{b + \Delta b} + \frac{\Delta a}{a}$
$(a + \Delta a)^n$	$n \Delta a$	$n \Delta a/a$
$\ln (a + \Delta a)$	Δa	$\Delta a/a$

Анализ программ по точности с учетом диапазона изменения промежуточных результатов может быть выполнен с использованием интервальных вычислений. При этом разрабатывается интерпретатор операций, который анализирует диапазон результата операции, погрешность результата с учетом диапазона операндов

и их погрешностей, выбирает масштаб и формат представления результата в различных поддиапазонах.

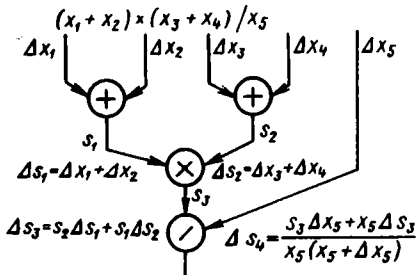


Рис. 6.10. Граф вычислительного процесса

по методам и алгоритмам для правильных дробей [76]. Отрицательные числа хранятся в памяти в дополнительном коде ($2 - |x|$), что позволяет весь диапазон значений в n -разрядном формате с запятой, фиксированной перед старшим разрядом, рассматривать как непрерывный. При этом старший дробный разряд также является значащей цифрой и значение его совпадает с кодом знака.

Диапазон представления данных с учетом точности согласуется с форматом при выборе подходящего масштаба. Обычно принимают масштаб равным 2^r , где r — целое число. Тогда $x = m_x \cdot \xi = 2^r \cdot \xi$, где ξ — машинное представление в формате.

Операция суммирования $s = x_1 + x_2$ с учетом масштабов соответствующих записей в форматах может быть представлена в виде $m_s \hat{s} = m_1 \hat{x}_1 + m_2 \hat{x}_2$, где m_s уточняется при выравнивании масштабов слагаемых: $m_s = \max(m_1, m_2)$. Процедура сложения n -разрядных чисел включает выравнивание масштабов, суммирование, анализ переполнения по признаку результата V и его коррекцию в случае переполнения, а также формирование минимального значения $\hat{s} = 2^{-n}$ с минимальным масштабом при исчезновении значимости ($\hat{s} = 0$).

Операция умножения выполняется в соответствии с алгоритмом для дробных чисел. При этом для n -разрядных сомножителей произведение занимает $2n$ -разрядный формат. С учетом масштабов произведение $s = x_1 \cdot x_2$ вычисляется по формуле $m_s \hat{s} = (m_1 \times m_2) \hat{x}_1 \cdot \hat{x}_2$, где $m_s = m_1 m_2$. Произведение приводится к n -разрядному формату с учетом погрешности $\Delta = \max\{\hat{x}_1, \hat{x}_2\}$ и уточняется масштаб m_s .

Операция деления также выполняется по алгоритму для дробных чисел. Для n -разрядных делителя и частного делимое приводится к $2n$ -разрядному формату в процессе деления или перед началом операции. С учетом масштабов операция деления выполняется по формуле $m_s \hat{s} = (m_1/m_2) (\hat{x}_1/\hat{x}_2)$, где $m_s = m_1/m_2$.

Делимое и делитель предварительно выравниваются так, что $2x_1 \geq x_2 > x_1$, и выполняется машинная операция или стандартная процедура деления n -разрядных чисел; при формировании частного учитываются максимальные отрицательные значения (-1) для делимого и делителя.

Для вывода расчетного значения u_p в ЦАП при вычислении с фиксированной запятой необходимо выполнить масштабное преобразование с учетом диапазона величины u на выходе ЦАП, причем $u = \delta_{\text{ЦАП}} u_p$. На вход ЦАП всегда поступает двоичное число u_p в n -разрядном формате с фиксированной запятой. Если $n = 8$, то это число может быть представлено в любом диапазоне с масштабом $m_p = 2^{8k}$, где k — целое или равное нулю число.

В результате вычислений в форматах с фиксированной точкой получено расчетное значение \hat{s}_z с масштабом $m_z = 2^r$, причем $m_p u_p = m_z \hat{s}_z$. После выравнивания масштабов для выделения значения u_p получим

$$u_p = \frac{m_z}{m_p} \cdot \hat{s}_z = 2^{r-8k} \hat{s}_z.$$

Если $r - 8k < 0$, то \hat{s}_z сдвигается вправо на $(r - 8k)$ разрядов и все разряды, которые выходят за пределы формата, усекаются. В частном случае может быть получено значение $u_p = 0$. Если значение $r - 8k > 0$, то \hat{s}_z сдвигается влево на $(r - 8k)$ разрядов.

Значение k выбирается из условия $2^{n_k} u_{p \max} = 2^r \hat{s}_{z \max}$, тогда

$$k = \left[\frac{r - \lceil \log_2 (u_{p \max} / \hat{s}_{z \max}) \rceil}{n} \right].$$

6.3. Методы и средства разработки и отладки программ

Средства массового проектирования программного обеспечения для микропроцессорных систем, объединенные единой методикой, обеспечивающие документирование на всех этапах, имеющие инструментальную поддержку, объединяются в технологию разработки программ [13].

Одним из важных достижений технологии является метод структурного программирования, который опирается на стратегию нисходящего проектирования программ и структурированную запись [11].

В стратегии нисходящего проектирования осуществляется последовательное «сверху вниз» формирование структуры программы. При этом проектирование разделяется на несколько уровней детализации. На верхнем уровне разрабатывается структура, состоящая из крупных блоков, подробно определяются связи (интерфейсы) между блоками и функции, реализуемые в блоках. Выделенные таким образом блоки в дальнейшем можно уточнять независимо и параллельно, сохраняя соглашения о связях.

На начальных этапах используются формы записи, близкие к естественному языку, и структурные схемы. На последующих этапах блоки заменяют отлаженными программными модулями на алгоритмических языках.

Модульность программ позволяет организовать параллельную разработку и отладку, унифицировать процедуры в библиотеки модулей и унифицировать методы сборки. Наиболее популярной формой изображения алгоритмов и программ на промежуточных стадиях нисходящего программирования являются схемы алгоритмов, основным достоинством которых является графическое наглядное изображение.

Графический стиль программирования в настоящее время интенсивно развивается и поддерживается инструментальными системами на персональных микроЭВМ.

Одной из новых графических форм представления алгоритмов и программ является Р-схема в Р-технологии программирования [13]. Язык Р-схем принят в качестве одного из стандартов и обеспечивается инструментальными средствами. При разработке программного обеспечения цифровых САУ широко используются и развиваются традиционные методы и средства программирования микропроцессорных устройств на основе серийных инструментальных систем. Эти системы включают средства автоматизации программирования, средства комплексной отладки и записи программ в ПЗУ.

Автоматизация программирования позволяет снизить трудоемкость и обеспечить высокую надежность на этапах алгоритмизации и реализации алгоритмов цифрового управления САУ в системе команд управляющей микроЭВМ. Для описания алго-

ритмов предлагаются разнообразные алгоритмические языки, средства редактирования и автоматического преобразования программ в коды команд микропроцессора.

Комплексная отладка аппаратуры и программного обеспечения цифровой САУ является наиболее трудоемким этапом, требующим высокой квалификации, и выполняется вручную. Для ускорения и повышения надежности отладки используются инструментальные системы, которые включают специальную аппаратуру и ПО, облегчающие сбор и анализ необходимой диагностической информации.

Отлаженные резидентные программные модули загружаются с помощью автоматических *программаторов* в БИС ППЗУ или СППЗУ. После контроля записи БИС устанавливаются в разъемы платы микроЭВМ.

Различают два основных класса инструментальных систем: кросс-системы и прототипные микропроцессорные системы.

Кросс-системы реализованы на высокопроизводительных ЭВМ с развитыми средствами ввода — вывода. Основные требования к таким системам — широкая доступность, минимальное время трансляции алгоритмического описания в объектную программу микропроцессора, документируемость всех этапов проектирования. В наибольшей степени этим требованиям удовлетворяют ЭВМ общего назначения, установленные в вычислительных центрах коллективного пользования практически во всех учреждениях и предприятиях. Доступность обеспечивается не только их распространенностью, но и минимальной стоимостью проектирования, в которой учитывается высокая производительность, развитые средства программирования и редактирования, доступные в режиме разделения времени.

С распространением персональных микроЭВМ и инструментальных систем на их основе одним из важных условий выбора кросс-ЭВМ становится наличие интерактивных средств проектирования.

Кросс-средства на ЕС ЭВМ, мини-ЭВМ СМ-1420 используются для программирования микропроцессоров КР580 (КР1810); на микроЭВМ «Электроника-60» — для микропроцессоров КР580 и однокристалльных микроЭВМ; на микроЭВМ СМ-1800 — для КР1810 и однокристалльных микроЭВМ.

В *прототипных средствах* используется инструментальная микроЭВМ, система команд которой совпадает с системой команд проектируемого микропроцессорного устройства. Благодаря общей системе команд, в прототипной инструментальной системе появляются дополнительные эффективные средства отладки программ и аппаратуры в комплексе. Кроме того, все средства автоматизации программирования, разрабатываемые для прототипных персональных и управляющих микроЭВМ, непосредственно применимы и в инструментальных системах при проектировании цифровых САУ. Вследствие этого средства программи-

рования существующих прототипных систем превосходят по объему, возможностям и качеству кросс-средства программирования, хотя и уступают в производительности.

Средства автоматизации программирования. Основой средств автоматизации является алгоритмический язык, который позволяет сформулировать алгоритм обработки данных в строгом, формализованном виде. Даже при отсутствии соответствующих трансляторов формальная запись алгоритма (спецификация) необходима для надежного программирования, документирования, организации технологического процесса программирования. Алго-

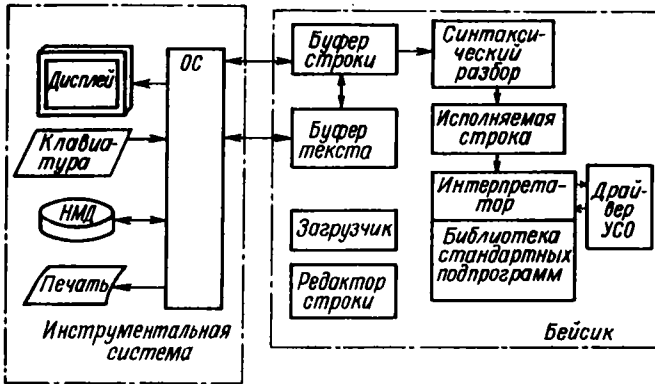


Рис. 6.11. Схема аппаратного и программного обеспечения интерпретации

ритмическое описание для выполнения соответствующей программы может быть преобразовано либо в процессе интерпретации, либо в результате компиляции. Алгоритмические языки также традиционно разделяются на языки, преимущественно используемые при интерпретации, и языки, которые поддерживаются только компиляторами. К языкам первого типа относится широко используемый в микропроцессорных системах БЭЙСИК.

Интерпретация программ на языке БЭЙСИК. Структура технических средств и схема интерпретации программ на языке БЭЙСИК приведены на рис. 6.11. Технические средства входят в инструментальную систему и включают дисплей, клавиатуру, накопитель на магнитном диске, печать. Инструментальная операционная система обеспечивает стандартный ввод — вывод при работе в интерактивном режиме. В этой схеме интерпретатор языка БЭЙСИК и исполняемая программа размещается во внешней памяти. В системе управления используется версия управляющего языка БЭЙСИК, допускающая расширение для работы с объектом управления. Хранение программ во внешней памяти позволяет использовать управляющую микроЭВМ как универсальную в различных системах управления (корректировать прикладные программы, изменять режим работы, вызывая с пульта

новую программу управления). Возможно также применение специализированной системы управления, в которой интерпретатор языка БЭЙСИК и управляющая программа являются резидентными и хранятся в ПЗУ.

Язык БЭЙСИК можно разделить на функциональные блоки: *загрузчика*, *редактора*, синтаксического разбора, *интерпретатора*, библиотеки стандартных подпрограмм.

Загрузчик используется для чтения или записи текста программы из внешней памяти в буфер текста, вывода на печать, выборки строки текста в буфер строки для интерпретации. Строка является основной адресуемой и синтаксически завершенной конструкцией в языке БЭЙСИК. При выполнении программы строки последовательно выбираются из текста в буфер строки, преобразуются синтаксическим анализатором в промежуточную форму и передаются для выполнения интерпретатору.

Интерпретатор просматривает строку и формирует вызовы стандартных подпрограмм из библиотеки для их выполнения.

БЭЙСИК включает текстовый *редактор*, который позволяет в интерактивном режиме корректировать вводимую или любую вызываемую строку в буфере строки. Строка может быть выполнена сразу после ввода или загружена в буфер текста. Загружаемые в буфер текста строки упорядочиваются по номерам. На любом этапе ввода можно перейти к исполнению текста для его отладки.

Подготовленный и отлаженный текст программы с помощью автоматического *программатора* в инструментальной системе вместе с интерпретатором записывается в ПЗУ, которое может быть установлено в цифровой САУ для резидентного использования. Стандартными средствами языка являются арифметические операции $\{+, -, *, /\}$, функции $\{\sin, \cos, \tan, \exp, \dots\}$, операторы управления $\{\text{for } \dots \text{ to } \dots, \text{if } \dots \text{ then, goto, gosub}\}$, оператор ввода с клавиатуры *inp*, оператор печати *print*, оператор записи файла *save*, чтения файла *old*, переименования файла *name* и другие.

Расширение языка для управляющих микропроцессорных систем на основе МП КР580 включает оператор ввода байта из j порта $A = \text{inp}(j)$, оператор вывода байта A в j порт $\text{out}(j)[A]$, запись байта в ОЗУ по адресу, запись и чтение байта из стека, обращение к подпрограмме пользователя *call* (...).

Через стек передаются параметры в подпрограмму пользователя, к которой затем из основного алгоритма возможно обращение по команде *call*. Драйверы ввода—вывода, побитовая обработка, управление таймером, обработка прерываний, отсутствующие в языке БЭЙСИК, могут быть реализованы в виде подпрограмм в машинных кодах. Достоинствами интерпретируемого БЭЙСИК'а являются: удобство программирования и отладки программ в исходных текстах в интерактивном режиме, а также диалоговый режим взаимодействия с оператором при выполнении. Основные недостатки: 1) большой объем памяти, который занимает исходный

текст и программное обеспечение языка БЭЙСИК; 2) замедление исполнения, связанное с анализом текста и формированием вызовов в интерпретаторе, программной интерпретацией десятичных арифметических операций высокой точности с плавающей точкой. Вследствие этого БЭЙСИК используется в относительно медленных цифровых САУ с интервалом квантования $T > 0,1$ с.

Ускорение выполнения программ в несколько раз достигается с использованием *компиляции*. Существуют компиляторы языка БЭЙСИК, которые после отладки текста в режиме интерпретации транслируют программу в компактную форму, при исполнении

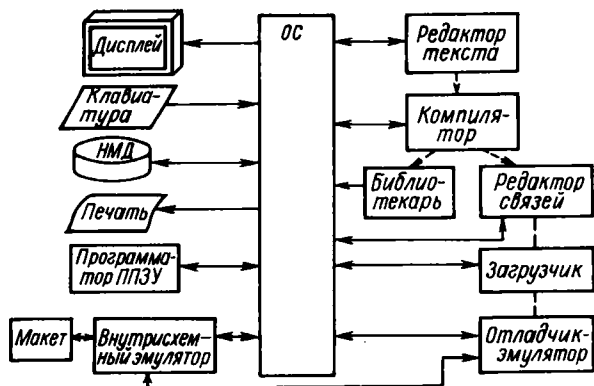


Рис. 6.12. Структура процесса компиляции

которой не требуется синтаксического разбора, формирования вызовов подпрограмм. Реализация большинства алгоритмических языков предусматривает только компиляцию.

Технология программирования с компиляцией. Структура инструментальных средств для компиляции приведена на рис. 6.12 и включает комплекс сервисных программ, реализующих отдельные этапы проектирования программного обеспечения. Сервисные программы используются под управлением инструментальной операционной системы, промежуточные результаты формируются и хранятся во внешней памяти в виде файлов.

Подготовка прикладной программы начинается вводом текста на алгоритмическом языке с клавиатуры и включает его редактирование на экране дисплея. Для этой цели средствами операционной системы вызывается сервисная программа *редактор*. Завершается редактирование записью текстового файла во внешней памяти.

Текстовый файл на следующем этапе загружается и обрабатывается *компилятором*, который создает *перемещаемый* (загрузочный) *файл* в кодах команд микропроцессора. Для каждого алгоритмического языка существуют соглашения по оформлению программ, загружаемых в ПЗУ. При компиляции в этом режиме

разделяются области программ в ПЗУ, область рабочих ячеек и переменных в ОЗУ.

Программа может быть оформлена в виде библиотечного модуля. Имя модуля заносится в каталог библиотеки, при создании загрузочного модуля сохраняются таблицы с именами переменных для связи между модулями. Организация каталога библиотеки является функцией сервисной программы *библиотекарь*.

Если в прикладной программе используются ссылки на программные модули, хранящиеся в библиотеке, то с помощью *редактора связей (компоновщика)* выполняется согласование связей и объединение модулей в одну программу.

Для исполнения программы вызывается либо простой загрузчик, либо специальная сервисная программа *отладчик*. *Загрузчик* формирует абсолютные адреса, соответствующие заданному размещению вызываемой программы в ОЗУ, и *абсолютный файл*. *Абсолютный*, или *объектный файл*, автоматически сохраняется во внешней памяти и может быть исполнен в прототипной инструментальной системе.

Отладчик может загружать также, как и *загрузчик*, перемещаемый файл в заданную область ОЗУ, но может и выполнять абсолютный файл, ранее подготовленный загрузчиком в реальных адресах управляющей микроЭВМ. В динамическом отладчике *DDT* в операционной системе CP/M, например, могут быть использованы следующие команды отладки: дизассемблирование части абсолютного файла с выводом на дисплей; запуск программы с заданного адреса до точки контроля; трассировка исполняемой программы, в процессе которой выводится содержимое регистров микропроцессора после выполнения каждой команды; ввод в указанное место программы дополнительного фрагмента в мнемонике *ассемблера*; вывод и изменение содержимого регистров микропроцессора и ячеек ОЗУ.

Отладчик является универсальной программой и может обеспечить только стандартные режимы отладки, независимые от конкретной организации и работы ввода—вывода в цифровой САУ. Поэтому в развитых инструментальных системах отладка выполняется с использованием аппаратно-программных средств (*эмуляторов*). В кросс-системах не всегда технически доступна эмуляция, по этой причине для отладки используются моделирующие программы (*симуляторы*).

Таким образом, технологический процесс, связанный с компиляцией, длительный и трудоемкий. При обнаружении ошибок часто необходимо повторять все ранее пройденные этапы. Вместе с тем, в процессе компиляции используются эффективные средства интерактивного редактирования всего текста, оптимизирующие компиляторы, создающие компактную программу в машинных кодах; библиотекари, поддерживающие модульный стиль программирования и способствующие унификации при проектировании ПО цифровых САУ.

На основе инструментальных систем разрабатываются универсальные средства автоматизации программирования, в которых алгоритмы могут быть записаны на разных алгоритмических языках, организованы в библиотеки модулей и объединены в одну программу на уровне машинных кодов. Широко распространены инструментальные программные средства, ориентированные на использование одного алгоритмического языка высокого уровня и языка *Ассемблера*.

Такие средства тесно связаны с инструментальной операционной системой микроЭВМ, которая реализует интерактивный экраный режим редактирования, обеспечивает хранение и автоматический доступ к программам и данным, организуемым в файлы, осуществляет связь со стандартными устройствами ввода—вывода (печать, терминал, НГМД, ФС, ПЛ и т. д.).

Система программирования на языке Паскаль. Язык Паскаль является одним из наиболее распространенных в инструментальных системах программирования. Достоинства языка заключаются в лаконичной записи алгоритма, возможности применения разнообразных структур данных, применении структурированного подхода к программированию.

Одна из распространенных систем программирования на основе языка Паскаль для микроЭВМ — система Паскаль МТ-плюс, которая включает *компилятор, редактор связей, дизассемблер, отладчик, библиотекарь*.

Язык Паскаль МТ-плюс, кроме стандартных средств [11], включает следующие машинно-ориентированные расширения для МП КР580: 1) дополнительные типы данных — байт (*byte*), слово (*word*), соответствующее машинному слову, строка символов (*string*); 2) операторы ввода (вывода) байта из (в) порта с номером *N inp [N] (out [N])*; 3) процедуры для установки и гашения бита в слове, сдвига слова влево и вправо; 4) процедуру передачи байтов; 5) специальные аппаратно-ориентированные процедуры доступа к адресу переменной, ожидания сигнала готовности при обращении к порту ввода — вывода, управления прерываниями (разрешение и запрещение); 6) средства включения констант любого типа и текста линейной программы на *Ассемблере*.

Подготовка и редактирование текста программы осуществляется с использованием универсального *редактора* операционной системы СР/М.

Компилятор с языка Паскаль МТ-плюс создает два файла по требованию: файл листинга для вывода и загрузочный (перемещаемый файл) для последующих этапов проектирования программы.

С помощью собственного *редактора связей* к загрузочному файлу подключаются библиотечные модули и модули управления отладкой. После загрузки подготовленного для использования отредактированного загрузочного файла разрешается исполнение в режиме отладки. Текст программы может быть распечатан в смешанной форме, включающей операторы языка и их реализацию на

Ассемблере. Отладка программы осуществляется по командам в нескольких режимах с остановами для ввода новых команд. Используются следующие режимы: слежение с остановками после каждой строки операторов или после заданного числа строк; запуск с начала программы; останов по имени процедуры.

Исполнение в режиме отладки сопровождается распечаткой диагностической и контрольной информации. Программы могут быть оформлены в виде модулей на языках Паскаль или *Ассемблера* с внешними ссылками, которые доопределяются на этапе редактирования связей.

Модули транслируются с использованием компилятора языка Паскаль или совместимым с ним на уровне соглашений по организации связей компилятором *Ассемблера*. При редактировании связей в одну программу объединяются модули обработки данных, модули управления вводом—выводом через УСО, модули обработки прерываний.

Система программирования на языке Фортран. Язык Фортран-80 для микропроцессора КР580 содержит весь набор стандартных операторов и средств, соответствующий стандартному описанию языка Фортран-IV, кроме комплексных типов. Язык является наиболее распространенным в инженерной практике, используется на ЭВМ всех классов, что обеспечивает совместимость программного обеспечения на уровне алгоритмического языка.

Расширения языка, связанные с его использованием для программирования управляющих микроЭВМ, включают функции обращения к ОП по адресу и к портам ввода—вывода.

Функция *peek (i)* позволяет выбрать байт по адресу i , адрес может быть представлен константой или переменной.

Обращение к подпрограмме *call poke (i, b)* позволяет записать байт b в ОП по адресу i . Ввод с порта i осуществляется функцией *inp (i)*, а вывод байта в i порт при обращении к подпрограмме *call out (i, b)*.

В языке Фортран не предусмотрена организация библиотеки модулей и модульный принцип программирования. Унифицированные программные блоки оформляются как подпрограммы или выбираются из библиотеки стандартных подпрограмм, которые включаются в прикладную программу на этапе компиляции.

Для эффективной реализации управления вводом—выводом, побитовой обработки данных, проектирования оптимальных по быстродействию и занимаемой памяти программ пользователю предоставляется возможность включать в текст на языке Фортран также фрагменты программ на языке *Ассемблера*, для чего разработаны соглашения по передаче параметров. Таким образом, ассемблирование выполняется на одном этапе с компиляцией алгоритмического описания.

Система программирования на языке СИ [33]. Широкий набор операций, стандартных функций в библиотеке языка СИ, высокая

эффективность и компактность транслированных программ позволяют использовать его при разработке как прикладного, так и системного программного обеспечения цифровых САУ.

СИ, как и Паскаль МТ-плюс, представляет собой комплекс инструментальных средств, поддерживающих технологию разработки и отладки программ с использованием компиляции (рис. 6.12). Комплекс включает *СИ-компилятор*, транслирующий программу с языка СИ в ассемблерную программу, *СИ-ассемблер*, *библиотекарь*, *редактор связей* и *СИ-отладчик*. Все инструментальные программы вызываются последовательно и используют промежуточные файлы, подготавливаемые на предыдущих этапах. Исходный текстовый файл создается универсальным экраным редактором, применяемым в соответствующей операционной системе инструментальной микроЭВМ.

В языке используется широкий набор специальных операций, а библиотека языка включает около 50 стандартных функций. Эти средства языка обеспечивают лаконичную запись программы. Функции ввода—вывода и обработки строк символов используются, главным образом, для системного программирования на инструментальной микроЭВМ, тригонометрические функции — в научных расчетах. Дополнительные возможности по ускорению выполнения программ обеспечивают использование регистровых переменных (тип *register*). Компактное размещение данных, ускоренный доступ к данным обеспечиваются смесями (*union*) разных типов данных, возможностью обращения по адресу к ОП и адресной арифметикой.

Предусматриваются простые средства сопряжения с функциями, записанными на языке *Ассемблера*. Параметры функции автоматически передаются через стек и результаты возвращаются через общие переменные. Благодаря возможностям побитовой обработки, доступа к данным по адресным ссылкам, связям с ассемблируемыми программами через внешние ссылки в языке СИ эффективно реализуются системные операции управления вводом—выводом, таймером и обработки прерываний. Программы на языке СИ могут быть оформлены в библиотеке стандартных функций с помощью *библиотекаря*.

Для языка СИ существуют реализации компиляторов для всех типов инструментальных ЭВМ, что позволяет выполнять быструю отладку трудоемких алгоритмов вычислений средствами доступных ЭВМ, обеспечивает переносимость и совместимость программ.

Приведенные примеры наиболее распространенных систем программирования, поддерживающих технологию компиляции программ, показывают, что предлагаемые средства универсальны, обеспечивают все этапы технологии, обладают широкими возможностями, позволяют организовать библиотеку прикладных унифицированных модулей. Связь с ассемблерными программами позволяет разработать качественные программы, в том числе специальные системные программы ввода—вывода в цифровых

САУ, обработки прерываний. Вместе с тем, технология компиляции имеет и существенный недостаток, связанный с продолжительностью многоэтапного процесса проектирования, с переходом от редактирования к исполнению и отладке.

Кроме того, применение универсальных транслирующих программ (компиляторов), в которых зафиксировано унифицированное внутреннее представление данных и, соответственно, используются универсальные процедуры преобразования, приводит к увеличению объема программы в кодах по сравнению с программами, подготовленными на языке *Ассемблера*, в 2—3 раза.

Инструментальная система программирования ФОРТ. В системе ФОРТ [95] реализованы возможности интерпретации и компиляции в реальном времени. Благодаря этому устраняются недостатки, свойственные БЭЙСИК-интерпретатору и всем технологическим процессам на основе компиляции. Другие полезные качества комплекса средств ФОРТ при проектировании цифровых САУ — расширяемость, компактность, высокая скорость исполнения, выход на уровень *Ассемблера*.

Все эти качества и новые возможности обеспечиваются оригинальной организационной системой, позволяющей разрабатывать необходимые инструментальные средства в процессе проектирования, выполняя частичную компиляцию и интерпретацию в процессе отладки.

Система ФОРТ может быть использована как резидентное средство для программирования работы цифровой САУ в реальном времени. При этом пользователю предоставляется некоторый язык программирования с интерпретатором. Язык пользователя может быть представлен простыми вызовами—словами или традиционными конструкциями и операторами алгоритмических языков.

ФОРТ используется в виде инструментального комплекса, включающего собственный *экраный редактор* и *Ассемблер*, средства управления стандартными устройствами ввода—вывода, он связан с инструментальной операционной системой и поддерживает целевую компиляцию, позволяющую создавать прикладные ФОРТ-системы программного обеспечения. Особенностью построения инструментальной ФОРТ-системы является то, что все средства компилируются как расширения на основе компактного ядра (до 4 Кбайт ОЗУ), реализованного в системе команд объектной микроЭВМ. Ядро включает такие минимальные средства, как *базовый словарь*, *адресный интерпретатор*, *компилятор*. Уровень программирования близок к машинному коду. Для перехода на более высокий уровень создается стандартное расширение ФИГ-ФОРТ (до 6 Кбайт ОЗУ) и ФОРТ-83 (до 8 Кбайт ОЗУ).

Список стандартных средств включает арифметико-логические операции и отношения. Данные могут быть представлены байтами, 16-разрядными и 32-разрядными двоичными кодами. Операции могут выполняться (интерпретироваться) после ввода или после ввода и компиляции определений новых функций, включающих

эти операции и ранее определенные функции. Управляющие конструкции {do ... loop, if ... then, repeat ...} исполняются только после компиляции. В отличие от традиционных методов, компиляция может выполняться частично. Таким образом, ФОРТ предоставляет более гибкие средства разработки и отладки программ в интерактивном режиме, чем язык БЭЙСИК.

При создании новых классов объектов используются определения, которые включают две фазы: фазу исполнения (интерпретации) и фазу компиляции. В первой фазе, например, выделяется память для объектов, во второй — компилируются функции, которые исполняются при обращении к новому объекту. Таким образом могут быть введены понятия переменной (*variable*), константы (*constant*), массива (*array*) и другие.

Новые функции, идентифицируемые словами, компилируются в виде линейного списка адресных ссылок и параметров, который называют «шитым кодом» (эквивалент программы). Для исполнения этой программы используется адресный интерпретатор «шитого кода». «Шитый код» связан также со списком слов, сохраняемым после компиляции определений функций. Этот список называется *словарем*. Пользователь может по своему усмотрению использовать несколько *словарей* (как бы несколько библиотек), что позволяет изменить содержание (исполнение) функций. Например, в одном словаре плюс (+) соответствует целочисленной арифметике, а в другом словаре — арифметике с плавающей точкой.

Программирование в системе ФОРТ не традиционно, что, с одной стороны, требует высокой квалификации от разработчика системы управления, работающего в традиционном стиле, а с другой стороны, легко осваиваемо и дает ряд преимуществ.

Одним из важных достоинств является возможность расширения системы ФОРТ до уровня развитой инструментальной системы, включающей редактирование, управление файлами и интерпретацию, как эффективный метод отладки. Свойственная системе ФОРТ расширяемость эквивалентна библиотекам в других системах программирования. Возможность программирования на уровне языка *Ассемблера* позволяет создавать драйверы специальных устройств ввода—вывода в САУ. Для организации входа в прерывания используется обычное программирование в машинном коде, а обработку прерываний можно организовать средствами ФОРТа.

Инструментальные операционные системы. Операционные системы позволяют организовать программное обеспечение инструментальных комплексов в виде связанного технологического процесса проектирования и отладки цифровых САУ. Если инструментальные средства представлены кросс-системой, то для их организации используются стандартные операционные системы ЭВМ общего назначения (ОС ЕС, ДОС КП СМ, UNIX), которые обеспечивают мультипрограммный режим работы систем коллективного пользования.

Операционные системы для микроЭВМ могут быть ориентированы только на *однопользовательское* применение (ДОС СМ-1800, ISIS-II, МИКРОДОС, ОС СР/М, РАФОС). Ядро такой ОС объемом до 8 Кбайт позволяет вызвать разнообразные обслуживающие и инструментальные программы, организовать хранение файлов во внешней памяти.

Многопользовательские ОС (ДИАМС, МОС СМ/4, МР/М) позволяют обслуживать несколько терминалов, что дает возможность распараллелить проектирование и отладку цифровых САУ. Однако для работы этих систем необходимы значительные ресурсы, не всегда доступные в серийных микроЭВМ.

Файл является стандартным способом организации и хранения различных текстов, инструментальных программ, результатов всех преобразований прикладных программ в инструментальной операционной системе. В виде файлов представлены в ОС все программные средства — компиляторы, редакторы, отладчики. Каждому файлу присваивается уникальное имя, которое вместе с атрибутами (адрес на накопителе, объем занимаемой памяти, признаки доступности и др.) хранится в каталоге и занимает фиксированное место во внешней памяти. Файловая подсистема ОС обеспечивает стандартный доступ по имени к файлам для обработки их инструментальными и обслуживающими программами, печати, редактирования, исполнения.

Закрытые ОС ориентированы на стандартные конфигурации технических средств инструментальных систем, перенастройка либо не возможна, либо требует высокой квалификации (ДОС СМ-1800, ISIS-II, РАФОС).

Открытые ОС предполагают возможность создания собственной конфигурации технических средств, настройки ОС путем доработки драйверов по стандартным рекомендациям. Такие ОС адаптированы для использования с внешней памятью на КНМЛ, НГМД, МД, ЦМД и применяются как стандартное системное программное обеспечение в цифровых САУ.

Широко распространены открытые операционные системы, совместимые с ОС СР/М, которая принята как стандартная для микроЭВМ с системой команд МП КР580 и КР1810. Функционально совместима, например, с ОС СР/М операционная система МИКРОДОС для СМ-1800. В СР/М используется большой набор инструментальных программ, в частности, несколько версий интерпретаторов и компиляторов с языка БЭЙСИК, система Паскаль МТ-плюс, системы программирования на языках СИ и Фортран, система ФОРТ и другие.

Операционная система СР/М-80 [87] разделяется на три части: базовая система ввода—вывода BIOS; базовая дисковая операционная система BDOS и интерпретатор команд консоли CCP. BIOS включает драйверы устройств ввода—вывода, в частности, обеспечивает ввод символа с клавиатуры, вывод символа на дисплей и печать, чтение и запись сектора НГМД. В этой части

пользователь может генерировать специальные драйверы для ввода—вывода САУ. BDOS является ядром ОС и обеспечивает стандартные обращения к вводу—выводу и функциям, необходимым для работы с файлами. CCP реализует резидентные команды (вывести каталог, записать файл, распечатать файл) и передает управление транзитным командам, которые вызывают соответствующие файлы из внешней памяти в выделенную транзитную область ОЗУ для исполнения. К транзитным командам относятся *Ассемблер, загрузчик, отладчик, редактор*, копирование и другие средства, входящие в инструментальные системы программирования.

Для ОС CP/M предлагается специальная технология генерации новой версии, учитывающая ограничения памяти и особенности организации ввода—вывода проектируемой микропроцессорной системы. Операционная система может быть использована как в инструментальных микроЭВМ, так и в системах управления. В САУ ОС может быть размещена в ПЗУ вместе с прикладными программами и вызывается в ОЗУ для исполнения автоматически при включении питания.

Разработана «кремниевая» ОС CP/M-86 в виде модуля i80150, который подключается к микропроцессору i8086 и, кроме стандартных функций инструментальной ОС, включает средства для работы с таймерами и обработки прерываний, необходимые в САУ. Прикладные программы в САУ могут быть загружены по сетевому каналу, из резидентного ПЗУ, КНМЛ и других типов доступных запоминающих устройств.

Средства комплексной отладки цифровых систем управления. Качественная отладка ПО цифровой САУ может быть выполнена только в комплексе с аппаратурой в реальном времени. С этой целью используются аппаратно-программные средства *эмуляции* в прототипной инструментальной системе. Необходимость использования этого метода отладки диктуется следующими особенностями организации и работы микропроцессорных цифровых САУ: 1) специальным, ориентированным на применение, распределением адресного пространства, в котором выделены области ОЗУ, ППЗУ, СПЗУ, ВВОДА, ВЫВОДА (п. 5.5); 2) применением унифицированного программного обеспечения, в том числе мониторов, реализуемых в ППЗУ, СПЗУ; операционных систем реального времени (п. 6.1); 3) работой САУ в реальном времени с использованием таймеров, прерываний, многозадачной ОС РВ, прямого доступа к ОП (п. 5.2, 6.1); 4) применением унифицированных БИС микропроцессорных наборов с программным внешним доступом к ресурсам БИС: регистрам, флагам, портам ввода—вывода (п. 5.5); 5) программным управлением вводом—выводом, включающим предварительное программирование конфигурации системы, интерфейсных БИС для сопряжения с УСО (п. 5.3); 6) мультипроцессорной и мультимашинной организацией и использованием мультипроцессорной ОС РВ (п. 6.1).

Таким образом, ошибки в программах, неисправности аппаратных средств, несогласованность интерфейсов весьма разнообразны и, многократно трансформируясь, приводят к непрогнозируемым последствиям. Отладка сложных цифровых САУ требует высокой квалификации в ряде смежных технических дисциплин. В настоящее время в связи с ростом сложности систем и трудоемкости их отладки ведутся интенсивные исследования средств автоматизации, которые в комплексе должны быть выполнены на уровне экспертных систем.

Общая методика отладки аппаратно-программных комплексов предполагает возможность доступа к необходимой и достаточной диагностической информации, которая позволяет локализовать последовательно неисправность с точностью до команды, константы, адреса, цифрового канала, соединения и т. д.

Отладка может быть разделена на несколько этапов, на каждом из которых используется свой набор инструментальных средств.

1. Этап отладки алгоритмов цифрового управления может быть выполнен на любой инструментальной системе с использованием программных имитаторов внешних устройств. Для сохранения преемственности можно использовать распространяемый алгоритмический язык, который имеет реализацию в доступной прототипной системе.

2. Этап отладки аппаратных средств цифровой САУ включает статическую и динамическую проверку (в статике проверяются потенциалы, электрические соединения, работоспособность элементов). Динамическая проверка позволяет выявить основной процент неисправностей в аппаратуре. В простейшем случае используется собственный генератор синхросигналов, отладочные ПЗУ, в которых записаны тестовые программы. Для локализации неисправностей информация о состоянии аппаратуры снимается в контрольных точках и визуализируется на осциллографе, а с помощью сигнатурных анализаторов — на дисплеях [54]. Анализ этой информации и диагностика неисправностей могут быть автоматизированы с использованием анализаторов, выполненных на основе микроЭВМ.

3. Комплексная отладка аппаратно-программных средств предполагает, что алгоритмы управления отлажены и транслированы на прототипных системах в машинный код (этап 1), а технические средства проверены и отлажены (этап 2).

Мониторные функции отлаживаются в первую очередь, так как обеспечивают программный доступ к аппаратуре цифровой САУ. Использование прототипной инструментальной системы позволяет, с одной стороны, применить стандартные средства программирования и отладки, с другой стороны, упрощает разработку специализированных средств, включая программы, тесты и специальное оборудование для отладки. Специальные средства отладки могут предусматривать аппаратную замену (*эмуляцию*)

процессора, памяти, стандартного интерфейса, в том числе системного ИРПР и ИРПС. В прототипной инструментальной системе такая аппаратура всегда существует, что позволяет подключить и отладить управление УСО, реализовать программно-управляемое сопряжение с ОУ (п. 1.4).

Распределение адресного пространства в цифровой САУ и инструментальной системе не совпадают, по этой причине для эмуляции ОП и ввода—вывода используется адресный преобразователь (п. 5.6).

Можно реализовать адресный преобразователь на основе быстрого ОЗУ, что позволяет перепрограммировать распределение адресного пространства оперативно в процессе отладки. С использованием стандартных *загрузчика* и *отладчика* в ОС можно: выполнить загрузку программ цифровой САУ по абсолютным адресам соответствующего адресного пространства; запустить исполнение с остановками в контрольных точках; вывести диагностическую информацию; выполнить коррекцию программы на уровне отдельных команд и констант.

В ряде случаев, в частности, с использованием встроенных микроконтроллеров, стандартные средства прототипной инструментальной системы (ИС) не позволяют получить достаточной диагностической информации. Такая информация доступна, например, в традиционном пошаговом режиме исполнения или связана с динамикой переходных процессов в интерфейсе контроллера, которые невозможно воспроизвести на стандартных интерфейсах ИС. При этом полную отладку можно выполнить только на макете с использованием *внутрисхемных эмуляторов* [3, 54].

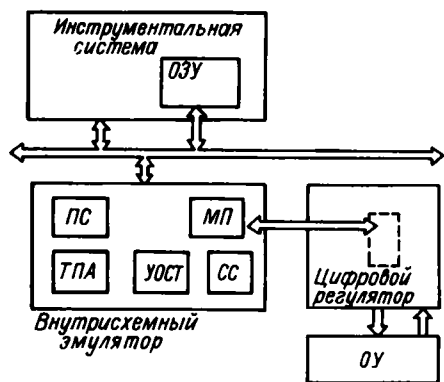


Рис. 6.13. Схема включения внутрисхемного эмулятора

Структура инструментального комплекса приведена на рис. 6.13. В состав комплекса входит ИС с программным обеспечением для управления эмуляцией, которая содержит полностью программно контролируемое ОЗУ, используемое для эмуляции ОЗУ и ПЗУ системы управления. Плата внутрисхемного эмулятора (*ПВЭ*) обеспечивает сопряжение системного интерфейса микропроцессора с системой управления. На плате *ПВЭ* реализована таблица переадресации (*ТПА*), схема управления точками останова (*УОСТ*), схема синхронизации (*СС*), память состояний (*ПС*). Схема синхронизации обеспечивает синхронизацию работы микропроцессора (*МП*), в частности, автоматический и пошаговый режимы и переключает

МП из режима эмуляции в режим выполнения команд системы.

В режиме эмуляции *ПВЭ* является расширением системы управления и обеспечивает управление *УСО* по командам, считываемым из *ОЗУ ИС*. Схема *ТПА* обеспечивает переадресацию в соответствии с заданной таблицей, в памяти состояний фиксируется состояние *МП* (адрес, данные) за несколько текущих тактов. Схема *УОСТ* обеспечивает аппаратное прерывание по заданным условиям для перехода *ПВЭ* в режим выполнения команд инструментальной системы.

Таким образом, с помощью дополнительной платы *ПВЭ* инструментальная система осуществляет контроль состояния системы управления.

Дополнительное ПО режима отладки с использованием эмуляции обеспечивает разнообразные режимы управления и доступ к более детальной информации о работе системы управления, чем в стандартных *отладчиках*.

Различают следующие команды: *ИС*, обеспечивающие настройку *ПВЭ*; управления отладкой; обращения к информации о состоянии цифровой *САУ*. Команды первого вида используются для программирования *ТПА*, загрузки исполняемой программы в *ОЗУ ИС*, задают условия остановки эмуляции. Команды второго типа обеспечивают запуск эмуляции-исполнения, останов, пошаговый и автоматический режим, условия продолжения работы. Команды третьего типа позволяют прочитать и изменить содержимое ячеек *ОП* по адресу, осуществить поиск ячейки по совпадению значений, пересылку информации, позволяют прочитать и модифицировать регистры *МП*, вывести состояние *МП* на дисплей в любой удобной форме.

Условия останова эмуляции задаются в командах управления и включают: останов при исполнении команды по адресу, при чтении данных, записи данных по адресу в произвольное место *ОЗУ* или в стек, обращение к каналу ввода—вывода.

Кроме адреса на *ША* и данных на *ШИ* микропроцессора, в памяти состояний *ПВЭ* автоматически могут быть зафиксированы значения двоичных сигналов, снимаемых щупами в заданных контрольных точках цифровой *САУ*. Таким образом, обеспечивается доступ к самой разнообразной информации о состоянии цифровой *САУ* в реальном времени, что позволяет выполнить быструю и качественную отладку.

Инструментальные системы, включающие разнообразные средства автоматизации программирования и отладки, необходимы для проектирования цифровых *САУ* с использованием как встроенных контроллеров на микропроцессорных наборах, так и серийных микроЭВМ. Серийная многоплатная микроЭВМ используется как инструментальная при соответствующей комплектации внешними устройствами и ПО, что облегчает реализацию методики отладки с применением эмуляции. Практически в этом случае

прототипная инструментальная система является макетом цифрового регулятора и после отладки цифровой САУ может быть заменена на одноплатную конфигурацию без существенной доработки.

Отладка программного обеспечения цифровой САУ, реализованной в ОС РВ однопроцессорной многозадачной iRMX-80, 86 и многопроцессорной iMMX-800, может быть выполнена специальным *отладчиком* с использованием эмуляции. С помощью *отладчика* обеспечивается доступ к таким сложным объектам как *описатели задач*, *описатели обменников*, можно модифицировать эти описатели и устанавливать необходимые приоритеты прерываний и задач.

Внутрисхемными эмуляторами обеспечиваются такие серийные инструментальные комплексы как СО-04 (для КР580), АРМ2-05 с СМ-1800 (для КР580, КР1810, КР589). Разработаны эмуляторы для однокристалльных микрокомпьютеров и секционированных БИС на основе микроЭВМ «Электроника-60», ДВК-2М и др.

Автоматическая запись отлаженных программ в ППЗУ и СПЗУ является важным завершающим этапом проектирования программного обеспечения цифровых САУ. Инструментальные системы и серийные микроЭВМ обеспечиваются платами универсальных *программаторов*, которые включают программно управляемые схемы записи и чтения и универсальные разъемы для серийных БИС ППЗУ и СПЗУ.

Программное обеспечение позволяет настроить программатор на определенный тип БИС [3]. При этом учитывается длина слова ЗУ, доступный объем памяти одного элемента, режимы управления. Запись информации осуществляется последовательно с инверсией или без инверсии, с подсчетом контрольной суммы. После записи всегда можно проверить исправность элемента памяти повторным считыванием и сравнением контрольных сумм. Применяется также считывание с сравнением по образцу, которое позволяет найти неисправную ячейку.

Приведенный обзор методов и средств разработки и отладки программ показывает, что наиболее перспективными являются прототипные инструментальные системы и реализуемые на них инструментальные средства. Программное обеспечение инструментальных систем постоянно расширяется, развивается аппаратура и методика комплексной отладки микропроцессорных систем, которые эффективно могут быть использованы и в цифровых САУ.

Приложение 1

НЕОБХОДИМЫЕ УСЛОВИЯ ОПТИМАЛЬНОСТИ

Доказательство теоремы 4.2. По теореме Гамкрелидзе [49] необходимым условием оптимальности по быстрдействию решений задачи 4.1 (п. 4.1), целиком лежащих на границе, является существование ненулевой вектор-функции $p(t)$ и функции $\lambda(t)$ таких, что:

$$\dot{p} = - \left(\frac{\partial H}{\partial x} \right)^T + \lambda \frac{\partial}{\partial x} \left[\frac{\partial \varphi}{\partial x} f + \frac{\partial \varphi}{\partial x} Gu \right]; \quad (\text{П1.1})$$

$$\frac{\partial H}{\partial u} = \lambda \frac{\partial}{\partial u} \left[\frac{\partial \varphi}{\partial x} f + \frac{\partial \varphi}{\partial x} Gu \right] + \sum_{j=1}^h v_j \frac{\partial}{\partial u} (|u_j| - 1); \quad (\text{П1.2})$$

$$\max_{\substack{u \in \Gamma \\ u: \dot{x}=0}} H = \mu \equiv 0, \quad (\text{П1.3})$$

$$d\lambda/dt \leq 0, \quad (\text{П1.4})$$

где v_j — вещественные числа, удовлетворяющие условию

$$v_j (|u_j| - 1) = 0. \quad (\text{П1.5})$$

Найдем $\lambda(t)$. Для этого вычислим частные производные в (П1.2), учитывая (П1.5) и преобразование управления

$$u = E_1 \bar{u} + E_2 V, \quad (\text{П1.6})$$

где $E_2 = \underbrace{[0 \dots 0]_{1 \times l}}_{l-1} \dots [0]_{1 \times 1}^T$.

Получим

$$p^T G = \lambda \frac{\partial \varphi}{\partial x} G + [\pm v_1 \pm v_2 \dots \pm v_{l-1} 0 \pm v_{l+1} \dots \pm v_h].$$

Умножая справа на E_2 , найдем

$$p^T G E_2 = \lambda \frac{\partial \varphi}{\partial x} G E_2,$$

и, следовательно,

$$\lambda = \frac{(G E_2)^T}{\beta} p = \frac{g_l^T}{\beta} p,$$

что совпадает с (4.34). Поэтому из (П1.4) следует условие в теореме 4.2.

Подставляя (4.34) и (П1.6) в (П1.1), получим уравнение (4.33). Тогда из (П1.3) при u , удовлетворяющем (П1.6) и, следовательно, тождеству $\dot{e} = 0$, получим условия a, b теоремы 4.2.

Нелинейное преобразование канонической модели. Рассмотрим общий случай, когда гамильтониан задается в выражением

$$H = H(x, p, u),$$

где

$$\dot{x} = \left(\frac{\partial H}{\partial p} \right)^T; \quad (\text{П1.7})$$

$$\dot{p} = - \left(\frac{\partial H}{\partial x} \right). \quad (\text{П1.8})$$

Пусть для преобразованной модели $\hat{H} = H$, где

$$\hat{H} = \hat{H}(z, q, u);$$

$$\dot{z} = \left(\frac{\partial \hat{H}}{\partial q} \right)^T; \quad (\text{П1.9})$$

$$\dot{q} = - \left(\frac{\partial \hat{H}}{\partial z} \right)^T. \quad (\text{П1.10})$$

Покажем, что преобразование

$$z = \xi(x); \quad (\text{П1.11})$$

$$p = \left(\frac{\partial \xi}{\partial x} \right)^T q, \quad (\text{П1.12})$$

где $\xi(x)$ — гладкая функция и $\det \partial \xi / \partial x \neq 0$, удовлетворяет уравнениям (П1.9), (П1.10).

В силу свойств $\xi(x)$ существует обратное преобразование

$$x = \gamma(z) \quad (\text{П1.13})$$

и

$$q = \left(\frac{\partial \xi}{\partial x} \right)^{-T} p = \left(\frac{\partial \gamma}{\partial z} \right)^T p. \quad (\text{П1.14})$$

Подставляя (П1.13) и (П1.12), получим

$$\hat{H} = H(\gamma(z), \left(\frac{\partial \xi}{\partial x} \right)^T q, u). \quad (\text{П1.15})$$

Найдем \dot{z} , дифференцируя (П1.11) по времени. С учетом (П1.7) получим

$$\dot{z} = \frac{\partial \xi}{\partial x} \dot{x} = \frac{\partial \xi}{\partial x} \left(\frac{\partial H}{\partial p} \right)^T.$$

Дифференцируя (П1.15) по q , найдем

$$\left(\frac{\partial \hat{H}}{\partial q} \right)^T = \frac{\partial \xi}{\partial x} \left(\frac{\partial H}{\partial p} \right)^T$$

и, следовательно, (П1.9) справедливо.

Докажем (П1.10). Дифференцируя (П1.12), получим

$$\dot{p} = \left(\frac{d}{dt} \frac{\partial \xi}{\partial x} \right)^T q + \left(\frac{\partial \xi}{\partial x} \right)^T \dot{q}.$$

Тогда с учетом (П1.8) и (П1.7) можно записать

$$\begin{aligned} \left(\frac{\partial \zeta_0}{\partial \mathbf{x}}\right)^T \dot{\mathbf{q}} &= -\left(\frac{\partial H}{\partial \mathbf{x}}\right)^T - \left(\frac{d}{dt} \frac{\partial \zeta_0}{\partial \mathbf{x}}\right)^T \mathbf{q} = \\ &= -\left(\frac{\partial H}{\partial \mathbf{x}}\right)^T - \frac{\partial}{\partial \mathbf{x}} \left[\left(\frac{\partial \zeta_0}{\partial \mathbf{x}}\right)^T \mathbf{q} \right] \left(\frac{\partial H}{\partial \mathbf{p}}\right)^T. \end{aligned}$$

Поэтому

$$\begin{aligned} \dot{\mathbf{q}} &= -\left(\frac{\partial \gamma}{\partial \mathbf{z}}\right)^T \left(\frac{\partial H}{\partial \mathbf{x}}\right)^T - \left(\frac{\partial \gamma}{\partial \mathbf{z}}\right)^T \frac{\partial}{\partial \mathbf{x}} \left[\left(\frac{\partial \zeta_0}{\partial \mathbf{x}}\right)^T \mathbf{q} \right] \left(\frac{\partial H}{\partial \mathbf{p}}\right)^T = \\ &= -\left\{ \frac{\partial H}{\partial \mathbf{x}} \frac{\partial \gamma}{\partial \mathbf{z}} + \frac{\partial H}{\partial \mathbf{p}} \frac{\partial}{\partial \mathbf{z}} \left[\left(\frac{\partial \zeta_0}{\partial \mathbf{x}}\right)^T \mathbf{q} \right] \right\}^T. \end{aligned}$$

Сравнивая последнее выражение с частной производной (П1.15) по \mathbf{z} , получим (П1.10).

Доказательство условия α теоремы 4.3. Неравенство (4.47) получено из условия α теоремы 4.2. Дифференцируя (4.34) по времени, найдем

$$\dot{\lambda} = \left[\frac{\partial}{\partial \mathbf{x}} \left(\frac{\mathbf{G}\mathbf{E}_2}{\beta} \right) \dot{\mathbf{x}} \right]^T \mathbf{p} + \frac{\mathbf{G}\mathbf{E}_2}{\beta} \dot{\mathbf{p}}.$$

Подставляя $\dot{\mathbf{x}}$ из (4.31) и $\dot{\mathbf{p}}$ из (4.33), получим

$$\dot{\lambda} = \left\{ \left[\frac{\partial}{\partial \mathbf{x}} \left(\frac{\mathbf{G}\mathbf{E}_2}{\beta} \right) (\mathbf{f}_x + \mathbf{G}_x \bar{\mathbf{u}}) \right]^T - \left[\frac{\mathbf{G}\mathbf{E}_2}{\beta} \left(\frac{\partial \mathbf{f}_x}{\partial \mathbf{x}} + \frac{\partial \mathbf{G}_x}{\partial \mathbf{x}} \bar{\mathbf{u}} \right) \right]^T \right\} \mathbf{p}$$

или

$$\begin{aligned} \dot{\lambda} &= \left\{ \left[\frac{\partial}{\partial \mathbf{x}} \left(\frac{\mathbf{G}\mathbf{E}_2}{\beta} \right) \mathbf{f}_x - \frac{\mathbf{G}\mathbf{E}_2}{\beta} \frac{\partial \mathbf{f}_x}{\partial \mathbf{x}} \right] + \right. \\ &\left. + \left[\frac{\partial}{\partial \mathbf{x}} \left(\frac{\mathbf{G}\mathbf{E}_2}{\beta} \right) \mathbf{G}_x - \frac{\mathbf{G}\mathbf{E}_2}{\beta} \frac{\partial \mathbf{G}_x}{\partial \mathbf{x}} \right] \bar{\mathbf{u}} \right\}^T \mathbf{p}. \end{aligned}$$

Перейдем к системам координат $\begin{bmatrix} e \\ \bar{\mathbf{x}} \end{bmatrix}$, $\begin{bmatrix} p_e \\ \bar{\mathbf{p}} \end{bmatrix}$. На поверхности S , т. е. при $e \equiv 0$, получим

$$\dot{\lambda} = [\mathbf{d}(\bar{\mathbf{x}}) + \mathbf{D}(\bar{\mathbf{x}}) \bar{\mathbf{u}}]^T \left[\left(\frac{\partial \varphi}{\partial \mathbf{x}}\right)^T \mathbf{p}_e + \left(\frac{\partial \Psi}{\partial \mathbf{x}}\right)^T \bar{\mathbf{p}} \right]_{\mathbf{x}=\Gamma(0, \bar{\mathbf{x}})}. \quad (\text{П1.16})$$

Взяв частные производные правой и левой частей тождества

$$\frac{(\partial \varphi / \partial \mathbf{x}) \mathbf{G}\mathbf{E}_2}{\beta} = 1,$$

нетрудно получить

$$\mathbf{d}^T(\bar{\mathbf{x}}) \left(\frac{\partial \varphi}{\partial \mathbf{x}}\right)^T = 0 \quad \text{и} \quad \mathbf{D}^T(\bar{\mathbf{x}}) \left(\frac{\partial \varphi}{\partial \mathbf{x}}\right)^T = 0.$$

Тогда (П1.5) принимает вид (4.47).

Доказательство теоремы 4.5. Используя условия β теорем 4.1 и 4.2, при $\mathbf{u} = \mathbf{u}^*$ получим

$$H_x(t_{1n} + 0) = H(t_{1n} - 0)$$

или, раскрывая,

$$(\mathbf{f} + \mathbf{G}\mathbf{E}_1 \bar{\mathbf{u}} + \mathbf{G}\mathbf{E}_2 V^+)^T (\mathbf{p}^+ - \mathbf{p}^-) = [\mathbf{G}\mathbf{E}_2 (V^- - V^+)]^T \mathbf{p}^-, \quad (\text{П1.17})$$

где индексы «+» и «-» отличают значения переменных при $t = t_{1n} + 0$ и $t = t_{1n} - 0$.

Принимая во внимание условие скачка в форме (4.49), найдем, что левая часть (П1.17) есть $\dot{e} = 0$ и, следовательно, $(V^- - V^+)^T (GE_2)^T = 0$, т. е. условие *a* справедливо.

Условие *b* получается аналогично, с учетом условия непрерывности (4.49а) в точке $t = t_{out}$.

Условие *в* находится непосредственно из *b* теоремы 4.1 с учетом (4.24).

Доказательство теоремы 4.6. В точке $t = t_{in}$ из (4.49) запишем

$$p^- = p^+ - c \left(\frac{\partial \varphi}{\partial x} \right)^T$$

или, учитывая преобразование (4.43) для p^+ ,

$$p^- = \left(\frac{\partial \Psi}{\partial x} \right)^T \bar{p} + \left(\frac{\partial \Psi}{\partial x} \right)^T (p_e - c). \quad (\text{П1.18})$$

По условию *a* теоремы 4.5 $(GE_2)^T p^- = 0$. Тогда, умножая обе части (П1.18) на $(GE_2)^T$, найдем

$$p_e - c = - \frac{(GE_2)^T (\partial \Psi / \partial x)^T}{(GE_2)^T (\partial \varphi / \partial x)^T} \bar{p} = - \frac{(GE_2)^T (\partial \Psi / \partial x)^T}{\beta} \bar{p}.$$

Подставляя в (П1.18), получим

$$p^- = \left[I - \frac{(\partial \varphi / \partial x)^T (GE_2)^T}{\beta} \right] \left(\frac{\partial \Psi}{\partial x} \right)^T \bar{p} = \Xi \left(\frac{\partial \varphi}{\partial x} \right)^T \bar{p},$$

что доказывает справедливость условия *a*.

В точке $t = t_{out}$, учитывая условие непрерывности (4.49а), из (4.43) получим (4.51). Условия (4.52), (4.53) следуют из теоремы 4.5 (*b*, *в*). Таким образом, условия *b*, *в* теоремы 4.6 справедливы.

Приложение 2

СВОЙСТВА НЕСТАЦИОНАРНЫХ МОДЕЛЕЙ

Доказательство теоремы 4.8. Уравнение (4.123) запишем в виде

$$(-A_0 - \eta I)^T M_0 + M_0 (-A_0 - \eta I) = -(B_0 R_1) (B_0 R_1)^T, \quad (\text{П2.1})$$

где $R_1 R_1^T = R^{-1}$, $\det R_1 \neq 0$.

Управляемость пары матриц $\{A_0 + \eta I, B_0\}$ обеспечивает полную наблюдаемость пары $\{B_0^T, -A_0^T - \eta I\}$ и, следовательно, $\{(B_0 R_1)^T, -A_0^T - \eta I\}$. Кроме того, по условию теоремы $\text{Re } \lambda_i [-A_0^T + \eta I] < 0$, поэтому решение уравнения (4.123) существует и $M_0 > 0$ [8].

Преобразуя (П2.1) к виду

$$M_0^{-1} A_0 + A_0^T M_0^{-1} - M_0^{-1} B_0 R^{-1} B_0^T M_0^{-1} = -2\eta M_0^{-1}$$

и сравнивая с (4.122), получим, что найденное решение $P_0 = M_0^{-1} > 0$, что доказывает условие *a*.

Приведем (4.122) к виду

$$(A_{c0} + \eta I)^T P_0 + P_0 (A_{c0} + \eta) = -P_0 B_0 R^{-1} B_0^T P_0.$$

Так как $P_0 B_0 R^{-1} B_0^T P_0 \geq 0$, то выполняется [8] $\text{Re } \lambda_i [A_{c0} + \eta I] \leq 0$ и условие *b* справедливо.

Доказательство теоремы 4.10. Определим функцию Ляпунова

$$v(e, x) = e^T P(x) e, \quad (П2.2)$$

где в силу теоремы 4.8 $P(x) > 0$.

Дифференцируя (П2.2) по времени, на решениях (4.116) получим

$$\dot{v}(e, x) = e^T [\dot{P}(x) + A_c^T(x) P(x) + P(x) A_c(x)] e.$$

Принимая во внимание (4.122), запишем далее

$$\dot{v}(e, x) = e^T [\dot{P}(x) - 2\eta P(x) - P(x) B(x) R^{-1} B^T(x) P(x)] e. \quad (П2.3)$$

Определим функцию

$$\lambda(x) = \max_t \lambda_t [\dot{P}(x) P^{-1}(x)].$$

Используя (П2.3), найдем

$$\dot{v}(e, x) \leq [-2\eta + \lambda(x)] e^T P(x) e = [-2\eta + \lambda(x)] v(e, x).$$

Принтегрируем последнее неравенство

$$v(e, x) \leq v(e_0, x_0) \exp \int_0^t \lambda(x) d\tau \exp(-2\eta t),$$

где

$$\begin{aligned} \exp \int_0^t \lambda(x) d\tau &= \exp \int_0^t \max_t \lambda_t \left[\frac{\partial P}{\partial x} \frac{dx}{d\tau} P^{-1}(x) \right] d\tau = \\ &= \int_{x_0}^{x(t)} \max_t \lambda_t \left[\frac{\partial P}{\partial x} M(x) \right] dx = \beta_2^2(x) \end{aligned}$$

и, следовательно,

$$v(e, x) \leq v(e_0, x_0) \beta_2^2(x) \exp(-2\eta t). \quad (П2.4)$$

Учитывая, что $v(e, x) = \|e\|_{P(x)}^2$, перейдем к нормам в (П2.4) и получим неравенство (4.137).

Доказательство теоремы 4.11. Известно [42], что

$$\max_{\|e\|_P=c} \frac{\|e\|_{\tilde{P}}}{\|e\|_P} = \left(\max_t \lambda_t [\tilde{P}(x) P^{-1}(x)] \right)^{1/2}. \quad (П2.5)$$

Преобразуем правую часть (П2.5):

$$\begin{aligned} \lambda_t [\tilde{P}(x) P^{-1}(x)] &= \lambda_t [(T^T P_0 T) (T_1^T \Lambda T_1)^{-1}] = \\ &= \lambda_t [\Lambda^{-1/2} (T_1 T^T) P_0 (T T_1^T) \Lambda^{-1/2}] = \lambda_t [\Lambda^{-1/2} (T_{10} T T_1)^T \Lambda_0 (T_{10} T T_1) \Lambda^{-1/2}]. \end{aligned}$$

Так как собственные числа матриц Λ и Λ_0 упорядочены, то [50]

$$\begin{aligned} \beta_1^2(x) &= \min_{T \in \mathbb{E}} \max_t \lambda_t [\Lambda^{-1/2} (T_{10} T T_1)^T \Lambda_0 (T_{10} T T_1) \Lambda^{-1/2}] = \\ &= \max_t \lambda_t [\Lambda^{-1/2} \Lambda_0 \Lambda^{-1/2}], \end{aligned} \quad (П2.6)$$

причем минимум достигается при $T_{10} T T_1 = T$ или, учитывая ортогональность матриц, при $T = T_{10}^T T_1$.

Принимая во внимание, что

$$\lambda_t [\Lambda^{-1/2} \Lambda_0 \Lambda^{-1/2}] = \lambda_t [\Lambda_0 \Lambda^{-1}] = \lambda_t [P_0] / \lambda_t [P(x)], \quad (П2.7)$$

и подставляя (П2.7) в (П2.6), получим (4.144).

СПИСОК ЛИТЕРАТУРЫ

1. Автоматизированное проектирование дискретных регуляторов/А. В. Богачев, В. В. Григорьев, В. Н. Дроздов и др. Л.: ЛДНТП, 1981. 23 с.
2. Алексеев Ю. Д., Бойченко А. В. Периферийные устройства мини- и микроЭВМ//Микропроцессорные средства и системы, 1985, № 4. С. 250.
3. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах. М.: Радио и связь, 1984. 269 с.
4. Андреев Ю. Н. Управление конечномерными линейными объектами. М.: Наука, 1976. 424 с.
5. Артвик Б. А. Сопряжение микроЭВМ с внешними устройствами. М.: Машиностроение, 1983. 343 с.
6. Атанс М., Фалб П. Оптимальное управление. М.: Машиностроение, 1968. 764 с.
7. Балашов Е. П., Григорьев В. Л., Петров Г. А. Микро- и мини-ЭВМ. Л.: Энергоатомиздат, 1984. 376 с.
8. Бахилина Н. М., Лернер Д. М. Свойства уравнений Ляпунова с неотрицательной матрицей свободных членов//Автоматика и телемеханика, 1978. № 5. С. 182—184.
9. Болтунов Г. И., Мирошник И. В. Метод управления пространственным движением//Системы управления и их элементы. Труды ЛИТМО. Л., 1981. С. 43—47.
10. Брайсон А., Хо-Ю-Ши. Прикладная теория оптимального управления. М.: Мир, 1972. 544 с.
11. Бутомо Н. Д., Самочадин А. В. Программирование на алгоритмическом языке Паскаль для микроЭВМ. Л.: Изд-во ЛГУ, 1985. 214 с.
12. Вегнер П. Программирование на языке Ада: Пер. с англ. М.: Мир, 1983. 240 с.
13. Вельбицкий И. В. Современное состояние и перспективы развития Р-технологии программирования. Киев: О-во «Знание» УССР, 1984. 22 с.
14. Воронов А. А. Введение в динамику сложных управляющих систем. М.: Наука, 1985. 352 с.
15. Гантмахер Ф. Р. Теория матриц. М.: Наука, 1966. 840 с.
16. Гибкие автоматизированные производства/Под ред. С. А. Майорова. Л.: Машиностроение, 1983. 376 с.
17. Гибсон Г., Лео Ю.-И. Аппаратные и программные средства микроЭВМ. М.: Финансы и статистика, 1983. 250 с.
18. Голубев К. П., Григорьев В. В., Дроздов В. Н. Алгоритмы определения вынужденных движений в линейных многомерных дискретных системах//Автоматика и телемеханика, 1981. № 10. С. 83—89.
19. Голубев К. П., Григорьев В. В., Дроздов В. Н. Синтез линейных дискретных комбинированных систем//Приборостроение, 1981. Т. 23. № 10. С. 25—29.
20. Голубев К. П., Дроздов В. Н. Аналитическое конструирование дискретных регуляторов для линейных объектов//Сложные системы управления. Л.: Изд-во ЛГУ, 1985. Вопросы теории автоматического управления. Вып. 6. С. 82—92.

21. Григорьев В. В., Коровьяков А. Н. Алгоритмы вычисления установившихся ошибок многомерных линейных систем//Автоматика и телемеханика, 1980. № 3. С. 13—19.
22. Дроздов В. Н., Козярук А. Е., Мирошник И. В. Системы управления электроприводом с использованием микроЭВМ. Л.: ЛДНТП, 1984. 24 с.
23. Дроздов В. Н., Мирошник И. В. Оптимальное регулирование скорости нелинейного электропривода//Электричество, 1986. № 10. С. 53—56.
24. Заде Л., Дезоер Ч. Теория линейных систем. М.: Наука, 1970. 703 с.
25. Зеленко Г. В., Попов С. Н. Проектирование микропроцессорных устройств на базе МП КР1810ВМ86. М.: МЦНТИ, 1986. 120 с.
26. Иванов В. А., Фалдин Н. В. Теория оптимальных систем автоматического управления. М.: Наука, 1981. 336 с.
27. Иванов Г. М., Левин Г. М., Хуторецкий В. М. Автоматизированный многодвигательный электропривод постоянного тока. М.: Энергия, 1978. 160 с.
28. Изерман Р. Цифровые системы управления. М.: Мир, 1984. 541 с.
29. Икрамов Х. Д. Численное решение матричных уравнений. М.: Наука, 1984. 192 с.
30. Ильин В. А., Поздняк Э. Г. Основы математического анализа. Часть 1. М.: Наука, 1982. 616 с.
31. Казаков И. Е. Статистическая динамика систем с переменной структурой. М.: Наука, 1977. 416 с.
32. Квакернаак Х., Сиван Р. Линейные оптимальные системы управления. М.: Мир, 1977. 650 с.
33. Керниган Б., Ритчи Д. Язык программирования СИ. Задачи по языку СИ: Пер. с англ. М.: Финансы и статистика, 1985. 279 с.
34. Ключев А. С., Колесников А. А. Оптимизация автоматических систем управления по быстродействию. М.: Энергоиздат, 1982. 240 с.
35. Кобылинский А. В., Горячев А. В., Сабадаш Н. Г. Семейство ЭВМ «Электроника К1». М.: Высшая школа. 1988. 191 с.
36. Крылов Е. И. Однокристальные микроЭВМ серии 1814, 1820. К1816// Микропроцессорные средства и системы, 1985. № 2. С. 25—27.
37. Кузнецов С. О., Прохоров Н. П., Раев В. К. Электронный диск на цилиндрических магнитных дисках//Микропроцессорные средства и системы, 1985. № 4. С. 36—39.
38. Кузовков Н. Т. Модальное управление и наблюдающие устройства. М.: Машиностроение, 1976. 184 с.
39. Кунцевич В. М. О синтезе оптимальных дискретных систем управления при постоянно действующих возмущениях//Автоматика и телемеханика, 1973. № 4. С. 46.
40. Кунцевич В. М., Лычак М. М. Синтез систем автоматического управления с помощью функций Ляпунова. Л.: Наука, 1977. 399 с.
41. Кунцевич В. М., Лычак М. М. Синтез систем оптимального управления, регулируемого на УВМ, и выбор оптимальных частот квантования во времени// Автоматика и телемеханика, 1980. № 5. С. 57—63.
42. Ланкастер П. Теория матриц. М.: Наука, 1978. 280 с.
43. Ларионов А. М., Майоров С. А., Новиков Г. И. Вычислительные комплексы, системы и сети. Л.: Энергоатомиздат, 1987. 288 с.
44. Летов А. М. Динамика и управление. М.: Наука, 1979. 488 с.
45. Лившиц Н. А., Виноградов В. Н., Голубев Г. А. Корреляционная теория оптимального управления многомерными процессами. М.: Сов. радио. 1974. 328 с.
46. Лю Ю-Чжен, Гибсон Г. Микропроцессоры семейства 8086/8088. М.: Радио и связь. 1987. 512 с.
47. Мак-Кракен Д., Дорн У. Численные методы и программирование на ФОРТРАНе. М.: Мир, 1969. 582 с.
48. Малашевич Б. М., Джунян В. Л. Магистрально-модульные микропроцессорные системы// Микропроцессорные средства и системы, 1984, № 4. С. 15—18.
49. Математическая теория оптимальных процессов/Л. С. Понтрягин, В. Г. Болтянский, Р. В. Гамкрелидзе и др. М.: Наука, 1983. 392 с.

50. Маркус М., Мишк Х. Обзор по теории матриц и матричных неравенств. М.: Наука, 1972. 232 с.
51. Медич Д. Статистически оптимальные линейные оценки и управление. М.: Энергия, 1973. 440 с.
52. Месарович М., Мако Д., Такахара И. Теория иерархических многоуровневых систем. М.: Мир, 1973. 344 с.
53. Мик Дж., Брик Дж. Проектирование микропроцессорных устройств с разрядно-модульной организацией. М.: Мир, 1984. Т. 1. 253 с.; т. 2. 223 с.
54. Микропроцессоры: системы программирования и отладки/В. А. Мясников, М. Б. Игнатьев, А. А. Кочкин, Ю. Е. Шейнин. М.: Энергоатомиздат, 1985. 272 с.
55. МикроЭВМ СМ-1800. М.: Финансы и статистика, 1984. 230 с.
56. Мирошник И. В., Ушаков А. В. Синтез алгоритма синхронного управления системой квазиоднотипных объектов//Автоматика и телемеханика, 1977. № 11. С. 22—29.
57. Мирошник И. В. Свойства управлений с заданной оценкой переходного процесса//Автоматика и телемеханика, 1982. № 4. С. 16—29.
58. Мирошник И. В. Алгоритмы управления мультиструктурных электромеханических систем//Электромеханика, 1986. № 3. С. 51—58.
59. Мирошник И. В. О стабилизации движения по многообразию//Автоматика, 1986. № 4. С. 65—68.
60. Мирошник И. В., Болтунов Г. И. Оптимальные микропроцессорные системы управления электроприводом. Л.: ЛДНТП, 1986. 28 с.
61. Мишель Ж., Лорже К., Эспьо Б. Программируемые контроллеры. М.: Машиностроение, 1986. 172 с.
62. Мищенко А. С., Фоменко А. Г. Курс дифференциальной геометрии и топологии. М.: Изд-во МГУ, 1980. 439 с.
63. Олейников В. А., Борисенко Р. А. Асимптотические свойства фазовых траекторий и особые управления в оптимальных быстродействиях/Л.: Изд-во ЛГУ, 1974. Вопросы теории систем автоматического управления. Вып. 1. С. 39—42.
64. Операционные системы реального времени для микроЭВМ/Под ред. И. Р. Крамфуса. М.: МЦНТИ, 1984. 133 с.
65. Основы теории вычислительных систем/Под ред. С. А. Майорова. М.: Высшая школа, 1978. 418 с.
66. Первозванский А. А., Гайцгори В. Г. Декомпозиция, агрегирование и приближенная оптимизация. М.: Наука, 1979. 375 с.
67. Питерсон Дж. Теория сетей Петри и моделирование систем. М.: Мир, 1984. 260 с.
68. Полупроводниковые БИС запоминающих устройств/В. В. Баранов, Н. В. Беккин, А. Ю. Гордонов и др. М.: Радио и связь, 1986. 360 с.
69. Построение систем программного движения/А. С. Галиуллин, И. А. Мухаметзянов, Р. Г. Мухарлямов и др. М.: Наука, 1971. 342 с.
70. Праггшвилл И. В. Микропроцессоры и локальные сети микроЭВМ в распределенных системах управления. М.: Энергоатомиздат, 1985. 272 с.
71. Проектирование регуляторов с использованием ЭВМ/В. В. Григорьев, В. Н. Дроздов, В. В. Лаврентьев и др.//Системы управления и их элементы. Труды ЛИТМО. Л.: ЛИТМО, 1981. С. 3—19.
72. Программируемый микропроцессорный набор. Методическое пособие/В. И. Скорубский, Г. С. Подклетнов, А. Е. Платунов и др. Л.: ЛИТМО, 1984. 87 с.
73. Ривкин С. С. Теория гироскопических устройств. Ч.1. Л.: Судостроение, 1962. 507 с.
74. Самонастраивающиеся системы. Справочник/Под ред. П. И. Чинаева. Киев: Наукова Думка, 1969. 490 с.
75. Синтез дискретных регуляторов при помощи ЭВМ/В. В. Григорьев, В. Н. Дроздов, В. В. Лаврентьев и др. Л.: Машиностроение, 1983. 245 с.
76. Скорубский В. И., Троицкая М. П. Арифметические основы ЭВМ. Учебное пособие. Л.: ЛИТМО, 1979. 91 с.

77. Солодовников В. В., Плотников В. И., Яковлев А. В. Основы теории и элементы систем автоматического регулирования. М.: Машиностроение, 1985. 536 с.
78. Талов И. Л., Соловьев А. Н., Борисенков В. Д. Семейство ЭВМ «Электроника-60». М.: Высш. школа, 1988. 172 с.
79. Темников Ф. Е., Поронин В. А., Дмитриев В. И. Теоретические основы информационной техники. М.: Энергия, 1971. 424 с.
80. Тимофеев А. В. Построение адаптивных систем управления программным движением. Л.: Энергия, 1980. 88 с.
81. Ткаченко А. П. Судовые системы автоматического управления и регулирования. Л.: Судостроение, 1984. 288 с.
82. Томович Р., Вукобратович М. Общая теория чувствительности. М.: Сов. радио, 1972. 239 с.
83. Ту Ю. Современная теория управления. М.: Машиностроение, 1971. 472 с.
84. Уонем М. Линейные многомерные системы управления: Геометрический подход. М.: Наука, 1980. 376 с.
85. Управляющие и вычислительные устройства роботизированных комплексов, ч. I: Учебное пособие для вузов/Е. П. Балашов, М. Б. Игнатьев, М. С. Куприянов и др. Л.: ЛИАП, 1984. 280 с.
86. Уткин В. И. Скользящие режимы и их применение в системах с переменной структурой. М.: Наука, 1974. 240 с.
87. Уэйт М., Ангермейер Дж. Операционная система СР/М: Пер. с англ. М.: Радио и связь, 1986. 312 с.
88. Фаулджер Р. Программирование встроенных микропроцессоров. М.: Мир, 1985. 250 с.
89. Федорков Б. Г., Телец В. А., Дегтяренко В. П. Микроэлектронные цифроаналоговые и аналого-цифровые преобразователи. М.: Радио и связь, 1984. Сер. Электроника; Вып. 41. 150 с.
90. Филиппов Л. Г., Фрейдзон И. Р. Мини- и микроЭВМ в управлении промышленными объектами. Л.: Машиностроение, 1984. 336 с.
91. Френк Л. Теория сигналов. М.: Сов. радио, 1974. 440 с.
92. Фурасов В. Д. Устойчивость движения, оценки, стабилизация. М.: Наука, 1977. 247 с.
93. Хамби Э. Программирование таблиц решений/Пер. с англ. М.: Мир, 1976. 86 с.
94. Хвоц С. Т., Варлинский Н. Н., Попов Е. А. Микропроцессоры и микроЭВМ в системах автоматического управления. Справочник. Л.: Машиностроение, 1987. 640 с.
95. Хротоко Г. Языки программирования высокого уровня для микроЭВМ. М.: МЦНТИ, 1985. 95 с.
96. Цифровое управление многорежимным электроприводом/В. Н. Дроздов, А. Е. Козярук, И. В. Мирошник, Ю. А. Сабинин//Электричество, 1985. № 6. С. 13—18.
97. Чистов В. П., Бондаренко В. И., Святославский В. А. Оптимальное управление электрическими приводами постоянного тока. М.: Энергия, 1968. 232 с.
98. Шевкопляс Б. В. Микропроцессорные структуры. Инженерные решения. М.: Радио и связь, 1986. 264 с.
99. Эйкхофф П. Основы идентификации систем управления. М.: Мир, 1975. 683 с.
100. Якубовский С. В., Барханов Н. А. Аналоговые и цифровые интегральные схемы. М.: Сов. радио, 1984. 432 с.
101. Miroshnik I., Nicosia F., Nicosia S. Robust model reference adaptive control//Ricerche di automatica, 1982. Vol. 13, N. 2. P. 260—268.
102. Johnson C. D. Algebraic solution of the servomechanism problem with external disturbances//Trans. ASME, ser. G, 1974. P. 25—35.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

- Автомат комбинационный 180
Автоматизация проектирования 41
— программирования 254
Адресное пространство 222
Алгоритм программного управления 119
— пространственного управления 151, 184
— оценивания состояния 88
— переключения структуры 19, 177, 182
— управления 16, 19, 66
— — локальный 19, 122
— — рекуррентный 36
— — унифицированный 186
Алгоритмическое обеспечение 34
Аналого-цифровой канал 218
Аппроксимация нелинейной модели 45, 48
— параметров алгоритма управления 177, 186
— переходного процесса 149
— поверхности 151
Арифметико-логический блок 191
- БИС 23
Быстродействие 39, 115
- Вектор возмущения 13
— относительного движения 121, 152
— продольного движения 130, 152
— регулируемых переменных 13, 44
— состояния 13, 44
— — командного генератора 50
— — сопряженной системы 125
— управления 13, 44
Вентиль 23
Время переходного процесса 62, 162
- Гамильтониан 125
Гиперповерхность граничная 115
— оптимального движения 179
— — — вырожденная 139
— переключения 117, 179
Гребная электрическая установка 10
- Движение возмущенное 163
— вынужденное 61
— относительное 121, 152
— продольное 130, 152
— собственное 61
Дискретизация модели 56
— пространства 178
Драйвер 234
- Задача оптимального управления 38, 70, 115, 116
— — — вырожденная 126, 137
— — — локальная 40, 153
— — — нормальная 126
Задачи системы управления 14
Запоминающее устройство 25, 221
- Интегральная схема 23
Интервал квантования 21
— —, выбор 54
Интерпретация 256
Интерфейс 26, 207
- Качество системы управления 15, 38
Команда микропроцессора 197
Командный генератор 50, 52
Критерий оптимальности 38, 70, 115
Кросс—система 33, 255
Конвейер 193
- Линеаризация 48
— на гиперповерхности 154
— на многообразии 159
Линия технологическая 9
Локализация корней (собственных чисел) 71, 164
Локальная сеть микроЭВМ 231
- Масштабирование 251
Магистраль 26
Метод синтеза частотный 35
— пространства состояний 37
Микрокоманда 193
Микрокомпьютер 27

- Микропрограммирование 194
 Микропрограммного управления, блок 190
 Микропроцессор 27, 196
 Микропроцессорный набор 26
 Микрошина 198
 МикроЭВМ одноплатная 27
 Многообразии вырожденное 139
 — оптимальное 40, 118, 179
 — переключения 117, 179
 Модель каноническая 126, 129
 — — редуцированная 132
 — линейного объекта 43, 44
 — нелинейного объекта 13, 48, 114
 — нестационарная 107, 161, 274
 — редуцированная 45
 — чувствительности 163
 Монитор 30
 Мультисистема 29, 225
 — иерархическая 226
 — параллельная 227
 Мультишина 209
- Наблюдатель** 88
 — редуцированный 91
 — ускоренный 94
 — с экстраполяцией 96
 — частичной оценки 101
Накопитель на магнитном диске 225
 — на магнитной ленте 224
Норма отклонения 162, 176
 — — нестационарная 167, 170
 — — — вращения 172
- Область допустимых состояний** 115
 — — управлений 115
 — — знакопостоянства 40, 117
Общая шина 207
Объект управления 7, 43
Ограничения переменных 14, 115
Окрестность многообразия 153, 179
Операционная система инструментальная 264
 — — реального времени 32, 236
Оптимизация системы управления 38, 70, 124
Основная память 27
- Планирующая программа** 120, 186
 — — проектирования 144
Порт 26
Предикат 243
Прерывание 31, 194, 212
Принцип максимума Понтрягина 124
Приоритеты событий 237
Программирование модульное 254
Программное обеспечение 30, 233
 — — внешнее 32
 — — резидентное 32
 — — системное 30, 233
- Прямой доступ к памяти** 214
Погрешности 246
- Расширитель арифметический** 201
Регенерация динамической памяти 224
Регулятор 16
 — грубый 78, 81, 104
 — комбинированный 73, 81, 84
 — локальный 19, 122, 185
 — модальный 67
 — нестационарного объекта 107, 161
 — оптимальный 70
 — релейный 120
Режим системы управления 18, 121, 178
- Синтез алгоритмов управления** 34
Система автоматизированного проектирования 41
 — программного управления 118
 — пространственного управления 119
 — сопряженная 125
 — управления микропроцессорная (цифровая) 20
 — — многоконтурная 17
 — — многорежимная 18, 121
 — — одноконтурная 16
 — — переменной структуры 20, 121
Сопроцессор 201
Стабилизация 14, 75
 — относительного движения 153
 — нестационарного объекта 161
Станок—автомат 7
Степень устойчивости 62, 156, 165
- Таймер** 211
Теорема Гамкрелдзе 129, 271
Траектория оптимальная 40, 116
 — —, расчет 141
- Унификация** 24, 42, 186
Управление модальное 67
 — комбинированное 73
 — программное 14, 118
 — пространственное 119, 151
 — терминальное 14, 39, 114
 — оптимальное 70, 116
Уравнение Ляпунова 164, 169
 — — дискретное 176
 — Сильвестра 64
 — Риккати 164, 169
 — — дискретное 70, 117
 — состояния 13, 43
 — — дискретное 37, 59
Условие квазистационарности 165
 — скачка 133
 — сопряжения 132
 — оптимальности 124, 137, 271
Устойчивость 15, 161
 — экспоненциальная 162

Устройство задающее 18, 118
— запоминающее 25
— оценивания 14
— связи с объектом 20, 216

Функционал качества 70, 115

Функция Ляпунова 275
— передаточная 35, 44
— — дискретная 36
— переключательная автомата 182
— чувствительности 163, 170

Цифро-аналоговый комплекс 33
— канал 217

Экстраполяция 98

Электропривод гребной электрической
установки 10

— постоянного тока 11

— станка—автомата 8

— технологической линии 9

Эмулятор внутрисхемный 268

Эмуляция 266

О Г Л А В Л Е Н И Е

Предисловие	3
Список основных обозначений и сокращений	6
Глава 1. Принципы построения и проектирования систем управления с микроЭВМ	7
1.1. Задачи автоматического управления	—
1.2. Принципы построения систем управления	16
1.3. Технические средства и программное обеспечение микропроцессорных систем управления	22
1.4. Методы синтеза цифровых алгоритмов управления	34
Глава 2. Модели систем автоматического управления	43
2.1. Модели объектов управления	—
2.2. Линейные модели внешних воздействий	50
2.3. Выбор интервала квантования	54
2.4. Разностные модели линеаризованных объектов управления	58
Глава 3. Синтез линейных алгоритмов управления	61
3.1. Собственные и вынужденные движения в дискретной системе	—
3.2. Алгоритмы управления собственным движением	65
3.3. Алгоритмы комбинированного управления	73
3.4. Грубый алгоритм управления вынужденным движением	78
3.5. Совместное использование комбинированного и грубого управления	81
3.6. Синтез комбинированной системы с ненулевой установившейся ошибкой	84
3.7. Алгоритмы оценивания состояний	88
3.8. Специальные алгоритмы оценивания состояний	96
3.9. Грубые регуляторы с экстраполированием	104
3.10. Цифровое управление нестационарными объектами	107
Глава 4. Алгоритмы оптимального управления нелинейных систем	114
4.1. Нелинейные задачи управления	—
4.2. Необходимые условия оптимальности переходных процессов	124
4.3. Анализ вырожденной задачи оптимального управления	137
4.4. Исследование оптимальных переходных процессов	141
4.5. Синтез локальных регуляторов	151
4.6. Стабилизация нестационарных моделей	161
4.7. Синтез системы управления	177
Глава 5. Технические средства микропроцессорных систем управления	190
5.1. Принципы организации вычислительных средств на основе секционированных микропроцессорных наборов	—
5.2. Принципы организации вычислительных средств на основе	

однокристалльных микропроцессоров	195
5.3. Стандартные внешние интерфейсы серийных микроЭВМ. .	205
5.4. Сопряжение микроЭВМ с объектом управления в реальном времени	210
5.5. Устройства сопряжения с объектом	216
5.6. Организация запоминающих устройств	221
5.7. Мультипроцессорные и мультимашинные цифровые системы управления	225
Глава 6. Программное обеспечение микропроцессорных систем управления	233
6.1. Структура системного программного обеспечения цифровых САУ	—
6.2. Программная реализация алгоритмов управления	243
6.3. Методы и средства разработки и отладки программ	254
Приложение 1. Необходимые условия оптимальности	271
Приложение 2. Свойства нестационарных моделей	274
Список литературы	276
Предметный указатель	280
Оглавление	283

Производственное издание

ДРОЗДОВ Валентин Нилович
МИРОШНИК Илья Васильевич
СКОРУБСКИЙ Владимир Иванович

**Системы
автоматического
управления
с микроЭВМ**

Редактор *Н. В. Сергеева*
Переплет художника *В. В. Пожидаева*
Художественный редактор *С. С. Венедиктов*
Технический редактор *Т. П. Малашкина*
Корректоры: *А. И. Лавриненко, Н. Г. Иванова*

ИБ № 5408

Сдано в набор 04.10.88. Подписано в печать 11.07.89. М-29110.
Формат 60×90^{1/16}. Бумага офсетная № 2.
Гарнитура литературная. Печать офсетная.
Усл. печ. л. 18,0. Усл. кр.-отг. 18,0. Уч.-над. л. 18,09.
Тираж 13 000 экз. Заказ 586. Цена 1 р. 20 к.

Ленинградское отделение ордена Трудового Красного Знамени
издательства «МАШИНОСТРОЕНИЕ»
191065, Ленинград, ул. Дзержинского, 10.

Ленинградская типография № 6 ордена Трудового Красного Знамени
Ленинградского объединения «Техническая книга» им. Евгении Соколовой
Союзполиграфпрома при Государственном комитете СССР
по делам издательств, полиграфии и книжной торговли.
193144, г. Ленинград, ул. Моисеенко, 10

**Издательство „Машиностроение“
в 1989 г.
выпускает книги:**

Хвощ С. Т., Дорошенко В. В., Горовой В. В. Организация последовательных мультимплексных каналов систем автоматического управления. — 17,5 л.: ил. — ISBN 5-217-00562-9 (в пер.): 1 р. 30 к.

Рассмотрены вопросы организации систем автоматического управления на основе мультимплексных последовательных каналов межмодульного обмена (МК). Описана структура МК с наиболее распространенной гальванической трансформаторной развязкой. Приведены примеры организации обмена данными в МК. Рассмотрены также вопросы отладки и диагностики систем автоматического управления (САУ), построенных на основе МК. Содержатся данные об отечественной элементной базе, необходимой для реализации устройств, подключаемых к мультимплексным каналам.

Для инженеров, занимающихся цифровой вычислительной техникой, и разработчиков САУ.

ЭВМ в проектировании и производстве: Сб. статей. Вып. 4/Под общ. ред. Г. В. Орловского. — 29 л.: ил. — ISBN 5-217-00553-3 (в пер.): 1 р. 80 к.

В книге рассмотрены вопросы использования современных ЭВМ в проектировании и производстве, включая практические системы с искусственным интеллектом и персональными компьютерами. Отражены вопросы разработки теоретических основ автоматизации проектирования (САПР), автоматизации технологической подготовки производства (АСТПП), автоматизации управления технологическими процессами (АСУТП). Дано описание конкретных систем по этим направлениям. Приведены сведения по организации внедрения САПР, АСТПП и АСУТП.

Для инженерно-технических работников, специализирующихся в области комплексной автоматизации и применения вычислительных машин в проектировании и производстве.

Уважаемый читатель!

С целью получения информации о качестве наших изданий просим Вас в прилагаемой анкете подчеркнуть позиции, соответствующие Вашей оценке этой книги.

1. В книге существует:
 - а) острая необходимость
 - б) значительная потребность
 - в) незначительная потребность
2. Эффективность книги с точки зрения практического вклада в отрасль:
 - а) весьма высокая
 - б) высокая
 - в) сомнительная
 - г) незначительная
3. Эффективность книги с точки зрения теоретического вклада в отрасль:
 - а) весьма высокая
 - б) высокая
 - в) сомнительная
 - г) незначительная
4. Материал книги соответствует достижениям мировой науки и техники в данной отрасли:
 - а) в полной мере
 - б) частично
 - в) слабо
5. Книга сохранит свою актуальность:
 - а) 1—2 года
 - б) в течение 5 лет
 - в) длительное время
6. Название книги отвечает содержанию:
 - а) в полной мере
 - б) частично
 - в) слабо

*Дополнительные замечания
предлагаем Вам приложить отдельно.*

Фамилия, имя, отчество

.

Ученое звание

.

Специальность

.

Место работы, должность

.

Стаж работы

ЛИНИЯ ОТРЕЗА

Просим отрезать страницу по линии отреза
и в почтовом конверте выслать по адресу:

*191065, Ленинград, ул. Дзержинского, 10,
ЛО изд-ва «Машиностроение».*

В. Н. Дроздов, И. В. Мирошник, В. И. Скорубский. Системы автоматического управления с микроЭВМ