



СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

SIBERIAN
FEDERAL
UNIVERSITY

Электронный учебно-методический комплекс

Автоматизированное проектирование средств и систем управления

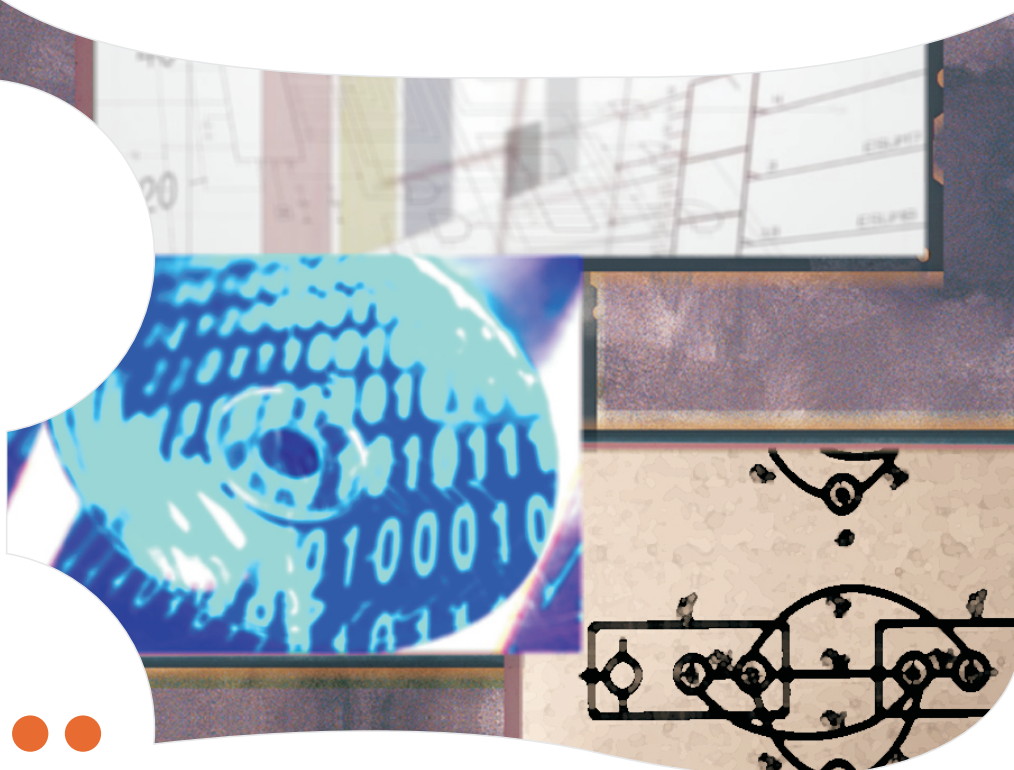
Учебная программа дисциплины

- **Курс лекций**

Лабораторный практикум

Методические указания по самостоятельной работе

Банк тестовых заданий в системе UniTest



Красноярск
ИПК СФУ
2009

УДК 681.5(075)
ББК 32.973я73
А22

Электронный учебно-методический комплекс по дисциплине «Автоматизированное проектирование средств и систем управления» подготовлен в рамках реализации Программы развития федерального государственного образовательного учреждения высшего профессионального образования «Сибирский федеральный университет» (СФУ) на 2007–2010 гг.

Рецензенты:

Красноярский краевой фонд науки;
Экспертная комиссия СФУ по подготовке учебно-методических комплексов дисциплин

- А22 Автоматизированное проектирование средств и систем управления [Электронный ресурс]: курс лекций / Е. Е. Носкова, Д. В. Капулин, Ю. В. Краснобаев, С. В. Ченцов. – Электрон. дан. (4 Мб). – Красноярск : ИПК СФУ, 2009. – (Автоматизированное проектирование средств и систем управления : УМКД № 1604–2008 / рук. творч. коллектива С. В. Ченцов). – 1 электрон. опт. диск (DVD). – Систем. требования : *Intel Pentium* (или аналогичный процессор других производителей) 1 ГГц ; 512 Мб оперативной памяти ; 50 Мб свободного дискового пространства ; привод *DVD* ; операционная система *Microsoft Windows XP SP 2 / Vista* (32 бит) ; *Adobe Reader 7.0* (или аналогичный продукт для чтения файлов формата *pdf*).
ISBN 978-5-7638-1704-1 (комплекса)
ISBN 978-5-7638-1766-9 (курса лекций)
Номер гос. регистрации в ФГУП НТЦ «Информрегистр» 0320902523 (комплекса)

Настоящее издание является частью электронного учебно-методического комплекса по дисциплине «Автоматизированное проектирование средств и систем управления», включающего учебную программу дисциплины, наглядное пособие «Автоматизированное проектирование средств и систем управления. Презентационные материалы», лабораторный практикум, методические указания по самостоятельной работе, контрольно-измерительные материалы «Автоматизированное проектирование средств и систем управления. Банк тестовых заданий».

В курсе лекций изложены основы разработки и применения математического описания, методов функционального анализа и синтеза систем управления, задачи и методы конструкторского проектирования ССУ как основы математического обеспечения современных систем автоматизированного проектирования (САПР) ССУ. Рассмотрены аспекты применения информационных систем в процессе проектирования ССУ, включая этап испытаний.

Предназначен для студентов направления подготовки магистров 220200.68 «Управление в технических системах» укрупненной группы 220000 «Автоматизация и управление».

© Сибирский федеральный университет, 2009

Рекомендовано к изданию Инновационно-методическим управлением СФУ

Редактор Т. М. Пыжик

Разработка и оформление электронного образовательного ресурса: Центр технологий электронного обучения Информационно-телекоммуникационного комплекса СФУ; лаборатория по разработке мультимедийных электронных образовательных ресурсов при КрЦНИТ

Содержимое ресурса охраняется законом об авторском праве. Несанкционированное копирование и использование данного продукта запрещается. Встречающиеся названия программного обеспечения, изделий, устройств или систем могут являться зарегистрированными товарными знаками тех или иных фирм.

Подп. к использованию 30.11.2009

Объем 4 Мб

Красноярск: СФУ, 660041, Красноярск, пр. Свободный, 79

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
МОДУЛЬ 1. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА И ТЕХНОЛОГИИ КОМПЛЕКСНОЙ АВТОМАТИЗАЦИИ ЭТАПА ПРОЕКТИРОВАНИЯ ССУ	8
ТЕМА 1. ПРОБЛЕМАТИКА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ СРЕДСТВ ССУ	8
Лекция 1. Задачи и средства автоматизированного проектирования ССУ ...	8
Предмет и задачи дисциплины	8
Постановка задачи автоматизации проектирования ССУ	9
Системный подход к проектированию ССУ	13
Структуризация процесса проектирования ССУ	15
Классификация САПР	21
ТЕМА 2. ФУНКЦИИ CAE/CAD/CAM-СИСТЕМ В РАМКАХ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ ПРОИЗВОДСТВА ССУ. СОСТАВ ИНТЕГРИРОВАННЫХ САПР	24
Лекция 2. Функциональное назначение интегрированных CAE/CAD/CAM-систем при проектировании ССУ	24
Функции CAD-систем	24
Функции CAM-систем	25
Функции CAE-систем	26
CALS-технологии	27
Функции АСУП (ERP-систем)	28
Функции SCADA-систем	29
Функции систем управления документами и документооборотом	30
Лекция 3. Функциональный и структурный состав интегрированных САПР	33
Функциональный состав интегрированных САПР	33
Интерфейсы, языки, форматы межпрограммных обменов в САПР	34
Структурный состав интегрированных САПР	39
МОДУЛЬ 2. МОДЕЛИ И МЕТОДЫ АНАЛИЗА ССУ ПРИ АВТОМАТИЗАЦИИ ЭТАПА ПРОЕКТИРОВАНИЯ	41



ТЕМА 3. МОДЕЛЬНОЕ ПРЕДСТАВЛЕНИЕ СРЕДСТВ И СИСТЕМ УПРАВЛЕНИЯ (ССУ) 41

Лекция 4. Модельное представление систем управления и элементов ССУ как объектов проектирования	41
Модельное представление СУ и элементов ССУ как объектов проектирования	41
Классификация моделей СУ как объектов проектирования	43
Этапы математического моделирования СУ	45
Математические модели систем управления	49
Математические модели устройств СУ	56
Математические модели элементов устройств СУ	63
Лекция 5. Методы формирования моделей ССУ.	65
Формирование математических моделей элементов устройств СУ	65
Формирование математических моделей устройств СУ	75
Формирование математических моделей систем управления	89

ТЕМА 4. МЕТОДЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ: МЕТОДЫ АНАЛИЗА ССУ . . 91

Лекция 6. Методы анализа ССУ в САПР и требования к ним	91
Методы анализа систем управления в САПР.	91
Виды анализа как проектной процедуры при автоматизированном проектировании ССУ.	92
Требования к методам анализа в САПР.	94
Лекция 7. Алгоритмы и методы анализа статических режимов ССУ в интегрированных САПР	96
Методы анализа в частотной области, их основные характеристики	96
Методы анализа статических характеристик	104
Оценка эффективности методов анализа СУ при их использовании в САПР	109
Лекция 8. Алгоритмы и методы анализа ССУ во временной области в интегрированных САПР	112
Методы анализа СУ во временной области	112
Классификация основных численных методов решения систем ОДУ	113
Методы анализа СУ с «жесткими» системами уравнений	116
Лекция 9. Параметры оценки эффективности методов анализа во временной области	125
Контроль точности в алгоритмах анализа СУ	125
Оценка устойчивости методов интегрирования	127
Алгоритмы автоматического выбора шага.	132
Выбор эффективных методов анализа переходных процессов СУ	141

Лекция 10. Алгоритмы и методы анализа чувствительности и статистических испытаний ССУ в САПР	143
Анализ чувствительности ССУ	143
Методы анализа чувствительности СУ при их использовании в САПР	146
Статистический анализ СУ в САПР	147

МОДУЛЬ 3. МОДЕЛИ СИНТЕЗА ССУ И ВЕРИФИКАЦИИ ПРОЕКТНЫХ РЕШЕНИЙ ПРИ АВТОМАТИЗАЦИИ ЭТАПА ПРОЕКТИРОВАНИЯ .153

ТЕМА 5. МЕТОДЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ: МЕТОДЫ СИНТЕЗА ССУ . 153

Лекция 11. Методы и алгоритмы технической оптимизации ССУ в САПР .	153
Параметрическая оптимизация средств и систем управления	153
Постановка задач параметрической оптимизации ССУ	157
Методы и алгоритмы оптимизации ССУ в интегрированных САПР	160
Лекция 12. Формализация сведений о ССУ как объектах структурного синтеза	164
Структурный синтез технических систем в САПР	164
Классификация процедур структурного синтеза СУ	165
Подходы к алгоритмизации задач структурного синтеза ССУ	169
Лекция 13. Алгоритмы и методы структурного синтеза ССУ в САПР	173
Методы математического программирования	175
Методы решения изобретательских задач	176
Методы с неопределенными исходными данными	179
Методы распространения ограничений	180
Переборные методы	181
Генетические алгоритмы	183

ТЕМА 6. АВТОМАТИЗАЦИЯ КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ ССУ

Лекция 14. Математическое моделирование ССУ при конструировании ...	189
Автоматизация конструкторского проектирования в рамках комплексной автоматизации этапа проектирования ССУ: основные понятия	189
Модели коммутационных схем	192
Математические модели для решения задач конструкторского проектирования	201
Модель квадратичного назначения	206
Метод ветвей и границ при конструировании	208
Непрерывные модели конструкций	211
Лекция 15. Алгоритмы автоматизации конструкторского проектирования ССУ	215

Алгоритмы конструкторского проектирования систем управления: конструктивные и итерационные	215
Алгоритмы компоновки	217
Последовательный алгоритм компоновки по связности	218
Итерационные алгоритмы компоновки	221
Алгоритмы размещения	231
Алгоритмы последовательного размещения по связности	233
Метод обратного размещения	235
Структура итерационных алгоритмов	236
Алгоритмы парных перестановок	237
Алгоритмы трассировки соединений	240
Лекция 16. Контроль полученных конструктивных решений	246
Контроль соответствия конструкции исходной принципиальной электрической схеме	247
Анализ помехоустойчивости	247
Оценка тепловых режимов конструкции	254
 ТЕМА 7. АВТОМАТИЗАЦИЯ ИСПЫТАНИЙ ССУ 255	
Лекция 17. Методы и алгоритмы испытаний ССУ	255
Испытания как часть процесса проектирования СУ	255
Автоматизация испытаний	257
Обзор современных автоматизированных систем управления испытаниями	261
 ЗАКЛЮЧЕНИЕ	264
 СПИСОК ЛИТЕРАТУРЫ	265

ВВЕДЕНИЕ

Технология информационной поддержки процесса проектирования является неотъемлемой частью современного производства. Именно успехи автоматизации проектирования в областях моделирования и анализа, параметрического и структурного синтеза служат базой дальнейшего развития информационных систем и технологий, используемых в производственных процессах.

Основными информационными системами поддержки процесса проектирования технических систем являются интегрированные современные системы автоматизированного проектирования (САПР). Назначение интегрированных САПР при автоматизации задач проектирования средств и систем управления будет зависеть от состава системы, видов анализа (устойчивость, качество и др.) и синтеза, этапов и уровней проектирования.

Конспект лекций по дисциплине «Автоматизированное проектирование средств и систем управления» для подготовки магистров по направлению 220200.68 «Управление в технических системах» служит основой при изучении принципов разработки и функционирования информационных систем поддержки проектирования средств и систем управления.

В лекционном материале изложены основы разработки и применения математического описания средств и систем управления (ССУ), современных методов их анализа и синтеза, используемых на этапах функционального и конструкторского проектирования. Показаны особенности математических моделей и методов их формирования при рассмотрении систем управления (СУ) на разных уровнях описания. Предложена классификация математических моделей по степени детализации СУ как сложной системы в соответствии с трехуровневым представлением: математические модели всей СУ как сложной системы; математические модели устройств СУ – средств управления – как отдельных систем; математические модели элементов устройств СУ как подсистем.

Указана специфика задач технической оптимизации применительно к разработке сложных систем управления. Отмечены основные тенденции при решении задач структурного синтеза на основе математического аппарата нечеткой логики, теории принятия решений, теории решения изобретательских задач и методов искусственного интеллекта. Также рассмотрены некоторые вопросы автоматизации этапа испытаний СУ.

В конце конспекта лекций представлен список литературы по дисциплине.



МОДУЛЬ 1. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА И ТЕХНОЛОГИИ КОМПЛЕКСНОЙ АВТОМАТИЗАЦИИ ЭТАПА ПРОЕКТИРОВАНИЯ ССУ

ТЕМА 1. ПРОБЛЕМАТИКА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ СРЕДСТВ ССУ

Лекция 1. Задачи и средства автоматизированного проектирования ССУ

Предмет и задачи дисциплины. Постановка задачи автоматизации проектирования ССУ. Системный подход к проектированию ССУ. Структуризация процесса проектирования ССУ. Типизация и унификация проектных решений и средств проектирования ССУ. Классификация САПР.

Предмет и задачи дисциплины

Изучение поведения и конструирование систем управления, обладающих требуемыми в приложениях свойствами, является ключевой задачей теории управления. При этом на первый план выдвигаются такие свойства систем, как устойчивость, оптимальность, поведение в присутствии неопределенных помех и т. д. Понятие абстрактной системы сформировалось за последние 30 лет. Оно обладает большой общностью и его строгое определение достаточно сложно. На описательном уровне под системой можно понимать техническую, физическую, биологическую, экологическую и любую другую систему. Математически указанные системы могут описываться различными способами, в зависимости от класса системы [5]. Приведем наиболее часто встречающиеся классы систем и способы их описания:

- непрерывные системы, описываемые обыкновенными дифференциальными уравнениями;
- дискретные системы, описываемые конечно-разностными уравнениями;
- системы с распределенными параметрами, описываемые дифференциальными уравнениями в частных производных;
- системы с последействием, описываемые функционально-дифференциальными уравнениями. Такие системы возникают тогда, когда протекание процесса определяется не только состоянием системы в данный момент, но также и предысторией процесса;
- стохастические системы, описываемые вероятностными понятиями и методами.



Для описания одной и той же реальной системы может быть использован разный математический аппарат, который зависит от целей исследования и требований точности и адекватности. Именно математическое описание будет определять выбор инструментальных средств и технологий проектирования систем.

При этом на современном этапе особое внимание уделяется проектированию технических систем, характерной особенностью которых является резкое повышение их логической сложности, ужесточения требований качества проектирования и снижения времени и стоимости разработки. Решить указанные противоречия, возникающие при проектировании технических систем, особенно средств и систем управления, на современном этапе возможно только через разработку и внедрение систем автоматизированного проектирования.

Умение применять такое инструментальное средство проектирования ССУ, как САПР, основывается на понимании того, каким образом строится САПР, ее структура и отдельные подсистемы, на знании математических моделей, методов их формирования, методов и алгоритмов анализа и синтеза, которые положены в основу этих подсистем.

Задача изучения дисциплины – дать будущему специалисту основополагающие сведения по решению научно-практических задач при создании, модернизации и эксплуатации систем автоматизированного проектирования технических средств и систем управления.

Основными задачами дисциплины являются: изучение основ методов формирования математических моделей объектов автоматизации и управления, методов автоматизированного проектирования автоматических и автоматизированных средств и систем управления объектами различной природы с применением современных компьютерных технологий; использование средств автоматизации проектирования последних поколений в целях разработки технических средств и систем управления на современной элементной базе.

Постановка задачи автоматизации проектирования ССУ

При создании современных технических систем определяющими являются системы автоматического управления и регулирования, которые за долгий путь своего развития – от центробежных регуляторов до современных комплексных СУ подвижными объектами – проникли во все области техники [4].

Назовем основные устройства, из которых состоит СУ.

Объект управления – аппарат, установка, требуемый режим работы которых должен поддерживаться.

Регулятор, или управляющая часть СУ, поддерживает требуемый режим работы объекта управления либо заменяет режим в соответствии с заданным законом или программой управления.

Регулятор включает в себя следующие устройства:

измерительные и чувствительные устройства, предназначенные для измерения и преобразования регулируемых величин или возмущающих воздействий разнообразного физического происхождения в сигналы, передаваемые и используемые в СУ;

программные, или задающие устройства, предназначенные для выработки сигналов, меняющихся по заранее определенному закону;

усилительно-преобразующие устройства, предназначенные для усиления сигналов, проходящих в регуляторах. Эти устройства преобразуют сигналы и управляют энергией, поступающей от внешнего источника в управляющие воздействия различного физического происхождения (электрические, гидравлические, пневматические и др.);

корректирующие устройства служат для обеспечения требуемых характеристик по устойчивости, качеству и точности управления. Эти устройства могут иметь различное физическое происхождение (микропроцессоры, электрические цепи, электромеханические преобразователи, гидравлические и пневматические контуры);

исполнительные устройства осуществляют непосредственное воздействие на объект управления (механическое, электрическое, гидравлическое, пневматическое и т. п.).

На [рис.1.1](#) приведена структура СУ летательного аппарата (ЛА) по курсу, включающая перечисленные устройства.

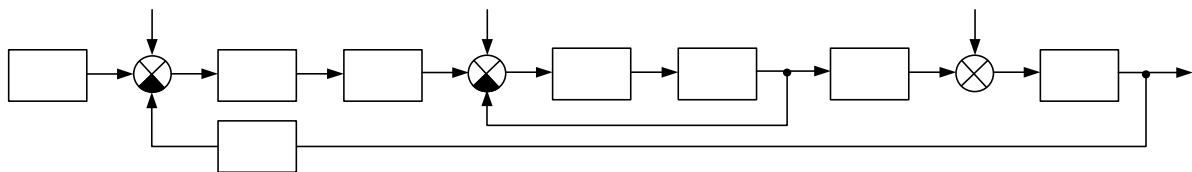


Рис. 1.1

Управляемый объект 1 (корпус ЛА) под влиянием исполнительного устройства 7 (руля курса) поворачивается на угол курса Ψ ; измерительное устройство 2 (навигационный прибор) измеряет угол курса Ψ и выдает пропорциональный ему электрический сигнал U на сравнивающее устройство, вырабатывающее сигнал рассогласования ε между задаваемым от программного устройства 8 (бортовая цифровая вычислительная машина (БЦВМ) или аналоговый задатчик курса) курсом и измеренным углом Ψ ; сигнал ε поступает на усилительно-преобразующее устройство 3 и далее на корректирующее устройство 4 (микропроцессор, электрические цепи на операционных усилителях). Скорректированный сигнал через усилительно-преобразующее устройство 5 поступает на привод исполнительного устройства 6, которое перемещает руль 7 на угол δ . Как следует из структуры этой СУ ([рис. 1.1](#)), в ней имеется (помимо основной) местная обратная связь. Она содержит физически разнородные элементы (аэродинамический объект управления, элек-

трехмеханические навигационные приборы, гидравлические приводы рулей, электронную БЦВМ и т. д.), в ней непрерывно меняется состояние отдельных устройств и системы в целом. Для выполнения основной цели – удерживать ЛА на заданном курсе – эта СУ должна удовлетворять многим критериям – быть высоконадежной, устойчивой к помехам и возмущениям $F_{1,2,3}$, обладать малой энергоемкостью и т. д. Ряд устройств, входящих в такую сложнейшую систему, также являются СУ. Например, в навигационных приборах широко применяются гиростабилизаторы, которые служат для измерения параметров углового движения ЛА, в данном случае курса, и стабилизации положения некоторой площадки (платформы) с установленными на ней приборами в неизменном положении относительно оси X .

На [рис. 1.2](#) представлена СУ турбогенераторной установки по частоте (скорости) вращения и давлению пара турбин. Объект регулирования – турбогенератор (ТГ) – включает в себя турбину и генератор 1. Регулирующий орган 6 изменяет скорость вращения вала турбины ω и давления пара p , подаваемого на лопасти турбины; чувствительные элементы 2 (индукционный датчик и датчик давления пара) посылают на сравнивающие устройства 3(3')

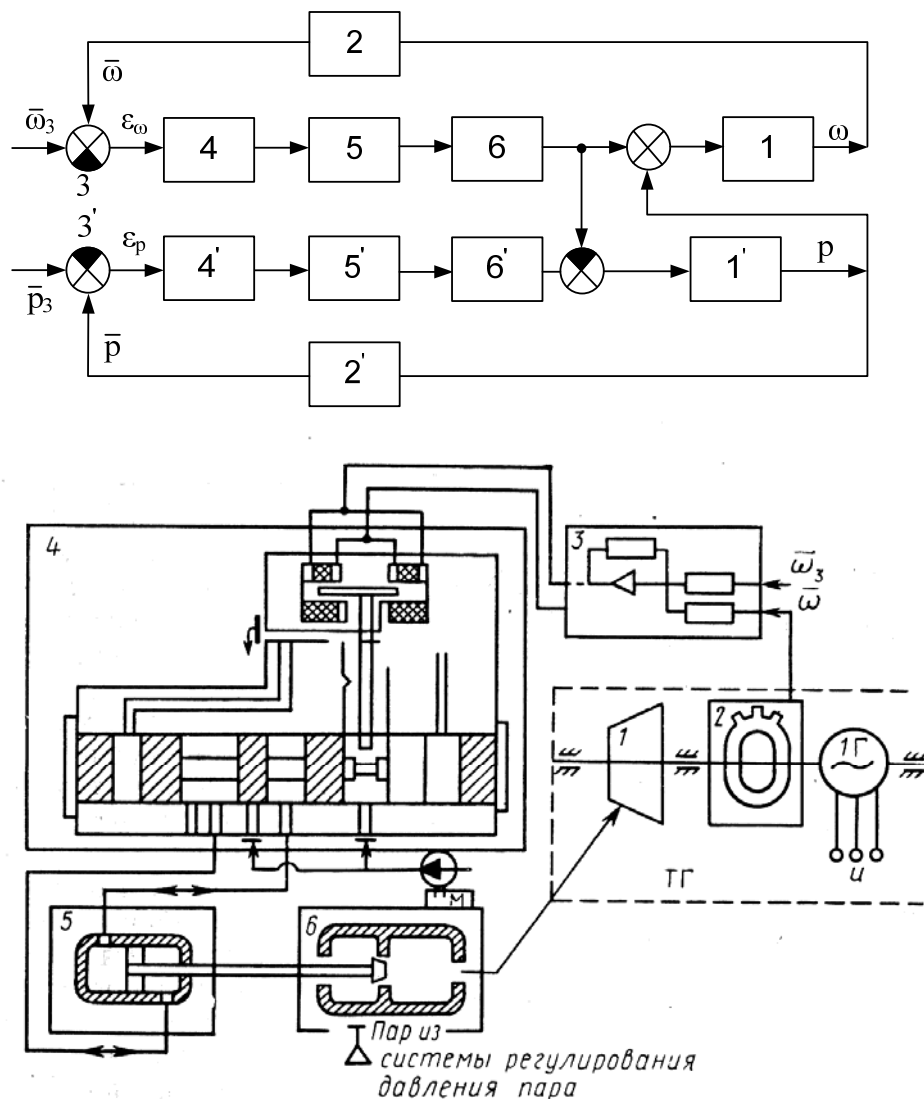


Рис. 1.2

измеренные и преобразованные скорость вращения турбины ω и давления пара p , откуда сигналы рассогласования $\varepsilon\omega$, εp попадают на усилительно-преобразующие устройства 4(4') и далее на исполнительные элементы 5(5'), перемещающие парораспределительные клапаны.

Этой СУ также присуща физическая неоднородность объекта управления, гидроприводов, электромеханических преобразователей энергии, многочисленные местные обратные связи, сложная динамика процессов управления.

Обобщая характерные свойства СУ широкого класса, можно выделить следующие особенности систем управления как объектов проектирования:

- Гетерогенность – физическая разнородность устройств и элементов, входящих в СУ;
- Многокритериальность условий их применения и работоспособности, при этом большинство критериев противоречивы (точность и устойчивость, надежность и массогабаритные характеристики и т. д.);
- Неопределенность задаваемых параметров и возмущающих воздействий, определяемая наличием не только внешних, но и внутренних «шумов», нестационарностью во времени параметров устройств СУ;
- Внедрение микропроцессоров и управляющих ЭВМ в СУ;
- Многомерность СУ, наличие нескольких контуров управления.

Эти основные свойства диктуют требования к проектированию и производству СУ как специфического класса технических систем.

Рассмотрим особенности СУ как объектов проектирования.

В отличие от других объектов машиностроения и приборостроения, являющихся обычно отдельными устройствами, СУ представляет собой систему из устройств, работающих в режиме управления заданным объектом. При этом большой вес приобретают такие проектные процедуры, как анализ устойчивости, качества и точности СУ, синтез регуляторов (в том числе на микропроцессорах), построение математических моделей объектов регулирования и отдельных устройств регуляторов, сборка и стыковка отдельных конструктивных узлов. При проектировании СУ существенное значение приобретает физическая разнородность, гетерогенность устройств, входящих в систему, и возмущающих воздействий, которые действуют на нее.

Анализ СУ показывает, что в средствах и устройствах управления используются практически все известные и пригодные для технического применения физические явления. Это придает особенности процессу проектирования, выражающиеся как в используемом инструменте проектирования – САПР, так и в организации процесса проектирования на предприятиях по специализации.

При разработке принципиальной схемы СУ приходится учитывать реальные (не идеализированные) характеристики и физическую природу ее элементов, источники их энергии, способы преобразования форм энергии и сигналов, место включения управляющих воздействий; при разработке конструкции и технологии изготовления существенное значение приобретают операции сборки и согласования разнообразных конструкторских и технологических форм отдельных устройств; важное значение приобретает проектирование в составе СУ управляющих ЭВМ и микропроцессоров. Проектиро-

вание таких устройств связано с разработкой как аппаратуры, так и алгоритмических и программных средств, что значительно расширяет круг вопросов, решаемых проектировщиками.

Испытания СУ становятся важнейшим этапом проектирования, так как при разработке принципиальных схем никогда не удастся учесть все специальные конструктивные, технологические и физические свойства отдельных устройств, входящих в СУ.

Указанные особенности средств и СУ делает их проектирование и производство дорогостоящим и весьма трудоемким процессом в рамках жизненного цикла, по сравнению с иными техническими объектами. Рассмотрим, что же такое процесс проектирования ССУ, полагая, что сама система управления является сложным техническим объектом.

Проектирование технического объекта – создание, преобразование и представление в принятой форме образа не существующего объекта. Образ объекта или его составных частей может создаваться в воображении человека в результате творческого процесса или генерироваться в соответствии с некоторыми алгоритмами в процессе взаимодействия человека и ЭВМ. В любом случае инженерное проектирование начинается при наличии выраженной потребности общества в некоторых технических объектах, которыми могут быть промышленные изделия или процессы, в частности, конкретная СУ.

Системный подход к проектированию ССУ

Проектирование сложных объектов, таких как СУ, основано на применении идей и принципов, изложенных в ряде теорий и подходов. Наиболее общим подходом является системный, его идеями пронизаны различные методики проектирования сложных систем.

Основные идеи и принципы проектирования сложных систем выражены в системном подходе. Для специалиста в области системотехники они являются очевидными и естественными, однако их соблюдение и реализация зачастую сопряжены с определенными трудностями, обусловливаемыми особенностями проектирования. Как и большинство взрослых образованных людей, правильно использующих родной язык без привлечения правил грамматики, инженеры используют системный подход без обращения к пособиям по системному анализу. Однако интуитивный подход без применения правил системного анализа может оказаться недостаточным для решения все более усложняющихся задач инженерной деятельности.

Основной общий принцип системного подхода заключается в рассмотрении частей явления или сложной системы с учетом их взаимодействия. Системный подход включает в себя выявление структуры системы, типизацию связей, выявление атрибутов, анализ влияния внешней среды.

Системный подход рассматривают как направление научного познания и социальной политики. Он является базой для обобщающей дисциплины «Системный анализ», другое используемое название – «Теория систем» – дисциплина, в которой конкретизируются положения системного подхода;

она посвящена исследованию и проектированию сложных экономических, социальных и технических систем, чаще всего слабоструктурированных. При проектировании систем цели достигаются в многошаговых процессах принятия решений. Методы принятия решений часто выделяют в самостоятельную дисциплину, которую называют «Теория принятия решений».

В технике дисциплину, в которой исследуются сложные технические системы, их проектирование, и аналогичные теории систем чаще всего называют системотехникой. Предмет системотехники – организация процесса создания, использования и развития технических систем. В системотехнике важно уметь сформулировать цели системы и организовать ее рассмотрение с позиций поставленных целей. Тогда можно отбросить лишние и малозначимые части при проектировании и моделировании, перейти к постановке оптимизационных задач.

Системы автоматизированного проектирования и управления относятся к числу наиболее сложных современных искусственных систем. Их проектирование и сопровождение невозможны без системного подхода. Поэтому идеи и положения системотехники входят составной частью в дисциплины, посвященные изучению современных автоматизированных систем и технологий их применения. Интерпретация и конкретизация системного подхода имеют место в ряде известных подходов с другими названиями, которые также можно рассматривать как компоненты системотехники. Таковы структурный, блочно-иерархический, объектно-ориентированный подходы.

При структурном подходе (как разновидности системного) требуется синтезировать варианты системы из компонентов (блоков) и оценивать варианты при их частичном переборе с предварительным прогнозированием характеристик компонентов.

Блочно-иерархический подход к проектированию использует идеи декомпозиции сложных описаний объектов и соответственно средств их создания на иерархические уровни и аспекты, вводит понятие стиля проектирования (восходящее и нисходящее), устанавливает связь между параметрами соседних иерархических уровней.

Ряд важных структурных принципов, используемых при разработке информационных систем и прежде всего их программного обеспечения (ПО), выражен в объектно-ориентированном подходе к проектированию (ООП). Такой подход имеет следующие преимущества в решении проблем управления сложностью и интеграции ПО: 1) вносит в модели приложений большую структурную определенность, распределяя представленные в приложении данные и процедуры между классами объектов; 2) сокращает объем спецификаций, благодаря введению в описания иерархии объектов и отношений наследования между свойствами объектов разных уровней иерархии; 3) уменьшает вероятность искажения данных вследствие ошибочных действий за счет ограничения доступа к определенным категориям данных в объектах. Описание в каждом классе объектов допустимых обращений к ним и принятых форматов сообщений облегчает согласование и интеграцию ПО.

Для всех подходов к проектированию сложных систем характерны следующие особенности [2]:

- структуризация процесса проектирования, выражаемая декомпозицией проектных задач и документации, выделением стадий, этапов, проектных процедур. Эта структуризация является сущностью *блочной-иерархического подхода* к проектированию;
- итерационный характер проектирования;
- типизация и унификация проектных решений и средств проектирования.

Структуризация процесса проектирования ССУ

При проектировании сложных объектов различают стадии, этапы, процедуры проектирования.

Стадии проектирования – наиболее крупные части проектирования как процесса, развивающегося во времени. В общем случае выделяют стадии научно-исследовательских работ (НИР), эскизного проекта или опытно-конструкторских работ (ОКР), технического, рабочего проектов, испытаний опытных образцов или опытных партий. Стадию НИР иногда называют предпроектными исследованиями или стадией технического предложения. Очевидно, что по мере перехода от стадии к стадии степень подробности и тщательность проработки проекта возрастают, и рабочий проект уже должен быть вполне достаточным для изготовления опытных или серийных образцов. Близким к определению стадии, но менее четко оговоренным понятием, является понятие *этапа проектирования*.

Стадии (этапы) проектирования подразделяют на составные части, называемые *проектными процедурами*. Выполнение проектной процедуры заканчивается принятием проектного решения. Примерами проектных процедур могут служить подготовка детализованных чертежей, анализ кинематики, моделирование переходного процесса, оптимизация параметров и другие проектные задачи.

Проектные процедуры делятся на процедуры анализа и синтеза, которые, в свою очередь, можно расчленивать на более мелкие компоненты, называемые проектными операциями. Например, при анализе прочности детали сеточными методами, операциями могут быть построение сетки, выбор или расчет внешних воздействий, собственно моделирование полей напряжений и деформаций, представление результатов моделирования в графической и текстовой формах. Процедуры анализа делятся на процедуры одновариантного и многовариантного анализа. Процедуры синтеза делятся на процедуры параметрического синтеза (параметрической оптимизации) и структурного синтеза.

Обычно этапы проектирования СУ задаются ГОСТом или ОСТом данной отрасли промышленности, а уточняются на конкретном предприятии и делятся на следующие: согласование ТЗ, техническое предложение, эскизный проект, рабочее проектирование, изготовление опытных образцов, испытания. На каждом этапе проектирования совершается очередной «виток» прохождения будущего проекта через все критерии и ограничения, воздвигаемые

целями проектирования и возможностями проектного предприятия. Процесс проектирования можно представить сходящейся спиралью (рис. 1.3) [4].

Каждый виток такой спирали обязательно проходит через все критерии и ограничения, сформулированные в ТЗ и уточненные в дальнейшем проектировании. По мере продвижения к окончательному проекту спираль скручивается, отбрасывая все негодные, побочные и малозначащие проектные решения и другие ненужные продукты процесса проектирования и оставляя только те проектные решения, которые, будучи воплощены в техническую документацию, и составят окончательный проект. Диаметр каждого витка этой условной спирали сужается, отражая характерную черту процесса проектирования – продвижение от расплывчатой, широкой, неопределенной во многих деталях начальной итерации проекта к конкретному завершеному в технической документации и опытном образце окончательному проекту. Чтобы перейти от одного витка к другому, более высокому, требуется затратить много труда и энергии большого коллектива проектировщиков, что во времени растягивается на долгие месяцы и даже годы, так как ограниченность проектного предприятия в численности квалифицированных проектировщиков и других ресурсах не дает возможности сразу же получать проектные решения по всем направлениям

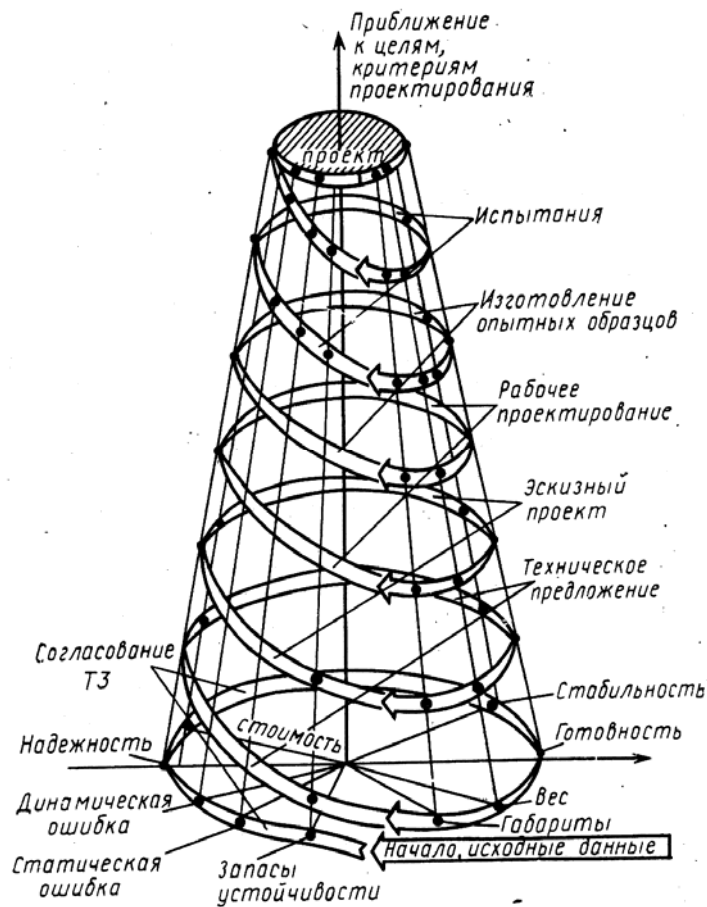


Рис.1.3

Обычно ТЗ представляют в виде некоторых документов, и оно является исходным описанием объекта. Результатом проектирования, как правило, служит полный комплект документации, содержащий достаточные сведения для изготовления объекта в заданных условиях. Эта документация и есть проект, точнее конечное описание объекта. Более коротко, *проектирование – процесс, заключающийся в получении и преобразовании исходного описания объекта в окончательное описание на основе выполнения комплекса работ исследовательского, расчетного и конструкторского характера.*

ТЗ на проектирование САУ и ее устройств в соответствии с ГОСТ 15100–80 включает в себя следующие основные разделы:

1. Наименование, цель и область применения.
2. Основание для создания.
3. Требования к техническим характеристикам.
4. Экономические показатели – затраты на проектирование, источники финансирования.
5. Условия и серийность производства.
6. Порядок испытаний и ввода в действие.
7. Сроки проектирования.

Кроме этих основных требований ТЗ может включать многочисленные характеристик, условия эксплуатации и производства САУ и ее устройств. Это основные разделы ТЗ наполняются конкретными требованиями и условиями проектирования САУ, записанными на профессиональном техническом языке проектировщика.

Этап согласования технического задания обычно реализуется во взаимодействии «предприятия-заказчика» и «предприятия-проектировщика». Заказчик, естественно, стремится получить от проектировщика все, что он хотел бы воплотить в проекте. Проектировщик исходит из конкретных возможностей проектного предприятия и тех необходимых в проектом деле и сохраняемых на непредвиденные случаи запасов ресурсов, а также предварительных заделов по аналогичным разработкам, которыми он располагает. Техническое задание бывает двух типов: на проектирование СУ по известным прототипам с указанием существенных отличий и на проектирование новой СУ с детальным перечислением всех существенных свойств и характеристик. В любом случае идейная основа задания на проектирование исходит от заказчика, который в условиях возрастающей конкуренции должен внимательно следить за мировым уровнем СУ вообще и конкретным их использованием в той или иной предметной области. Заказчик и проектировщик при разработке ТЗ должны обязательно убедиться в существовании решения задач проектирования – сходимости спирали проектирования к проекту при заданных условиях и ограничениях.

Предварительный вариант ТЗ составляется проектировщиком и согласуется с заказчиком по каждому пункту. Часто по отдельным пунктам ТЗ возникают конфликтные ситуации, которые затягивают процесс согласования. В принципе требования и ограничения при проектировании противоречат друг другу, поэтому спроектировать «идеальную» СУ невозможно.

Наилучший реальный проект – тот, в котором наиболее эффективно найден компромисс.

В результате согласования появляется ТЗ как официальный документ, который содержит десятки листов текста, выполняется по ГОСТ 15101–80 и утверждается заказчиком и проектировщиком.

Техническое задание на СУ в целом порождает множество так называемых частных ТЗ на отдельные устройства САУ и их элементы. Эти ТЗ согласовываются между самими проектировщиками внутри проектного предприятия. Здесь также, разумеется, возникают конфликтные ситуации, и процесс согласования ТЗ повторяется, но на более низком уровне. Например, проектировщик СУ турбогенератора выдает ТЗ на чувствительные элементы, следящий привод и другие устройства подразделениям, проектирующим элементы СУ турбогенератора. Возникает дерево из ТЗ, аналогичное дереву целей. Первый ярус такого дерева – ТЗ на отдельные устройства, второй ярус получается из ТЗ на элементы этих устройств, далее идут ТЗ на конструкторскую и технологическую разработку, испытания.

Этап технического предложения предназначен для исследования и разработки принципов построения СУ, определения возможности выбора готовых или разработки новых ее устройств, определения путей и способов проектирования. На этом этапе формируется принципиальная схема СУ, осуществляются предварительные расчеты и оценки динамических, точностных и других характеристик СУ, т. е. проходят второй виток спирали (см. [рис. 1.3](#)) по всем критериям и ограничениям. На этом же этапе применяют анализ прототипов СУ, математическое моделирование и испытания прототипов, макетов САУ и ее устройств; осуществляют предварительные конструкторские и технологические проработки по стыковке устройств СУ и самим устройствам; подготавливают частные ТЗ на устройства и элементы СА, а также делают первые попытки решения конфликтных ситуаций процесса проектирования между требованиями ТЗ и ограничениями проектного предприятия. Этап завершается защитой предварительного проекта, который, развивая ТЗ, дает следующую итерацию проекта.

Предварительный проект обычно представляет собой достаточно объемный комплект документов, как правило, текстовых, выполненных по ГОСТ 2.118–73. Этот документ в отличие от ТЗ содержит не только задание, условия и ограничения, но и предварительные данные по будущей СУ – варианты структурных и функциональных схем всей системы и ее устройств, расчеты динамики и точности, массы и габаритов, предложения по конструкторским и технологическим решениям.

Этап эскизного проектирования начинается с результатов предыдущего этапа и заканчивается принципиальной схемой СУ и ее устройств. Именно принципиальные схемы, выбранные и детально обоснованные на этом этапе расчеты, моделирование, испытания на макетах, служат основой для разработки конструкторской и технологической документации по изготовлению СУ и ее устройств на заводе «в металле». Принципиальные схемы являются

неотъемлемой частью проекта, что качественно отличает этот этап от предыдущих.

На этапе эскизного проектирования осуществляется декомпозиция (разбиение) СУ на устройства, а устройств – на элементы. Ведется сложная, кропотливая разработка и взаимная увязка этих устройств и элементов. На данном (третьем) витке «большой спирали» (см. [рис. 1.3](#)) вновь проходят все критерии, условия и ограничения проектирования, но при этом активно начинают работать «малые спирали» – проектирование отдельных устройств СУ. Здесь все достаточно аналогично СУ в целом, но критерии, условия и ограничения носят более детальный и конкретный характер, сроки проектирования и стоимости меньше. Документация, выпускаемая на этом этапе (за исключением принципиальных схем) называется эскизной. Она носит временный характер и служит для изготовления отдельных экспериментальных макетов и образцов устройств СУ и контрольно-измерительной аппаратуры (КИА). Такая документация содержит эскизы технологических карт, монтажных схем, условий технической эксплуатации, предварительное программное обеспечение БЦВМ СУ ЛА и микропроцессоров в СУ ШР, ГСП, турбогенератора, программы проведения испытаний. По этой эскизной документации на опытном производстве изготавливают макеты устройств САУ и КИА. Макеты устройств СУ объединяются в комплексные стенды, на которых начинается реализация наиболее трудоемких и сложных проектных процедур СУ – стыковки отдельных ее устройств и элементов – методами стендовых испытаний и физического моделирования.

Параллельно проводят расчеты СУ. Проектировщики составляют математические модели объекта управления, устройств СУ, алгоритмы работы мини- и микропроцессоров; решают задачи анализа и синтеза регуляторов в СУ, оптимизации ее параметров, вырабатывают вычислительные алгоритмы для мини- и микропроцессоров СУ. В дальнейшем начинают проводить конструкторские расчеты устройств СУ на прочность, жесткость, определяют моменты инерции и центры тяжести, составляют программы для микропроцессоров СУ, осуществляют и рассчитывают надежность, стабильность, отлаживают программное обеспечение мини- и микропроцессоров СУ; далее описывают детальное технико-экономическое обоснование проекта. Эскизный проект выпускается в виде нескольких книг текстовой и эскизной конструкторской документации. На научно-техническом совете предприятия производится защита эскизного проекта. По результатам эскизного проекта вносят изменения и уточнения в ТЗ; заказчик проводит оценку правильности движения процесса проектирования к главной цели, соответствие этой цели принятых проектировщиком технических решений и принципов.

Этап рабочего проектирования предназначен для разработки полного комплекта технической документации, необходимой и достаточной для изготовления СУ и ее устройств на заводе. Начало этого «витка» процесса проектирования – принципиальные схемы и другие результаты эскизного проекта, его конец – электрические, кинематические, монтажные схемы; конструкторские чертежи общего вида и деталей, спецификации комплектующих изделий

и материалов; программная документация для мини- и микропроцессоров – инструкции программиста, описание программ, листинги; технологическая документация – технологические карты, маршруты.

В соответствии с требованиями ГОСТов, ЕСКД в проекте должны быть следующие обязательные документы:

1. Чертежи деталей.
2. Сборочный чертеж.
3. Теоретический чертеж.
4. Габаритный чертеж.
5. Монтажный чертеж.
6. Схема.
7. Спецификации.
8. Ведомость спецификаций.
9. Ведомость согласования документов.
10. Ведомость паспортов изделий.
11. Ведомость согласования применения покупных изделий.
12. Ведомость держателей подлинников.
13. Технические условия.
14. Программа и методика испытаний.
15. Таблицы результатов испытаний.
16. Конструкторские расчеты.
17. Документы на прочие расчеты.
18. Патентный формуляр.
19. Условия эксплуатации.
20. Документы на ремонтные работы.
21. Карта технического уровня и качества продукции.

По ЕСТД [4] технологическая документация содержит 14 документов, один из них графический (карта эскизов), остальные текстовые, в том числе: маршрутная карта; ведомость деталей сборочных единиц; типовой технологический процесс; карта технологического процесса; операционная карта.

В процессе рабочего проектирования по конструкторской и технологической документации изготавливают опытные образцы СУ и ее устройств. Завершается этап рабочего проектирования изготовлением опытных образцов и их автономными испытаниями.

Этап испытаний является неотъемлемой частью процесса проектирования. Испытания прототипов, макетов начинаются уже на первых витках проектирования, а испытаниями опытных образцов завершается процесс проектирования.

Преобразование исходного описания в окончательное порождает ряд промежуточных описаний, подводящих итоги решения некоторых задач и используемых для обсуждения и принятия проектных решений для окончания или продолжения проектирования.

Проектирование, при котором все проектные решения или их часть получают путем взаимодействия человека и ЭВМ, называют *автоматизированным*, в отличие от ручного (без использования ЭВМ) или автоматическим

(без участия человека на промежуточных этапах). Система, реализующая автоматизированное проектирование, представляет собой САПР (в англоязычном написании *CAD System – Computer Aided Design System*). Автоматическое проектирование возможно лишь в отдельных частных случаях для сравнительно легких объектов. Превалирующим в настоящее время является автоматизированное проектирование.

Проектирование сводится к выполнению некоторых последовательностей проектных процедур – маршрутов проектирования. Иногда разработку ТЗ на проектирование называют внешним проектированием, а реализацию ТЗ – внутренним.

САПР – организационно-техническая система, состоящая из комплекса средств автоматизации проектирования, взаимосвязанного с необходимыми подразделениями проектной организации или коллективом специалистов (пользователей) и выполняющая автоматизированное проектирование.

Создание и развитие САПР ведется очередями. Каждая из очередей САПР является расширением предыдущей, что достигается путем внедрения новых подсистем и компонентов САПР.

Дальнейшее развитие САПР осуществляется путем совершенствования имеющихся подсистем и компонентов САПР, их связей между собой и с подразделениями проектной организации.

Классификация САПР

Классификацию САПР осуществляют по ряду признаков, например, по приложению, целевому назначению, масштабам (комплексности решаемых задач), характеру базовой подсистемы – ядра САПР [7]. В общем смысле классификация – система соподчиненных понятий, часто представляемая в виде схем, таблиц и используемая как средство для установления связей между этими понятиями. Классификация фиксирует место объекта в системе, которая указывает на его свойства. Классификация создает условия для разработки технически обоснованных норм обеспечения процесса создания, функционирования и стандартизации в области САПР.

По приложениям наиболее представительными и широко используемыми являются следующие группы САПР.

1. САПР для применения в отраслях общего машиностроения. Их часто называют машиностроительными САПР или MCAD-системами (Mechanical CAD).

2. САПР для радиоэлектроники. Их названия – ECAD системы (Electronic CAD) или EDA (Electronic Design Automation).

3. САПР в области архитектуры и строительства.

Кроме того, известно большое число более специализированных САПР, или выделяемых в указанных группах, или представляющих самостоятельную ветвь в классификации. Примерами таких систем являются

САПР больших интегральных схем (БИС); САПР летательных аппаратов; САПР электрических машин и т. п.

По целевому назначению различают САПР или подсистемы САПР, которые обеспечивают разные аспекты (страты) проектирования:

1. САПР функционального проектирования, иначе САЕ (Computer Aided Engineering)-системы.

2. Конструкторские САПР, часто называемые САД-системами;

3. Технологические САПР, иначе называемые автоматизированными системами технологической подготовки производства АСТПП или САМ-системами (Computer Aided Manufacturing).

По масштабам различают отдельные программно-методические комплексы (ПМК) САПР, например, комплекс анализа прочности механических изделий в соответствии с методом конечных элементов (МКЭ) или комплекс анализа электронных схем; системы ПМК; системы с уникальными архитектурами не только программного (software), но и технического (hardware) обеспечения.

По характеру базовой подсистемы различают следующие разновидности САПР.

1. САПР на базе подсистем машинной графики и геометрического моделирования. Эти САПР ориентированы на приложения, где основной процедурой проектирования является конструирование, т. е. определение пространственных форм и взаимного расположения объектов. Поэтому к данной группе систем относится большинство графических ядер САПР в области машиностроения. В настоящее время появились унифицированные графические ядра, применяемые более чем в одной САПР, это ядра Parasolid фирмы EDS Unigraphics и ACIS фирмы Intergraph.

2. САПР на базе СУБД. Они ориентированы на приложения, в которых при сравнительно несложных математических расчетах перерабатывается большой объем данных. Такие САПР преимущественно встречаются в технико-экономических приложениях, например, при проектировании бизнес-планов, но имеют место также при проектировании объектов, подобных щитам управления в системах автоматики.

3. САПР на базе конкретного прикладного пакета. Фактически это автономно используемые программно-методические комплексы, например, имитационного моделирования производственных процессов, расчета прочности по методу конечных элементов, синтеза и анализа систем автоматического управления и т. п. Часто такие САПР относятся к системам САЕ. Примерами могут служить программы логического проектирования на базе языка VHDL, математические пакеты типа MathCAD.

4. Комплексные (интегрированные) САПР, состоящие из совокупности подсистем предыдущих видов. Характерными примерами комплексных САПР являются САЕ/CAD/CAM-системы в машиностроении или САПР БИС. Так, САПР БИС включает в себя СУБД и подсистемы проектирования компонентов, принципиальных, логических и функциональных схем, топологии кристаллов, тестов для проверки годности изделий. Для управления

столь сложными системами применяют специализированные системные среды.

По сложности объекта проектирования различают: САПР простых объектов, содержащих до 100 составных частей; объектов средней сложности, содержащих от 100 до 1000 составных частей; сложных объектов, содержащих от 10^3 до 10^4 составных частей; объектов высокой сложности – $>10^4$ составных частей.

По уровню автоматизации проектирования различают САПР низкоавтоматизированные – число автоматизированных проектных процедур составляет 25 % общего числа проектных процедур; среднеавтоматизированные – от 25 % до 50 % общего числа проектных процедур; высокоавтоматизированные – от 50 до 75 % общего числа проектных процедур. В этих системах применяют методы многовариантного оптимального проектирования. Чтобы получить эффективные результаты, в ГАП необходимо использовать САПР средне- и высокоавтоматизированного проектирования.

По комплексности САПР классифицируют на: одноэтапные, выполняющие один этап проектирования из всех установленных для объекта; многоэтапные, выполняющие несколько этапов проектирования из всех установленных для объекта; комплексные, выполняющие все этапы проектирования, установленные для объекта. По данному классификационному признаку современные САПР делят на «тяжелые» (многоэтапные и комплексные) и «легкие» (одноэтапные).

Одним из классификационных признаков САПР является количество выпускаемых проектных документов; САПР малой производительности, средней и высокой. В различных отраслях вводятся различные количественные характеристики информации, определяющие производительность САПР.

Указанная классификация – не догма. Теория автоматизации проектирования непрерывно развивается, появляются новые технические и программные средства и поэтому существующие схемы классификации САПР видоизменяются и совершенствуются.

Наиболее известны следующие САЕ/CAD/CAM-системы. «Тяжелые» системы (в скобках указана фирма, разработавшая или распространяющая продукт): Unigraphics (EDS Unigraphics); Solid Edge (Intergraph); Pro/Engineer (PTC – Parametric Technology Corp.), CATIA (Dassault Systemes), EUCLID (Matra Datavision), CADD5.5 (Computervision, ныне входит в PTC) и др. «Легкие» системы: AutoCAD (Autodesk); АДЕМ; bCAD (ПроПро Группа, Новосибирск); Caddy (Ziegler Informatics); Компас (Аскон, С.-Петербург); Спрут (Sprut Technology, Набережные Челны); Кредо (НИВЦ АСК, Москва). Системы, занимающие промежуточное положение (среднемасштабные): Cimatron, Microstation (Bentley), Euclid Prelude (Matra Datavision), T-FlexCAD (Топ Системы, Москва) и др. С ростом возможностей персональных ЭВМ грани между «тяжелыми» и «легкими» CAD/CAM-системами постепенно стираются.

ТЕМА 2. ФУНКЦИИ САЕ/CAD/CAM-СИСТЕМ В РАМКАХ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ ПРОИЗВОДСТВА ССУ. СОСТАВ ИНТЕГРИРОВАННЫХ САПР

Лекция 2. Функциональное назначение интегрированных САЕ/CAD/CAM-систем при проектировании ССУ

Процедуры анализа, моделирования, оптимизации проектных решений в САЕ-системах. Функции САD-систем. САLS-технологии. Функции АСУП (ERP-систем). Функции SCADA-систем. Функции систем управления документами и документооборотом.

Назначение интегрированных САПР при автоматизации задач проектирования будет зависеть от состава СУ, видов анализа (устойчивость, качество и др.) и синтеза, этапа и уровня проектирования. Проектировщик при разработке элементов и устройств СУ должен выбирать САПР в зависимости от целей проектирования и уровней описания СУ и функционального назначения САПР. Рассмотрим основное функциональное назначение автоматизированных систем информационной поддержки этапов жизненного цикла ССУ и, в частности, процесса проектирования.

Функции САD-систем

САD-системы в процессе разработки ССУ применяются для выполнения этапа конструкторского проектирования при выполнении процедур геометрического проектирования. Функции САD-систем при геометрическом проектировании подразделяют на функции двухмерного (2D) и трехмерного (3D) проектирования. К функциям 2D относятся черчение, оформление конструкторской документации; к функциям 3D – получение трехмерных моделей, метрические расчеты, реалистичная визуализация, взаимное преобразование 2D и 3D моделей.

Среди САD-систем различают «легкие» и «тяжелые» системы. Первые из них ориентированы преимущественно на 2D графику, сравнительно дешевы и менее требовательны в отношении вычислительных ресурсов. Вторые ориентированы на геометрическое моделирование (3D), они более универсальны и дороги, оформление чертежной документации в них обычно осуществляется с помощью предварительной разработки трехмерных геометрических моделей.

Функции САМ-систем

Основные функции САМ-систем: разработка технологических процессов, синтез управляющих программ для технологического оборудования с числовым программным управлением (ЧПУ), моделирование процессов обработки, в том числе построение траекторий относительного движения инструмента и заготовки в процессе обработки, генерация постпроцессоров для конкретных типов оборудования с ЧПУ (*NC* – Numerical Control), расчет норм времени обработки [2].

Характерным признаком современной техники является частая сменяемость изделий производства. Требования к повышению производительности удовлетворяются при создании гибких автоматизированных производств (ГАП). ГАП – это производственная единица, функционирующая на основе «безлюдной» технологии. В ГАП работа всех производственных компонентов (технологического оборудования, складских помещений, транспортных систем) координируется системой управления, обеспечивающей быструю перестройку. Комплекс задач технологической подготовки производства – разработка наиболее экономичного процесса изготовления изделия, полностью отвечающего техническим требованиям – в условиях ГАП решается путем создания автоматизированных систем технологической подготовки производства (АСТПП).

Основным направлением современного этапа автоматизации проектирования является создание комплексных (интегрированных) систем, включающих конструирование изделий, технологическое проектирование и изготовление (САПР/АСТПП/ГАП = CAD/CAM). Особенности ГАП являются невысокая серийность, меняющаяся номенклатура изделий, жесткие ограничения на сроки проектирования и производства. Интенсивный поток конструкторской и технологической информации требует сквозной автоматизации всех этапов разработки изделия – от согласования технического задания до получения полного комплекта конструкторско-технологической документации. При жизненном цикле промышленного изделия все виды систем автоматизации в той или иной мере взаимодействуют друг с другом, причем САПР непосредственно и в наибольшей степени должна взаимодействовать с автоматизированными системами научных исследований (АСНИ), АСТПП и автоматизированными системами управления производством (АСУП). АСТПП, составляющие основу системы САМ, выполняют синтез технологических процессов и программ для оборудования с числовым программным управлением (ЧПУ), выбор технологического оборудования и инструмента, оснастки и т. п. Модули системы САМ обычно входят в состав развитых САПР и поэтому интегрированные САПР часто называют системами САЕ/CAD/CAM.

Функции САЕ-систем

Функции САЕ-систем довольно разнообразны, так как связаны с проектными процедурами анализа, моделирования, оптимизации проектных решений. При функциональном проектировании ССУ тип используемой САЕ-системы будет определяться уровнем описания СУ: уровень системы в целом; уровень устройств СУ; уровень элементов устройств СУ.

В состав САЕ-систем объектов электрической природы, как правило, включают программы расчета:

- частотных характеристик;
- расчета установившихся процессов (анализ статики);
- расчет переходных процессов (анализ динамики);
- расчет шумов, спектров, вариации температуры;
- статистический анализ по методу Монте-Карло;
- расчет чувствительности;
- расчет наихудшего случая;
- учет задержек распространения сигналов в цифровых компонентах;
- параметрическую оптимизацию.

Большинство современных САЕ-систем для проектирования электронных устройств используют формат SPICE. Это формат входного языка проектирования, который поддерживается большинством универсальных САПР электронных устройств: Protel, OrCAD, MicroCAP, Proteus и др. Эти САПР можно использовать при проектировании ССУ на уровнях системы в целом и устройств СУ.

В состав САЕ-систем, традиционно называемых машиностроительными, прежде всего включают программы для моделирования полей физических величин, в том числе анализа прочности, который чаще всего выполняется в соответствии с МКЭ. Такие САПР применяются при проектировании элементов устройств СУ.

Примеры систем моделирования полей физических величин в соответствии с МКЭ: Nastran, Ansys, Cosmos, Nisa, Moldflow. Примеры систем моделирования динамических процессов на макроуровне: Adams и Dyna – в механических системах.

Для удобства адаптации САПР к нуждам конкретных приложений, для ее развития целесообразно иметь в составе САПР инструментальные средства адаптации и развития. Эти средства представлены той или иной CASE-технологией, включая языки расширения. В некоторых САПР применяют оригинальные инструментальные среды.

Примерами могут служить объектно-ориентированная интерактивная среда CAS.CADE в системе EUCLID, содержащая библиотеку компонентов, в САПР T-Flex CAD 3D предусмотрена разработка дополнений в средах Visual C++ и Visual Basic.

CALS-технологии

CALS-технология – это технология комплексной компьютеризации сфер промышленного производства, цель которой – унификация и стандартизация спецификаций промышленной продукции на всех этапах ее жизненного цикла. Основные спецификации представлены проектной, технологической, производственной, маркетинговой, эксплуатационной документацией. В CALS-системах предусмотрены хранение, обработка и передача информации в компьютерных средах, оперативный доступ к данным в нужное время и в нужном месте. Соответствующие системы автоматизации назвали автоматизированными логистическими системами или CALS (Computer Aided Logistic Systems). Поскольку под логистикой обычно понимают дисциплину, посвященную вопросам снабжения и управления запасами, а функции CALS намного шире и связаны со всеми этапами жизненного цикла промышленных изделий, то применяют и более соответствующую предмету расшифровку аббревиатуры CALS – Continuous Acquisition and LifeCycle Support [2].

Применение CALS позволяет существенно сократить объемы проектных работ, так как описания многих составных частей оборудования, машин и систем, проектировавшихся ранее, хранятся в базах данных сетевых серверов, доступных любому пользователю технологии CALS. Существенно облегчается решение проблем ремонтпригодности, интеграции продукции в различного рода системы и среды, адаптации к меняющимся условиям эксплуатации, специализации проектных организаций и т. п. Ожидается, что успех на рынке сложной технической продукции будет немислим вне технологии CALS.

Развитие CALS-технологии должно привести к появлению так называемых виртуальных производств, в которых процесс создания спецификаций с информацией для программно управляемого технологического оборудования, достаточной для изготовления изделия, может быть распределен во времени и пространстве между многими организационно автономными проектными студиями. Среди несомненных достижений CALS-технологии следует отметить легкость распространения передовых проектных решений, возможность многократного воспроизведения частей проекта в новых разработках и др.

Построение открытых распределенных автоматизированных систем для проектирования и управления в промышленности составляет основу современной CALS-технологии. Главная проблема их построения – обеспечение единообразного описания и интерпретации данных, независимо от места и времени их получения в общей системе, имеющей масштабы вплоть до глобальных. Структура проектной, технологической и эксплуатационной документации, языки ее представления должны быть стандартизованными. Тогда становится реальной успешная работа над общим проектом разных коллективов, разделенных во времени и пространстве и использующих разные CAE/CAD/CAM-системы.

Одна и та же конструкторская документация может быть использована многократно в разных проектах, а одна и та же технологическая документация адаптирована к разным производственным условиям, что позволяет существенно сократить и удешевить общий цикл проектирования и производства. Кроме того, упрощается эксплуатация систем. Следовательно, информационная интеграция является неотъемлемым свойством CALS-систем. Поэтому в основу CALS-технологии положен ряд стандартов, обеспечивающих такую интеграцию.

Важные проблемы, требующие решения при создании комплексных САПР, – управление сложностью проектов и интеграция ПО. Эти проблемы включают вопросы декомпозиции проектов, распараллеливания проектных работ, целостности данных, межпрограммных интерфейсов и др.

Для решения проблем совместного функционирования компонентов САПР различного назначения разрабатываются системы управления проектными данными – системы PDM (Product Data Management). Они либо входят в состав модулей конкретной САПР, либо имеют самостоятельное значение и могут работать совместно с разными САПР.

Услуги системы SCM – системы управления цепочками поставок – требуются уже на этапе производства

Управление данными в едином информационном пространстве на протяжении всех этапов жизненного цикла изделий возлагается на систему управления жизненным циклом продукции PLM-систему (Product Lifecycle Management). Характерная особенность PLM – обеспечение взаимодействия различных автоматизированных систем многих предприятий.

Функции АСУП (ERP-систем)

Функции автоматизированных систем управления производством (АСУП) выполняются системами на нескольких иерархических уровнях. Автоматизацию управления на верхних уровнях от корпорации до цеха осуществляют АСУП, классифицируемые как системы ERP или MRP-2. Наиболее развитые системы ERP выполняют различные бизнес-функции, связанные с планированием производства, закупками, сбытом продукции, анализом перспектив маркетинга, управлением финансами, персоналом, складским хозяйством, учетом основных фондов и т. п. Системы MRP-2 ориентированы главным образом на бизнес-функции, непосредственно связанные с производством [2].

На этапе реализации продукции выполняются функции управления отношениями с заказчиками и покупателями, проводится анализ рыночной ситуации, определяются перспективы спроса на планируемые к выпуску изделия. Эти задачи решаются с помощью системы CRM. Маркетинговые функции иногда возлагаются на систему S&SM, которая, кроме того, помогает решать проблемы обслуживания. Функции некоторых систем часто перекрываются.

Мировыми лидерами среди систем программного обеспечения АСУП являются системы R3 (фирма SAP) и Baan IV (Baan), широко известны также MANMAN/X (Computer Associates CIS), Elite Series (Tecsys Inc.), Marix (IBM) и др. Примерами комплексных систем управления предприятием, созданных в России, служат системы АККОРД фирмы «Атлант Информ», а также системы фирм «Галактика» и «Парус». Корпоративные информационные системы разрабатывают такие российские фирмы, как АйТи, R-Style и др.

Характерные особенности современных АСУП.

1. Открытость по отношению к ведущим платформам (UNIX, Windows, OS/2) и различным СУБД и прежде всего мощным СУБД типа Oracle, Ingres, Informix, Sybase; поддержка технологий типа ODBC (Open Data Base Connection), OLE (Object Linking and Embedding), DDE (Dynamic Data Exchange); поддержка архитектур клиент/сервер. Важная характеристика – возможность работы в среде распределенных вычислений.

2. Возможность сквозного выполнения всех допустимых бизнес-функций или их части, что обеспечивается модульным построением (количество функций может превышать 100).

3. Адаптируемость к конкретным заказчикам и условиям рынка.

4. Наличие инструментальных средств, в том числе языка расширения или 4GL (языка четвертого поколения). Так, в R3 используется язык АВАР/L, в Elite Series – язык Informix-4GL.

5. Техническое обеспечение АСУП – компьютерная сеть, узлы которой расположены как в административных отделах предприятия, так и в цехах.

Очевидно, что для создания и развития виртуальных предприятий необходимы распространение CALS-технологии не только на САПР, но и на АСУП, их интеграция в комплексные системы информационной поддержки всех этапов жизненного цикла промышленной продукции.

Функции SCADA-систем

Функциями АСУТП на уровнях цеха и участка являются сбор и обработка данных о состоянии оборудования и протекании производственных процессов для принятия решений по загрузке станков, по выполнению технологических маршрутов. Программное обеспечение АСУТП на этих уровнях представлено системой диспетчерского управления и сбора данных SCADA (Supervisory Control and Data Acquisition), а техническое обеспечение – персональными ЭВМ и микрокомпьютерами, связанными локальной вычислительной сетью. Кроме диспетчерских функций, SCADA выполняет роль инструментальной системы разработки программного обеспечения для промышленных систем компьютерной автоматизации, т. е. роль специфической CASE-системы. Для систем АСУТП характерно использование программируемых контроллеров (ПЛК или PLC – Programmed Logic Controller), – компьютеров, встроенных в технологическое оборудование.

Функции SCADA:



- сбор первичной информации от датчиков;
- хранение, обработка и визуализация данных;
- управление и регистрация аварийных сигналов;
- связь с корпоративной информационной сетью;
- автоматизированная разработка прикладного ПО.

К разработке программ для программируемых контроллеров обычно привлекаются не профессиональные программисты, а заводские технологи. Поэтому языки программирования должны быть достаточно простыми, обычно построенными на визуальных изображениях ситуаций. Например, используются различные схемные языки. Ряд языков стандартизован и представлен в международном стандарте IEC 1131-3.

На уровне управления технологическим оборудованием в АСУТП выполняют запуск, тестирование, выключение станков и сигнализации о неисправностях, выработку управляющих воздействий для рабочих органов программно управляемого оборудования (NC – Numerical Control). Для этого в составе технологического оборудования используются системы управления на базе встроенных контроллеров.

Функции систем управления документами и документооборотом

Перечисленные автоматизированные системы обычно работают автономно, но эффективность автоматизации будет заметно выше, если данные, генерируемые в одной системе, будут доступны в других. Для интеграции автоматизированных систем требуется создание единого информационного пространства не только на отдельных предприятиях, но и в рамках объединения предприятий, поэтому информационные технологии и автоматизированные системы управления документами и документооборотом пользуются все возрастающим вниманием среди предприятий и фирм различного профиля, поскольку организация работы с документами существенно влияет на эффективность производственных и бизнес-процессов. Такие системы имеют как самостоятельное значение, так и играют важную роль в интегрированных автоматизированных системах управления и проектирования.

Автоматизированные системы делопроизводства по своему назначению подразделяют на системы управления документами (СУД), управления документооборотом (СДО), управления знаниями (в сфере делопроизводства) и инструментальные среды делопроизводства. В соответствии с другими критериями классификации системы делопроизводства подразделяют на специализированные и комплексные, локальные и распределенные, фактографические и документографические (полнотекстовые), заказные и тиражируемые.

Системы управления документами предназначены для обеспечения санкционированного доступа к документам. Характерные функции СУД:

- ввод документов, в частности, с помощью средств их автоматического распознавания;
- индексирование документов, например, оформление регистрационных карточек с полями для атрибутов; возможно атрибутивное индексирование – к атрибутам относятся автор документа, дата создания и ключевые слова или полнотекстовое индексирование – в индекс заносят весь текст, но без предлогов и окончаний некоторых слов.
- хранение документов;
- поиск нужных данных, который может быть атрибутивным в фактографических БД или полнотекстовым в случае слабоструктурированных документов;
- поддержка групповой работы над документами;
- разграничение прав доступа к документам;
- контроль и управление версиями документов, регламентирующие внесение в них изменений;
- сбор и анализ статистических данных по параметрам документов и функционированию системы;
- подготовка отчетов.

Системы управления документооборотом служат для управления деловыми процессами прохождения и обработки документов в соответствующих подразделениях и службах организации. Характерные функции СДО:

- регистрация документов при их вхождении в систему;
- маршрутизация документов, учет их движения (маршрутизация может быть жесткой при фиксированных маршрутах или свободной); управление потоками документов обеспечивает прохождение документов по заданному маршруту с контролем внесения в них резолюций, управление внесением изменений включает систему приоритетов, средства протоколирования изменений;
- контроль исполнения предписываемых документами действий;
- защита информации при ее передаче между пунктами распределенной системы;
- автоматическое уведомление соответствующих лиц о состоянии документов и содержащихся в них директив и рекомендаций;
- планирование работ, связанных с прохождением документов.

К системам управления знаниями в области делопроизводства относят системы, выполняющие функции, характерные для интеллектуальных систем. Примеры таких функций:

- классификация документов по тем или иным признакам;
- взаимное связывание документов, например, с помощью гипертекста;
- тематический отбор документов;

- интеграция данных, поступающих из различных источников;
- аналитическая обработка данных;
- моделирование деловых процессов.

Инструментальные среды в системах делопроизводства служат для формирования систем делопроизводства, адаптированных к условиям конкретных предприятий и фирм. Часто такое формирование производится путем дополнения некоторого базового компонента, в состав системы входит соответствующий язык расширения.

Кроме перечня решаемых задач, выделяют следующие свойства и характеристики систем делопроизводства:

- открытость, программные интерфейсы и форматы данных для обмена с другими информационными системами;
- мобильность для инсталляции на ведущих платформах;
- модульное построение, что обеспечивает масштабируемость – возможность эволюционного развития, адаптируемость, возможность внедрения на предприятиях по частям.
- пользовательский интерфейс;
- быстрое действие, временные затраты на выполнение задач;
- уровень защиты информации;
- соответствие стандартам информационных технологий;
- операционные среды и используемые СУБД, требования к аппаратным ресурсам;
- перенос документов по мере их устаревания на более дешевые носители.

В крупных АСД предусматривается распределенное хранение с доступом в режимах как off-line, так и on-line. В первом случае пользователь формирует запрос в виде совокупности ключевых слов и направляет его средствами электронной почты (E-mail), СДО выдает список релевантных документов, пользователь выбирает из списка нужные документы и посылает вторичный более конкретный запрос, получая по E-mail запрошенные документы. Во втором случае используется связь в реальном времени, документ вызывается на экран компьютера и пользователь может непосредственно его просматривать и редактировать.

Современные корпоративные системы делопроизводства являются распределенными, имеющими архитектуру клиент-сервер. На серверной стороне находят применение серверы баз данных, полнотекстовых документов, электронной почты, приложений, SQL- и Web-серверы. На клиентской стороне могут выделяться рабочие места пользователей, администратора и разработчиков баз данных, информационно-поисковых систем, форм документов и т. п. В частности, применяются трехзвенные распределенные системы.

К широко известным системам документооборота и делопроизводства относятся Lotus Notes, Docs Open и др.

Лекция 3. Функциональный и структурный состав интегрированных САПР

Функциональный состав интегрированных САПР. Интерфейсы, языки, форматы межпрограммных обменов САПР. Структурный состав интегрированных САПР.

Функциональный состав интегрированных САПР

Разнообразие средств и выполняемых функций обуславливает сложность структуры САПР, в которой выделяют ряд видов функционального обеспечения. Компонентами САПР являются виды обеспечения САПР: техническое, математическое, программное, лингвистическое, информационное, методическое и организационное [6; 7].

Техническое обеспечение САПР (ТО) – совокупность аппаратных средств, используемых в САПР для переработки, хранения, передачи информации, организации общения инженера-проектировщика с вычислительными средствами, изготовления проектной документации. Основу технического обеспечения составляют ЭВМ, различные виды периферийного оборудования, средства машинной графики и пр. К техническому обеспечению САПР относят также средства организационной техники, измерительное оборудование для получения данных, используемых при проектировании.

Математическое обеспечение – совокупность математических моделей, методов, алгоритмов для решения задач автоматизированного проектирования. Математическое обеспечение реализуется в программном обеспечении САПР.

Программное обеспечение (ПО) – совокупность программ, представленных в заданной форме, вместе с необходимой программной документацией, предназначенная для использования в САПР.

Лингвистическое обеспечение – совокупность языков, используемых в САПР для представления информации о проектируемых объектах, процессе и средствах проектирования, которой обмениваются инженеры-проектировщики с ЭВМ и между собой в процессе автоматизированного проектирования (АП).

Информационное обеспечение САПР – документы, содержащие описание стандартных проектных процедур, типовых проектных решений, комплектующих изделий, материалов и другие данные.

Методическое обеспечение – документы, в которых отражены состав, правила отбора и эксплуатации средств АП.

Организационное обеспечение САПР – положения, инструкции, приказы и штатные расписания, квалификационные требования и другие документы, регламентирующие организационную структуру подразделений проектного предприятия и их взаимодействие с комплексом средств АП.

При проектировании все компоненты САПР функционируют во взаимодействии. Для проектировщика все программные и аппаратные средства выступают как единое целое, образуя инструмент проектирования.

ПО САПР есть совокупность программ и программных комплексов, независимо разработанных и автономно функционирующих или информационно связанных между собой. Во втором случае пользователь САПР имеет возможность формировать разные маршруты проектирования с помощью языка описаний заданий на проектирование.

Интерфейсы, языки, форматы межпрограммных обменов в САПР

САПР, в которых предусмотрены средства поддержки информационного интерфейса между программными комплексами и их компонентами, называются *интегрированными*. Сложность создания крупных интегрированных систем САПР определяется не столько объемом ПО и количеством составных частей, сколько наличием разветвленных информационных связей между этими частями. Межпрограммные связи характеризуются разнообразием типов, структур и объемов данных, которые являются предметом обработки в более чем одной программе и которые, следовательно, участвуют в процедурах информационных межпрограммных обменов. Разнообразны условия и интенсивность таких обменов. Поэтому проблема организации информационных межпрограммных связей в САПР является одной из основных, от ее решения зависят такие важные свойства САПР, как возможность настройки на различные маршруты проектирования, длительность выполнения этих маршрутов, открытость по отношению к вновь разрабатываемым или приобретаемым ПМК [1].

Важное значение для обеспечения открытости САПР, ее интегрируемости с другими автоматизированными системами (АС) имеют интерфейсы, представляемые реализованными в системе форматами межпрограммных обменов. Очевидно, что, в первую очередь, необходимо обеспечить связи между САЕ, САД и САМ-подсистемами.

Информационный интерфейс в ПО САПР может осуществляться через:

1. *Передачу параметров*, которая может быть по адресу или по значению. При передаче параметров по адресу в вызываемый модуль передаются адреса используемых данных (идентификаторы переменных). Увеличивается число модулей, которым доступно присваивание переменным новых значений, что может вызвать трудности в процессе разработки и сопровождения ПО. Поэтому предпочтительнее передача параметров по значению. Связь через передачу параметров удобна при сравнительно небольшом объеме передаваемых данных. В алгоритмических языках этот способ связи обеспечивается аппаратом формальных и фактических параметров процедур и функций.

2. *Связь через общие области* подразумевает выделение общей области памяти для данных, используемых многими модулями (например в языке Fortran через блок COMMON). Разработка всех модулей, связанных через

общую область, должна вестись с учетом структуры и семантики данных, помещаемых в эту область. Нельзя изменить структуру данных, не меняя всей совокупности модулей, связанных через общую область. Этот способ связи ухудшает сцепление. Кроме того, структура данных в общей области должна быть одинакова для всех модулей, что может потребовать введения в ПО информационных адаптеров – специальных модулей, приспособляющих алгоритм модуля к структуре данных в общей области.

3. *Связь через банк данных* решает проблемы информационного интерфейса многих модулей за счет унификации структур данных и информационных адаптеров. Унифицированные структуры и адаптеры выделяются в виде БД и СУБД в отдельную обслуживающую подсистему – банк данных (БНД). Разработчику ПО при наличии БНД не нужно заботиться об организации общих областей и согласовании структур, достаточно в разрабатываемых модулях ПО САПР соблюдать правила обращения к БНД.

Реализация информационных обменов между различными программами через общую БД, которая обслуживается какой-либо из имеющихся готовых СУБД, имеет существенный недостаток, а именно, универсальные СУБД созданы для их использования в АСУ и в них не учтены особенности информационного фонда САПР – разнообразие структур данных, существенная изменчивость не только значений данных, но и их структур, объемов, семантики по мере развития проекта.

Таким образом, ни один из рассмотренных подходов к организации информационного интерфейса не свободен от недостатков при интеграции ПО в САПР. Поэтому следующим шагом на пути интеграции средств САПР стало выделение наиболее характерных частей информационного фонда, участвующих в межпрограммных обменах, и установления для них унифицированных форм представления. В этих условиях любой программный продукт, созданный с учетом принятых правил представления входной и выходной информации, может взаимодействовать с другими программами, созданными с соблюдением тех же правил.

Современный подход к интеграции в САПР – введение нейтральной структуры базы данных, называемой *нейтральным файлом*, которая была бы независимой от существующих САПР. Эта структура будет действовать как промежуточная точка коммуникации между различными структурами баз данных САПР. Таким образом, когда две или более САЕ/CAD/CAM-системы объединяются или связываются в единое приложение для совместного использования данных, в каждой системе будет пара конвертеров для экспорта и импорта данных в этот нейтральный формат. Конвертор, преобразующий данные из собственного формата САПР в нейтральный формат, называется *препроцессором*, а конвертор, выполняющий обратное преобразование, – *постпроцессором*. Следовательно, для обмена данными между n системами потребуется $2n$ конверторов – *косвенный метод обмена данными* (в отличие от *прямого метода обмена данными*, где необходима пара конверторов для каждой конкретной пары САПР, т. е. для n систем необходимо $n(n-1)$ конвер-

теров). Косвенный метод свободен от присущего прямому методу недостатка, когда требовалось писать все возрастающее количество программ. Это основная причина, по которой косвенный метод принят в качестве главного метода обмена данными между различными САПР, несмотря на то что в сравнении с прямым методом у него имеются некоторые недостатки. В частности, прямые конверторы работают быстрее косвенных и создаваемые ими файлы данных имеют меньший размер, чем нейтральные файлы, генерируемые косвенными конвертерами. Файл данных в собственном формате конкретной системы также оказывается меньше нейтрального файла из-за обобщенной природы последнего. Когда переносятся данные через нейтральный файл, то, как правило, некоторая информация теряется (особенно информация о топологическом дереве и ограничениях в системах параметрического моделирования).

Очевидно, что унификация в масштабах совокупности предприятий, отрасли, а тем более в национальных или международных масштабах не может охватить все особенности конкретных САПР. Поэтому каждый из имеющихся стандартов по представлению данных – форматов нейтрального файла – имеет ту или иную предметную окраску.

Один из первых подобных международных стандартов *IGES* (Initial Graphics Exchange Specification – первоначальная спецификация обмена графическими данными) ориентирован прежде всего на обмен графической информацией [1].

IGES-файл состоит из шести разделов, которые должны идти в следующем порядке: каждый раздел состоит из записей – строк из 80 символов, запись разделена на поля. В записях можно использовать числа, текст, указатели, операторы и свободный формат. Описываются данные документирования, геометрии и свойств. S и T разделы содержат служебную информацию, позволяющую идентифицировать файл, определить его границы. В разделе Global задаются форматы данных, масштабы изображений, единицы измерения расстояний на чертежах. Основные разделы – Directory Entry (справочный) и Parameter Data (параметров). В справочном разделе описываются как геометрические, так и негеометрические элементы. Для геометрических элементов задаются типы элементов, указатели на списки параметров, типы линий, состояние видимости и т. д. Примерами геометрических элементов служат графические примитивы, сплайны, поверхности вращения, цилиндры и т. п. Описание может задаваться в виде коэффициентов уравнений линий и поверхностей или текстом, ссылками на свойства и способы интерпретации. Негеометрические элементы описываются в виде аннотаций (например, пояснения, надписи и размеры на чертежах, макроопределений, задающих информацию о графических объектах в процедурной форме, и т. п. В разделе параметров содержатся численные значения параметров.

Аналогичный стандарт, имеющий значение EDIF, разработан для представления информации о топологии и свойствах интегральных схем.

При использовании препроцессоров и постпроцессоров с нейтральным форматом IGES на практике возникают проблемы.

1. Внутренний способ представления элемента в системе может отличаться от того, как этот элемент представлен в IGES. Например, дуга окружности может быть определена через центр, радиус, начальный и конечный углы, но в IGES она определяется через центр, начальную и конечную точки. Таким образом, специализированный IGES-конвертор должен выполнить преобразование с использованием параметрического уравнения дуги. Такое преобразование должно выполняться дважды (прямая и обратная конвертации), и каждый раз значения параметров дуги искажаются из-за ошибок усечения и округления.

2. Если элемент не поддерживается явно, то его необходимо преобразовать в ближайший по форме доступный элемент. При этом происходит потеря информации.

Формат *DXF* (Drawing interchange Format – формат обмена чертежами) разрабатывался для управления данными и преобразования чертежей программы AutoCAD в форматы файлов, которые могли читаться и использоваться другими САПР. Из-за популярности AutoCAD формат DXF стал стандартом обмена файлами CAD-чертежей почти для всех САПР. DXF-файл – это текстовый ASCII-файл, состоящий из пяти разделов. В разделе Header описывается среда AutoCAD, в которой был создан файл. В разделе Table представлена информация о типах линий, слоях, стилях текста и видах, которые могут быть определены на чертеже. В разделе Block содержится список графических элементов, определенных как группа. Конкретные данные по каждому элементу хранятся в разделе Entity. Это главный раздел DXF-файла, в котором описываются все элементы, присутствующие на чертеже. С появлением новых версий AutoCAD список возможных элементов файла расширялся.

Для связей между программами, взаимодействующими на маршрутах проектирования, стандарты типа IGES часто бывают малоудобными. Это обусловлено двумя причинами [1].

1. В этих стандартах регламентируются форматы только тех частей информационного фонда, которые имеют для различных САПР общий характер (описание графических документов).

2. Форма задания информации в унифицированных форматах должна быть инвариантна относительно конкретных классов проектируемых объектов, языков программирования. Такая инвариантность приводит к тому, что для расшифровки сведений при их использовании требуется значительное количество времени. Если интенсивность обменов между программами велика, то накладные расходы могут оказаться недопустимо большими.

Поэтому наличие форматов типа IGES – это решение лишь части проблемы интеграции ПО в конкретных САПР. Данную проблему можно снять, создавая программные среды, в которых приняты договоренности не только о фиксации форматов внутреннего представления данных (ВПД), но и действуют определенные методики интеграции.

Концепция ВПД есть концепция специализированных СУБД, учитывающих особенности БД в САПР. Специфика САПР проявляется, главным образом, в БД проектов, содержащих информацию о текущем состоянии проектов, промежуточных проектных решениях, получаемых описаниях. Эта информация может выражать ТЗ, графические документы, текстовые пояснения, результаты расчетов в форме таблиц и графиков, описания алгоритмов и процедур и т. п. Ускорение межпрограммных обменов достигается за счет специализации средств управления по группам данных, программ или участков маршрутов, упрощающей доступ к данным. При высокой интенсивности взаимодействия обмены совершаются через общую область памяти. При этом должна предусматриваться возможность управления составом данных, направляемых в общую область, на уровне входных языков взаимодействующих программ. Например, при взаимодействии программы 1 моделирования переходных процессов и программы 2 расчета выходных параметров функционалов из программы 1 и программы 2 должны передаваться лишь некоторые фазовые переменные, указание которых целесообразно предусмотреть на входном языке программы 1. Альтернативный вариант связан с пересылкой в общую область всего массива фазовых переменных.

Идеи IGES по унификации межсистемных интерфейсов получили развитие в стандарте STEP – формат данных, используемый для хранения полной информации обо всем жизненном цикле продукта, включая проектирование, анализ, производство, контроль качества. В основе разработки STEP лежат следующие принципы:

- Стандарт STEP ориентируется на данные о продукте, включающие информацию обо всем жизненном цикле продукта, т. е. информацию о допусках, технологических особенностях формы, конечноэлементную модель, модель для кинематического анализа и т. д.
- В структурах данных STEP информация, относящаяся к приложению, должна храниться в модуле уровня приложения, отдельно от общей информации о форме, что позволяет поддерживать широкий спектр приложений, избегая избыточности в общей структуре данных.
- Для определения структуры данных должен использоваться формальный язык. Спецификации IGES и DXF описывают формат физического файла, в котором хранятся все геометрические и прочие данные. В STEP данные описываются на языке EXPRESS, а затем результат преобразовывается в физический файл, что позволяет избежать неоднозначностей при интерпретации данных о продукте, извлеченных из файла.

STEP разрабатывается рядом комитетов и рабочих групп, занимающихся разными частями стандарта. Эти части группируются по методам описания, интегрированным информационным ресурсам, прикладным протоколам, методам реализаций и методологией согласования.

STEP привлекает к себе повышенное внимание, так как ожидается, что он войдет в систему стандартов технологий CALS (Computer-aided Acquisition and Logistics Support – непрерывные поставки и информационная поддержка жизненного цикла продукции) как стандарт обмена данными о продуктах. CALS становится промышленным стандартом хранения и обмена компьютерными данными в организации.

Основным языком стандарта STEP является язык Express и его диалекты, что объясняется его общим характером и направленностью на различные приложения, а также на использование в современных распределенных проектных и производственных системах. Действительно, такие форматы, как IGES или DXF, описывают только геометрию объектов, в то время как в обменах между различными САПР и их подсистемами фигурируют данные о различных свойствах и атрибутах изделий.

Язык Express [2] используется во многих системах интерфейса между CAD/CAM-системами. В частности, в систему CAD++STEP включена среда SDAI (Standard Data Access Interface), в которой возможно представление данных об объектах из разных систем CAD и приложений (но описанных по правилам языка Express). CAD++STEP обеспечивает доступ к базам данных большинства известных САПР с представлением извлекаемых данных в виде STEP-файлов. Интерфейс программиста позволяет открывать и закрывать файлы проектов в базах данных, производить чтение и запись сущностей. В качестве объектов могут использоваться точки, кривые, поверхности, текст, примеры проектных решений, размеры, связи, типовые изображения, комплексы данных и т. п.

Структурный состав интегрированных САПР

Составными *структурными частями САПР* являются подсистемы и комплексы [6].

Подсистемы САПР обладают всеми свойствами систем и создаются как самостоятельные. Подсистема САПР – выделенная по некоторым признакам часть САПР, обеспечивающая получение законченных проектных решений. По назначению подсистемы САПР разделяют на проектирующие и обслуживающие. Проектирующие подсистемы САПР – это подсистемы, выполняющие проектные процедуры. Например, подсистема конструкторского проектирования, подсистема технологического проектирования, подсистема проектирования деталей и сборочных единиц и т. п. Обслуживающие подсистемы САПР – это подсистемы, обеспечивающие работоспособность проектирующих подсистем. Их совокупность часто называют *системной средой* (или *оболочкой*) САПР. Типичными обслуживающими подсистемами являются подсистемы управления проектными данными (PDM), управления процессом

проектирования (DesPM – Design Process Management), пользовательского интерфейса для связи разработчиков с ЭВМ, CASE (Computer Aided Software Engineering) для разработки и сопровождения программного обеспечения САПР, обучающие подсистемы для освоения пользователями технологий, реализованных в САПР.

Промежуточное положение между обслуживающими и проектирующими подсистемами в большинстве САПР занимает подсистема машинной графики.

По отношению к объекту проектирования различают объектно-ориентированные и объектно-независимые подсистемы. К первым относят подсистемы, выполняющие одну или несколько проектных процедур или операций, непосредственно зависящих от конкретного объекта проектирования. Ко вторым относят подсистемы, выполняющие унифицированные проектные процедуры и операции, например, функции обработки, независящие от особенностей проектируемого объекта.

Понятие подсистемы САПР близко к понятию комплекса. В САПР выделяют программно-методический комплекс (ПМК) и программно-технический комплекс (ПТК). ПМК – взаимосвязанная совокупность некоторых частей программного, методического и информационного обеспечения, необходимая для получения законченного проектного решения по объекту проектирования. Выделяют ПМК оформления документации, синтеза проектных решений, моделирования и т. п. ПТК – взаимосвязанная совокупность программно-методических комплексов, объединенных по некоторому признаку, и средств технического обеспечения САПР. Понятие ПТК относится к вычислительным системам, объединяющим аппаратные и программные средства и предназначенные для применения в САПР. Примером ПТК может служить автоматизированное рабочее место (АРМ), включающее ЭВМ, периферийные устройства и ряд ПМК для выполнения проектных маршрутов и процедур.

ПМК и ПТК представляют собой промышленный продукт, разрабатываемый, изготавливаемый и поставляемый для создания или развития САПР на предприятиях заказчиков.

МОДУЛЬ 2. МОДЕЛИ И МЕТОДЫ АНАЛИЗА ССУ ПРИ АВТОМАТИЗАЦИИ ЭТАПА ПРОЕКТИРОВАНИЯ

ТЕМА 3. МОДЕЛЬНОЕ ПРЕДСТАВЛЕНИЕ СРЕДСТВ И СИСТЕМ УПРАВЛЕНИЯ (ССУ)

Лекция 4. Модельное представление систем управления и элементов ССУ как объектов проектирования

Модельное представление СУ и элементов ССУ как объектов проектирования. Классификация моделей СУ как объектов проектирования. Этапы математического моделирования СУ. Математические модели систем управления. Математические модели устройств СУ. Математические модели элементов устройств СУ.

Модельное представление СУ и элементов ССУ как объектов проектирования

Модельное представление систем управления и элементов ССУ при автоматизированном проектировании является частью математического обеспечения САПР и составляет основу математического обеспечения CALS-технологий в рамках единой информационной модели ССУ как объекта проектирования. Разработка обобщенной математической модели ССУ является очень трудоемкой задачей, решение которой в рамках блочно-иерархического подхода к проектированию состоит в формировании математического описания ССУ для функционального, конструкторского и технологического проектирования и интеграции каждого из описаний в единую информационную модель. В данной лекции будем рассматривать математическое описание ССУ для этапа функционального проектирования, как базового компонента математического обеспечения САЕ-систем.

Понятие математической модели (ММ), как и ряд других понятий, используемых в математическом моделировании, не имеет строгого формального определения. Тем не менее в это понятие вкладывают вполне конкретное содержание, с которым, в частности, тесно связано применение математики в инженерной практике. Более того, такие научные дисциплины, как механика, физика и их многочисленные разделы, являются по существу упорядоченными множествами ММ, построение которых сопровождается теоретическим обоснованием адекватного отражения этими моделями свойств

рассматриваемых процессов и явлений. Именно посредством ММ научные дисциплины взаимодействуют с математикой.

В достаточно общем случае изучаемую техническую систему (ТС) и СУ, в частности, количественно можно охарактеризовать векторами внешних Q , внутренних X и выходных Y параметров соответственно. Одни и те же физические, механические или информационные характеристики ТО в моделях различного уровня и содержания могут выполнять роль как внешних или внутренних, так и выходных параметров.

Например, для электронного усилителя выходными параметрами являются коэффициент усиления, полоса частот пропускаемых сигналов, входное сопротивление, рассеиваемая мощность, внешними – сопротивление и емкость нагрузки, напряжения источников питания, температура окружающей среды, а внутренними – сопротивления резисторов, емкости конденсаторов, характеристики транзисторов. Но если в качестве ТС рассматривать отдельно взятый транзистор, то такие его характеристики, как отпирающее напряжение и коллекторный ток, следует уже отнести к его выходным параметрам, а в качестве внешних надо будет рассматривать токи и напряжения, задаваемые коммутирующими с ним элементами усилителя.

При создании ТС значения выходных параметров или диапазоны их возможного изменения оговаривают в техническом задании на разработку ТС, тогда как внешние параметры характеризуют условия ее функционирования. В сравнительно простом случае математическая модель (ММ) ТС может представлять собой соотношение [12]:

$$Y = F(X, Q). \quad (4.1)$$

Теоретический путь построения ММ состоит в установлении связи между Y , X и Q в виде операторного уравнения

$$L(V(Z)) = 0, \quad (4.2)$$

где L – некоторый оператор (в общем случае нелинейный), 0 – нулевой элемент пространства, в котором действует этот оператор, Z – вектор независимых переменных, в общем случае включающий время и пространственные координаты, а V – вектор фазовых переменных, включающий те параметры ТС, которые характеризуют ее состояние. Но даже если возможно получить решение (4.2) и найти зависимость V от Z , то далеко не всегда удастся представить ММ ТС в явном относительно вектора V виде. Поэтому именно (4.2) определяет в общем случае структуру ММ ТО, а (4.1) является более простым частным случаем такой модели.

Различные особенности и признаки математических моделей (ММ) лежат в основе их типизации (или классификации). Среди таких признаков выделяют характер отображаемых свойств технической системы, степень их детализации, способы получения и представления ММ.

Классификация моделей СУ как объектов проектирования

Один из существенных признаков классификации связан с отражением в ММ тех или иных особенностей ТС. Если ММ отображает устройство ТС и связи между составляющими его элементами, то ее называют структурной математической моделью. Если же ММ отражает происходящие в ТС физические, механические, химические или информационные процессы, то ее относят к функциональным математическим моделям. Ясно, что могут существовать и комбинированные ММ, которые описывают как функционирование, так и устройство ТС. Такие ММ естественно называть структурно-функциональными математическими моделями [6; 7; 12].

Структурные ММ делят на топологические и геометрические, составляющие два уровня иерархии ММ этого типа. Первые отображают состав ТС и связи между ее элементами. *Топологическую* ММ целесообразно применять на начальной стадии исследования сложной по структуре ТС, состоящего из большого числа элементов, прежде всего для уяснения и уточнения их взаимосвязи. Такая ММ имеет форму графов, таблиц, матриц, списков и т. п., и ее построению обычно предшествует разработка структурной схемы ТС.

Геометрическая ММ дополнительно к информации, представленной в топологической ММ, содержит сведения о форме и размерах ТС и ее элементах, об их взаимном расположении в пространстве. В геометрическую ММ обычно входят совокупность уравнений линий и поверхностей и алгебраические соотношения, определяющие принадлежность областей пространства телу ТС или ее элементам. Такую ММ иногда задают координатами некоторого множества точек, по которым интерполированием можно построить ограничивающую область линии или поверхности. Границы области задают и кинематическим способом: линию – как траекторию движения точки, а поверхность – как результат перемещения линии. Возможно представление формы и размеров области совокупностью типовых фрагментов достаточно простой конфигурации. Такой способ характерен, например, для метода конечных элементов, широко используемого в математическом моделировании.

Геометрические ММ находят применение при конструкторском проектировании ТС в САД/САМ-системах, разработке технической документации и технологических процессов изготовления деталей (например, на станках с числовым программным управлением).

Функциональные ММ состоят из соотношений, связывающих между собой фазовые переменные, т. е. внутренние, внешние и выходные параметры ТС. Функционирование сложных ТС нередко удается описать лишь при помощи совокупности его реакций на некоторые известные (или заданные) входные воздействия (сигналы). Такую разновидность функциональной ММ относят к типу черного ящика и обычно называют имитационной математической моделью, имея в виду, что она лишь имитирует внешние проявления функционирования ТС, не раскрывая и не описывая существа протекающих в нем процессов. Имитационные ММ находят широкое применение в техни-

ческой кибернетике – научном направлении, изучающем системы управления сложными объектами.

По способу получения математические модели (ММ) делят на *теоретические и эмпирические*. Первые получают в результате изучения свойств системы и протекающих в ней процессов, а вторые – это итог обработки результатов наблюдения внешних проявлений этих свойств и процессов. Один из способов построения эмпирических ММ заключается в проведении экспериментальных исследований, связанных с измерением фазовых переменных и в последующем обобщении результатов этих измерений в алгоритмической форме или в виде аналитических зависимостей. Поэтому эмпирическая ММ по форме представления может содержать признаки как алгоритмической, так и аналитической математической модели. Следовательно, построение эмпирической ММ сводится к решению задачи идентификации.

При построении теоретических ММ прежде всего стремятся использовать известные фундаментальные законы сохранения таких субстанций, как масса, электрический заряд, энергия, количество движения и момент количества движения. Кроме того, привлекают определяющие соотношения (или уравнения состояния, в роли которых могут выступать так называемые феноменологические законы (например, уравнение Клапейрона – Менделеева – состояния совершенного газа, закон Ома о связи силы тока в проводнике и падения электрического напряжения, закон Гука о связи деформации и механического напряжения в линейно упругом материале, закон Фурье о связи градиента температуры в теле с плотностью теплового потока и т. п.).

Сочетание теоретических соображений качественного характера с обработкой результатов наблюдения внешних проявлений свойств изучаемой системы и ее элементов приводит к смешанному типу ММ, называемых *полумэмпирическими*. При построении таких ММ используют основные положения теории размерностей, в том числе так называемую П-теорему (Питереми): если между n -параметрами, характеризующими изучаемый объект, существует зависимость, имеющая физический смысл, то эту зависимость можно представить в виде зависимости между $n - k$ их безразмерными комбинациями, где k – число независимых единиц измерения, через которые можно выразить размерности этих параметров. При этом n определяет число независимых (не выражаемых друг через друга) безразмерных комбинаций, обычно называемых критериями подобия.

Объекты, для которых равны значения соответствующих критериев подобия, считают подобными. Например, любой треугольник однозначно определен длинами a , b и c его сторон, т. е. $n = 3$, а $k = 1$. Поэтому, согласно П-теореме, множество подобных треугольников можно задать значениями $n - k = 2$ критериев подобия.

Для успешного применения П-теоремы к построению моделей технических систем необходимо располагать полным набором параметров, описы-

вающих изучаемый объект, причем выбор этих параметров должен опираться на аргументированный качественный анализ тех свойств и особенностей, влияние которых существенно в данном конкретном случае. Такой анализ необходим при любом способе построения ММ.

Этапы математического моделирования СУ

Для обсуждения и обоснования основных подходов к разработке проблем математического моделирования средств и систем управления и процессов в них представляется целесообразным предварительно рассмотреть условную схему (рис. 4.1), определяющую последовательность проведения отдельных этапов общей процедуры вычислительного эксперимента [12].

Исходной позицией этой схемы служит технический объект (ТО), под которым будем понимать конкретное техническое устройство, его агрегат или узел, систему устройств, процесс, явление или отдельную ситуацию в какой-либо системе или устройстве.

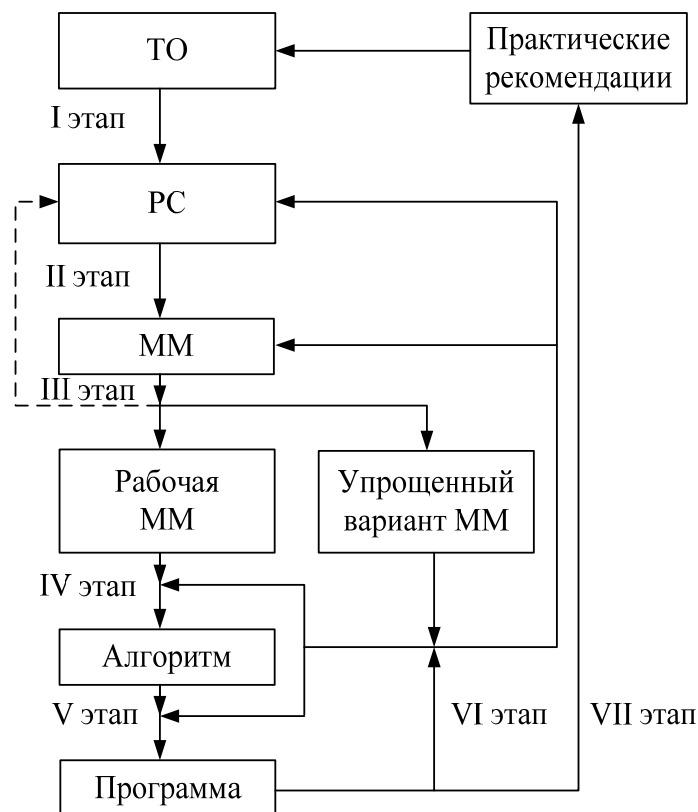


Рис. 4.1

На первом этапе осуществляют неформальный переход от рассматриваемого (разрабатываемого или существующего) ТО к его расчетной схеме (РС).

При этом в зависимости от направленности вычислительного эксперимента и его конечной цели акцентируют те свойства, условия работы и осо-

бенности ТО, которые вместе с характеризующими их параметрами должны найти отражение в РС, и, наоборот, аргументируют допущения и упрощения, позволяющие не учитывать в РС те качества ТО, влияние которых предполагают в рассматриваемом случае несущественным. Иногда вместо РС используют термин «содержательная модель ТО», а в некоторых случаях – «концептуальная модель». В сложившихся инженерных дисциплинах (например, в сопротивлении материалов, электротехнике и электронике) помимо описательной (вербальной) информации для характеристики РС разработаны специальные приемы и символы наглядного графического изображения. По ряду новых направлений развития техники подобная символика находится в стадии формирования [11].

При разработке новых ТО успешное проведение первого этапа в значительной мере зависит от профессионального уровня инженера, его творческого потенциала и интуиции. Полнота и правильность учета в РС свойств ТО, существенных с точки зрения поставленной цели исследования, являются основной предпосылкой получения в дальнейшем достоверных результатов математического моделирования. И наоборот, сильная идеализация ТО ради получения простой РС может обесценить все последующие этапы работы.

Содержание второго этапа состоит в формальном, математическом описании РС. Это описание в виде математических соотношений, устанавливающих связь между параметрами, характеризующими РС ТО, и называют математической моделью (ММ).

Для некоторых типовых РС существуют банки ММ, что упрощает проведение второго этапа. Более того, одна и та же ММ может соответствовать РС из различных предметных областей. Однако при разработке новых ТО часто не удается ограничиться применением типовых РС и отвечающих им уже построенных ММ. Создание новых ММ или модификация существующих должны опираться на достаточно глубокую математическую подготовку и владение математикой как универсальным языком науки.

На третьем этапе проводят качественный и оценочный количественный анализ построенной ММ. При этом могут быть выявлены противоречия, ликвидация которых потребует уточнения или пересмотра РС (штриховая линия на [рис. 4.1](#)). Количественные оценки могут дать основания упростить модель, исключив из рассмотрения некоторые параметры, соотношения или их отдельные составляющие, несмотря на то, что влияние описываемых ими факторов учтено в РС. В большинстве случаев, принимая дополнительные по отношению к РС допущения, полезно построить такой упрощенный вариант ММ, который позволял бы получить или привлечь известное точное решение. Это решение затем можно использовать для сравнения при тестировании результатов на последующих этапах. В некоторых случаях удается построить несколько ММ для одного и того же ТО, отличающихся различным уровнем упрощения. В этом случае говорят об иерархии ММ, что означает упорядочение ММ по признаку их сложности и полноты.

Построение иерархии ММ связано с различной детализацией свойств изучаемого ТО. Сравнение результатов исследования различных ММ может существенно расширить и обогатить знания об этом ТО. Кроме того, такое сравнение позволяет оценить достоверность результатов последующего вычислительного эксперимента: если более простая ММ правильно отражает некоторые свойства ТО, то результаты исследования этих свойств должны быть близки к результатам, полученным при использовании более полной и сложной ММ.

Итог анализа на рассматриваемом этапе – это обоснованный выбор рабочей ММ ТО, которая подлежит в дальнейшем детальному количественному анализу. Успех в проведении третьего этапа зависит, как правило, от глубины понимания связи отдельных составляющих ММ со свойствами ТО, нашедшими отражение в его РС, что предполагает органическое сочетание владения математикой и инженерными знаниями в конкретной предметной области.

Четвертый этап состоит в обоснованном выборе метода количественного анализа ММ, в разработке эффективного алгоритма вычислительного эксперимента, а пятый этап – в создании работоспособной программы, реализующей этот алгоритм средствами вычислительной техники. Для успешного проведения четвертого этапа необходимо владеть арсеналом современных методов вычислительной математики, а при математическом моделировании довольно сложных ТО выполнение пятого этапа требует профессиональной подготовки в области программирования на ЭВМ.

Получаемые на шестом этапе (в итоге работы программы) результаты вычислений должны прежде всего пройти тестирование путем сопоставления с данными количественного анализа упрощенного варианта ММ рассматриваемого ТО. Тестирование может выявить недочеты как в программе, так и в алгоритме и потребовать доработки программы или же модификации и алгоритма, и программы. Анализ результатов вычислений и их инженерная интерпретация могут вызвать необходимость в корректировке РС и соответствующей ММ. После устранения всех выявленных недочетов триаду «модель – алгоритм – программа» можно использовать в качестве рабочего инструмента для проведения вычислительного эксперимента и выработки на основе получаемой количественной информации практических рекомендаций, направленных на совершенствование ТО.

Представленная последовательность этапов носит общий и универсальный характер, хотя в некоторых конкретных случаях она может и несколько видоизменяться. Если при разработке ССУ можно использовать типовые РС и ММ, то отпадает необходимость в выполнении ряда этапов, а при наличии и соответствующего программного комплекса процесс проектирования становится в значительной степени автоматизированным. Однако математическое моделирование объектов, не имеющих близких прототипов,

как правило, связано с проведением всех этапов описанного «технологического цикла».

Для математического описания СУ можно также классифицировать математические модели по степени детализации СУ как сложной системы в соответствии с трехуровневым представлением (рис. 4.2, а): ММ всей СУ как сложной системы Р; ММ устройств СУ – средств управления – как отдельных систем A_i ; ММ элементов этих устройств как подсистем B_{ij} [4].

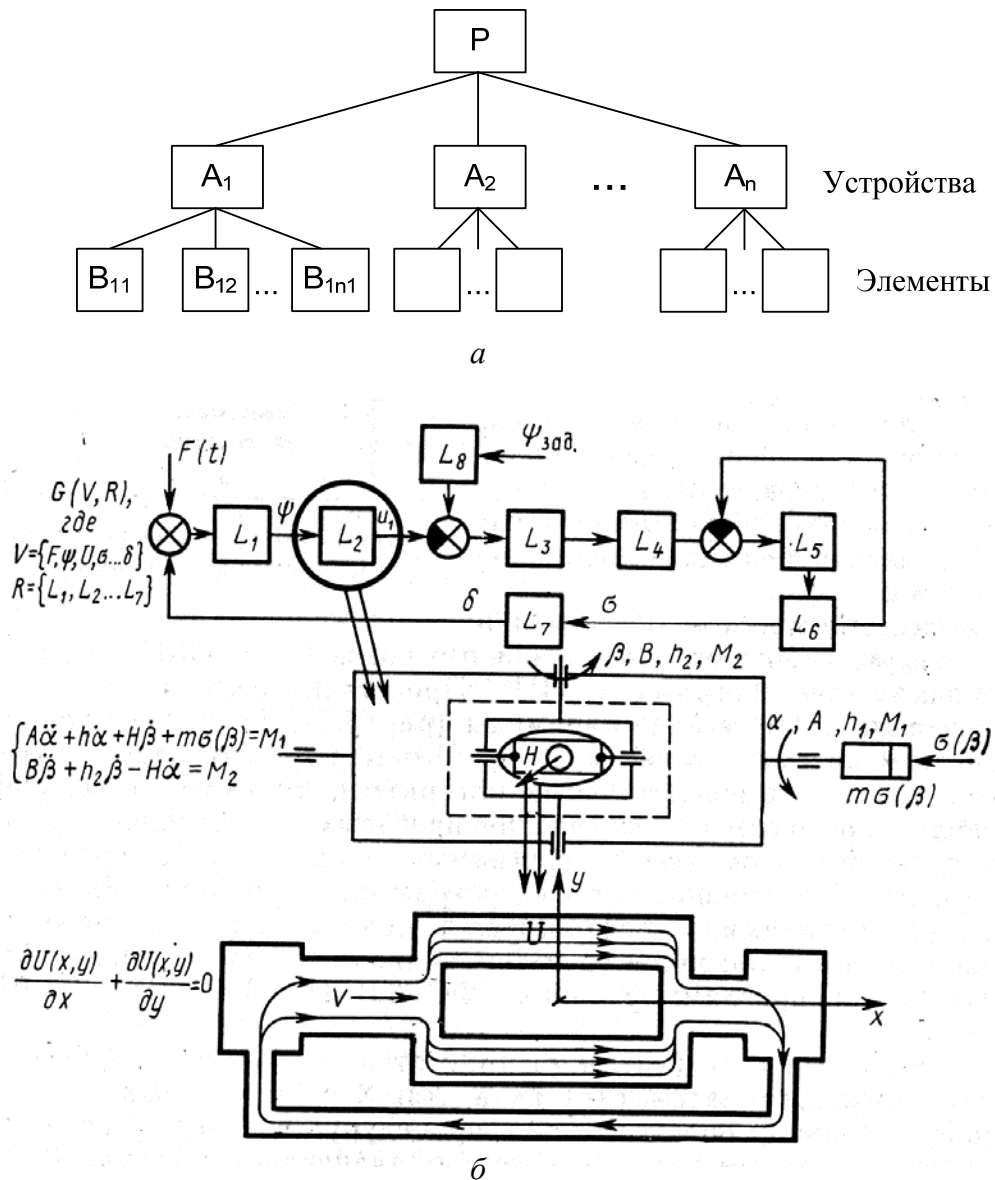


Рис. 4.2

На рис. 4.2, б для иллюстрации такой иерархии ММ показаны уровни и соответствующие формы ММ: для САУ ЛА, например, гиростабилизатора) и чувствительного элемента гиростабилизатора – гироблока с гироскопом на «газовом подвесе».

Математические модели систем управления

На самом верхнем уровне абстрагирования наибольшее распространение в теории управления имеют ММ СУ, полученные в виде структурных схем и графов [13]. Структурной схемой СУ называют ее графическое изображение в виде соединений звеньев, в которых осуществляется преобразование входного сигнала в выходной. Каждое звено моделирует реальные устройства СУ, при этом не всегда разделение звеньев в структуре системы соответствует «естественному». Некоторые естественные звенья «математически» объединяются в одно, а другие, наоборот, разделяются для удобства дальнейших исследований СУ. Звенья систем управления делят на три группы: линейные безынерционные звенья, нелинейные безынерционные звенья, линейные инерционные звенья.

На основе структурных схем проводят исследования СУ в целом: формируется ее облик и функциональная схема, осуществляется выбор устройств.

Удобным для исследования отображением структурных схем СУ являются ориентированные графы, которые применительно к СУ обладают следующими свойствами:

- дуги графа изображают звено и характеризуются оператором этого звена – передаточная функция, уравнение звена;
- каждой вершине ставится в соответствие одна из переменных. Согласно с правилами работы с графами вершина, к которой подходит одна или несколько дуг, соответствует переменной, равной выходу одной дуги или сумме выходов дуг. Если из вершины исходит несколько дуг, то входная величина для всех дуг одна и та же.

Также на первом (верхнем при нисходящем проектировании) уровне абстрагирования (рис. 4.2) используется описание СУ в пространстве состояний для исследования управляемости и наблюдаемости многомерных систем, которые в отличие от одномерных имеют несколько входов и несколько выходов.

Для описания таких систем используются три набора параметров (три вектора) (рис. 4.3): вектор входных воздействий (управлений); вектор переменных состояний; вектор выходных параметров. Кроме того, используются два преобразования: преобразование «входы-состояния»; преобразование «состояния-выходы».

Широкое распространение, обусловленное разработанным математическим аппаратом, получили линейные модели многомерных систем в пространстве состояний, которые имеют вид

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t); \\ y(t) &= C(t)x(t); \end{aligned} \quad (4.3)$$

первое соотношение называется уравнением состояния, второе – уравнением выхода. Здесь $x = (x_1, x_2, \dots, x_n)^T \in R^n$ – вектор переменных состояний; $u = (u_1, u_2, \dots, u_r)^T \in U \subseteq R^n$ – вектор управлений; $y = (y_1, y_2, \dots, y_m)^T \in R^n$ – вектор измеряемых параметров; t – время; $A(t)$, $B(t)$, $C(t)$ – матрицы размерности $(n \times n)$, $(n \times r)$, $(m \times n)$ соответственно. Предполагается, что известны начальные состояния $x(t_0) = x_0$, где t_0 – начальный момент времени.

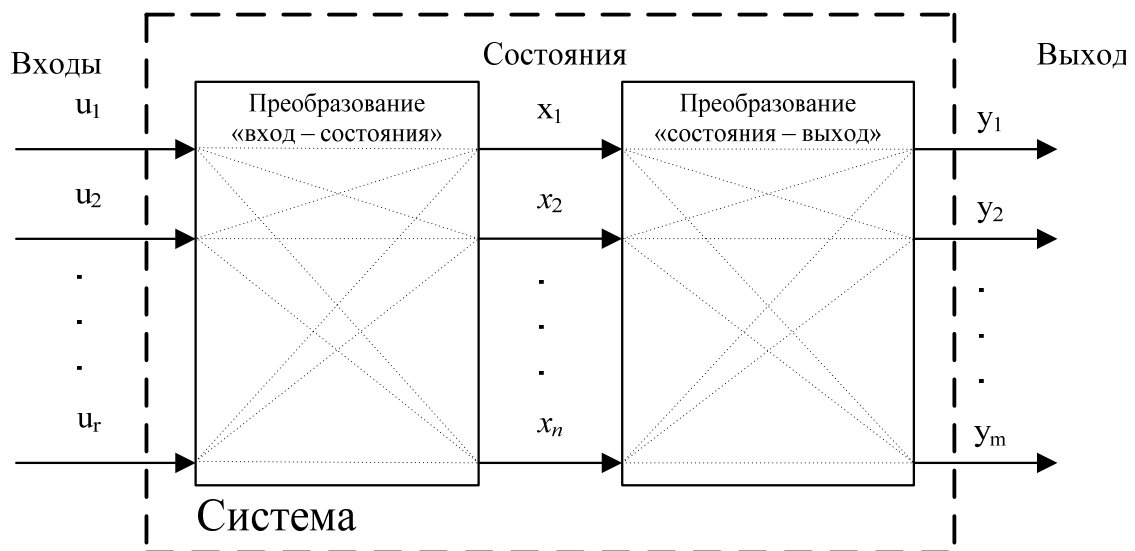


Рис. 4.3

Если матрицы $A(t)$, $B(t)$, $C(t)$ не зависят от времени t , то система называется стационарной. Далее предполагается, что системы стационарны.

Рассмотрим задачи соединения двух подсистем в систему. При соединении возможны три варианта (рис. 4.4): параллельное (а), последовательное (б) и в обратной связи (в). Предполагается, что обе системы описываются в пространстве состояний соотношениями

$$\begin{aligned} \dot{x}^1 &= A_1 x^1 + B_1 u^1; \quad y_1 = C x^1; \\ \dot{x}^2 &= A_2 x^2 + B_2 u^2; \quad y_2 = C x^2; \end{aligned}$$

где x^1, u^1, y_1 – векторы состояний, управлений, выходов первой системы, x^2, u^2, y_2 – второй. Необходимо по известным матрицам $A_1, B_1, C_1, A_2, B_2, C_2$ получить матрицы A, B, C (рис. 4.4, з).

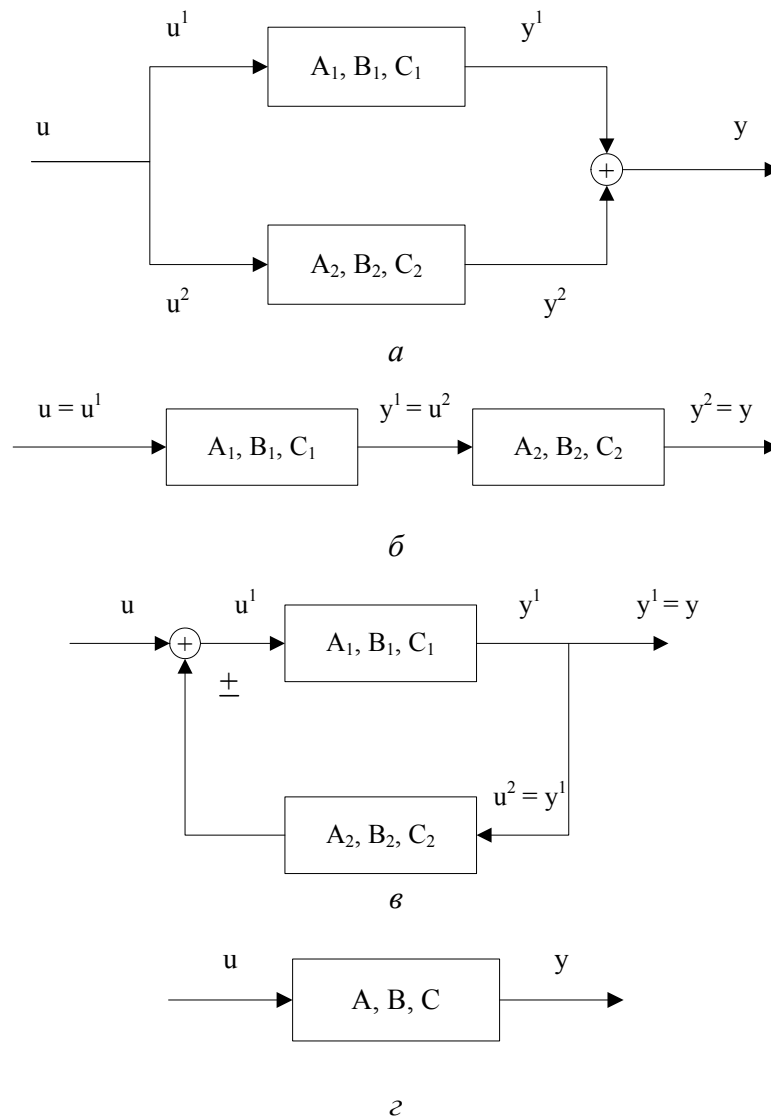


Рис. 4.4

Параллельное соединение. Уравнения системы, с учетом особенностей соединения, указанных на [рис. 4.4, а](#), записываются как

$$\begin{aligned} \dot{x}^1 &= A_1 x^1 + B_1 u; \\ \dot{x}^2 &= A_2 x^2 + B_2 u; \\ y &= C_1 x^1 + C_2 x^2; \end{aligned}$$

отсюда

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} &= \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u; \\ y &= (C_1 \quad C_2) \begin{pmatrix} x^1 \\ x^2 \end{pmatrix}. \end{aligned}$$

Окончательно матрицы соединения имеют вид –

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}; \quad B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}; \quad C = (C_1 \quad C_2).$$

Последовательное соединение –

$$\begin{aligned} \dot{x}^1 &= A_1 x^1 + B_1 u; \\ \dot{x}^2 &= A_2 x^2 + B_2 C_1 x^1; \\ y &= C_2 x^2; \end{aligned}$$

в матричном виде –

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} &= \begin{pmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} + \begin{pmatrix} B_1 \\ 0 \end{pmatrix} u; \\ y &= (0 \quad C_2) \begin{pmatrix} x^1 \\ x^2 \end{pmatrix}; \end{aligned}$$

окончательно имеем

$$A = \begin{pmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{pmatrix}; \quad B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix}; \quad C = (0 \quad C_2).$$

Обратная связь –

$$\begin{aligned} \dot{x}^1 &= A_1 x^1 + B_1 u \pm B_1 C_2 x^2; \\ \dot{x}^2 &= A_2 x^2 + B_2 C_1 x^1; \\ y &= C_1 x^1; \end{aligned}$$

в матричном виде –

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} A_1 & \pm B_1 C_2 \\ B_2 C_1 & A_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} B_1 \\ 0 \end{pmatrix} u; \\ y &= (C_1 \quad 0) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \end{aligned}$$

Следовательно,

$$A = \begin{pmatrix} A_1 & \pm B_1 C_2 \\ B_2 C_1 & A_2 \end{pmatrix}; \quad B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix}; \quad C = (C_1 \quad 0).$$

Для линейных систем легко показать справедливость следующего результата, называемого принципом суперпозиции: эффект, вызываемый суммой нескольких воздействий, равен сумме нескольких воздействий, равен сумме эффектов от нескольких воздействий в отдельности. Закон изменения вектора состояний линейной системы представляется в виде суммы свободного и вынужденного колебания

$$x(t) = x_c(t) + x_b(t).$$

Свободное движение $x_c(t)$ происходит при отсутствии внешнего воздействия в ненулевых начальных условиях [4]. Оно определяется решением однородной системы уравнений, соответствующей исходному уравнению состояний

$$\dot{x}(t) = A(t)x(t)$$

с начальными условиями $x(t_0) = x_0$.

Вынужденное движение $x_b(t)$ – это реакция системы на внешнее воздействие $u(t)$ при нулевых начальных условиях. Оно определяется решением неоднородного уравнения при нулевых начальных условиях.

Для многомерных нестационарных систем, описываемых соотношениями, поведение векторов состояния и выхода определяется по формулам

$$x(t) = \Phi(t, t_0)x(t_0) + \int_{t_0}^t \Phi(t, \tau)B(\tau)u(\tau)d\tau, \quad (4.4)$$

$$y(t) = C(t)\Phi(t, t_0)x(t_0) + \int_{t_0}^t C(t)\Phi(t, \tau)B(\tau)u(\tau)d\tau, \quad (4.5)$$

где $\Phi(t, \tau)$ – переходная матрица, или матрица Коши, являющаяся решением уравнения

$$\frac{\partial \Phi(t, \tau)}{\partial t} = A(t)\Phi(t, \tau) \quad (4.6)$$

с начальным условием $\Phi(\tau, \tau) = E$.

Первые слагаемые в (4.4), (4.5) описывают свободное движение, а вторые – вынужденное.

Для многомерных стационарных систем, описываемых уравнениями (4.3), законы изменения вектора состояния и вектора выхода находятся по формулам

$$x(t) = \Phi(t)x(0) + \int_{t_0}^t \Phi(t-\tau)Bu(\tau)d\tau,$$

$$y(t) = C\Phi(t)x(0) + \int_{t_0}^t C\Phi(t-\tau)Bu(\tau)d\tau,$$

где $\Phi(t - \tau)$ – переходная матрица стационарной системы, зависящая от разности $t - \tau$. В данном случае решение уравнения (4.3) имеет вид

$$\Phi(t, \tau) = \Phi(t - \tau) = \exp[A(t - \tau)].$$

Для проектирования структурных схем СУ удобно применять также аппарат теории систем массового обслуживания. Системы массового обслуживания (СМО) – системы, назначением которых является обслуживание поступающих в систему заявок.

В СМО различают: статические объекты – обслуживающие аппараты (ресурсы или каналы), которые моделируют средства обработки информации, например, ЭВМ, аппаратура передачи данных, аппаратные и программные средства; динамические объекты – транзакты или заявки, которые моделируют решаемые в системе задачи. Если с помощью СМО моделируются производственные линии, то технологическое оборудование отображается в виде обслуживающего аппарата (ОА), а обрабатываемые детали отображаются в виде заявок.

Схема простейшей СМО приведена на рис. 4.5 [6], где ГЗ – генератор заявок, О – очередь, ОА – обслуживающий аппарат.

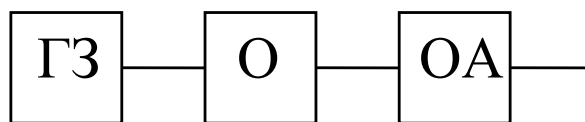


Рис. 4.5

Задача анализа СМО формируется как определение выходных параметров системы при заданных внутренних (ОА) и внешних (заявок) параметрах. Выходными параметрами системы могут быть производительность, быстродействие, среднее и максимальное время обслуживания заявок и т. д.

Основные типы СМО:

- в зависимости от числа ОА параллельно обрабатывающих входной поток заявок: одноканальные, многоканальные;
- от числа последовательно включенных ОА: однофазные, многофазные;

- замкнутая СМО – с циркуляцией постоянного числа заявок внутри системы;
- сеть СМО – структуры из ОА, соединенные произвольным образом.

Состояние СМО характеризуется вектором фазовых переменных её элементов. Функционирование СМО представляется как процесс прохождения заявок через систему. При этом заявка может войти в некоторый ОА для обслуживания. Состояние ОА представлено двоичной величиной $W = \{\text{«свободен»}, \text{«занят»}\}$ и r – длиной очереди заявки на входе ОА (множество целых чисел). Состояние ОА характеризуется множеством $U = \{\text{«обслуживание»}, \text{«ожидание»}\}$

Правило, по которому заявки поступают из очереди на обслуживание, называется дисциплиной обслуживания. Приоритет – величина, выражающая преимущественное право на обслуживание. Если все заявки имеют равные приоритеты, то дисциплина обслуживания бесприоритетная. Часто используют бесприоритетную дисциплину обслуживания FIFO – «первым пришел – первым обслужен». Применяется также дисциплина LIFO, по которой заявки выбираются из конца очереди. Возможны дисциплины со случайным выбором заявок из очереди.

На входе ОА для каждого приоритета своя очередь. Заявки из одной очереди обслуживаются по FIFO. Заявки из более приоритетной очереди обслуживаются раньше заявок с низким приоритетом.

Различают приоритеты: абсолютный – поступающая на вход занятого ОА прерывает обслуживание заявки с более низким приоритетом; относительный – прерывание начатого обслуживания не происходит; динамический – может меняться в процессе обработки заявки в системе.

В общем случае обслуживание заявки ОА характеризуется промежутком времени, необходимым для обслуживания, и является случайной величиной с некоторым законом распределения. Чаще всего используются экспоненциальное и нормальное распределения. Математическое ожидание этого закона – среднее время обслуживания заявки.

Для получения математических моделей объектов в виде моделей СМО нет формализованных методов. Разработанная проектировщиком модель, как правило, описывается на языке моделирования, который имеет ту или иную специализацию и отражает определенные концепции моделирования.

Существуют два типа моделей СМО: аналитические модели СМО, имитационные модели СМО.

Аналитические модели СМО – совокупность явных зависимостей Y от X и Q . Вектор X составляют параметры ОА. Вектор Q – параметры входных потоков заявок. Допускают, что время обслуживания заявок и интервал времени между соседними заявками распределены по экспоненциальному закону

$$F(Y) = \int_0^Y \lambda \cdot e^{-\lambda \cdot x} dx = 1 - e^{-\lambda \cdot Y}.$$

Большинство практических задач не удается решать в должном объеме на основе аналитических моделей.

Имитационные модели СМО – воспроизводят обслуживание в системе множества заявок при заданных статистических сведениях о входных потоках и ресурсах СМО. Имитационная модель СМО состоит из моделей элементов, в качестве которых фигурируют: источники входного потока заявок – алгоритм, по которому вычисляются моменты поступления заявок; устройства – алгоритм выработки значений интервалов обслуживания; память – алгоритм определения объема памяти, требующегося для обслуживания заявки; узел – отображает маршрут движения заявки, связи между элементами модели. Следовательно, имитационная модель СМО представляет собой алгоритм, состоящий из упорядоченных обращений к моделям элементов (источников, устройств, памятей, узлов).

Таким образом, в зависимости от целей функционального проектирования при рассмотрении СУ на самом верхнем уровне абстракции (первом) возможны различные описания систем управления и их структурных схем в соответствии с выбранным математическим аппаратом.

Математические модели устройств СУ

С повышением степени детализации описания СУ переходят на следующий уровень абстрагирования (второй), являющийся более низким по отношению к предыдущему – уровень устройств (средств) СУ.

Как и на первом уровне ММ устройств, СУ может быть представлена в виде структурной схемы. Например, одним из часто встречающихся устройств в составе СУ является электронный усилитель, ММ которого приведена на [рис. 4.6](#).

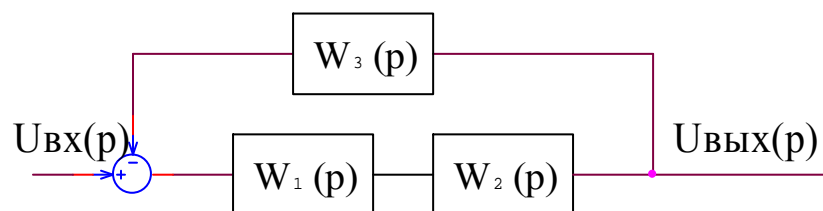


Рис. 4.6

Электронный усилитель работает в малосигнальном режиме и состоит из двух каскадов и цепи обратной связи, имеющих передаточные функции $W_1(p)$, $W_2(p)$ и $W_3(p)$ соответственно. Математическая модель может быть получена непосредственно по схеме [рис. 4.6](#):

$$U_{\text{ВЫХ}}(p) = K(U_{\text{ВХ}}(p) + W_3(p)U_{\text{ВЫХ}}(p)),$$

т. е. передаточная функция усилителя

$$W_y(p) = \frac{U_{\text{ВЫХ}}(p)}{U_{\text{ВХ}}(p)} = \frac{K}{1 - KW_3(p)}, \quad (4.7)$$

где $K = W_1(p)W_2(p)$.

Математическое описание многих устройств СУ, а также объектов управления, в общем случае есть система обыкновенных дифференциальных уравнений, записанная на общепринятом математическом языке, безотносительно к методу численного решения. Применительно к системам нелинейных ОДУ различают две формы представления:

общая – система ОДУ представлена в неявном виде

$$F\left(\frac{dV}{dt}, V, t\right) = 0, \quad (4.8)$$

где V – вектор фазовых переменных, F – вектор-функция;

нормальная – система уравнений представлена в явном виде, разрешенном относительно вектора производных

$$\frac{dV}{dt} = F(V, t). \quad (4.9)$$

Нормальная форма представления удобна для использования ряда методов анализа технических систем.

Параметрами ММ устройств СУ будем считать составляющие коэффициентов дифференциальных уравнений, которые несут определенный физический смысл (например, моменты инерции, коэффициенты усиления, постоянные времени).

Линеаризованная форма представления – представление уравнений модели в линейном виде. Линеаризацию выполняют с помощью разложения нелинейных элементов вектора F в ряд Тейлора с сохранением в разложении только линейных членов. После линеаризации систему (4.9) представляют в форме

$$\begin{aligned} \frac{dv_i}{dt} &= \sum_{j=1}^n a_{ij} v_j + \sum_{k=1}^m b_{ik} u_k, \quad i = 1, \dots, n, \\ y_l &= \sum_{i=1}^n c_{li} v_i, \quad l = 1, \dots, s. \end{aligned} \quad (4.10)$$

В этой модели имеется n взаимосвязанных дифференциальных уравнений 1-го порядка, в правую часть которых входят m различных внешних воздействий U_k , а также s алгебраических соотношений, связывающих с выходных (управляемых) процессов y_l с переменными состояниями v_i , число которых (n) совпадает с числом уравнений. Коэффициенты a_{ij} , b_{ik} , c_{li} – параметры модели.

Уравнения (4.10) удобно представить в матричной форме

$$\begin{aligned} \frac{dv}{dt} &= AV + BU, \\ Y &= CV, \end{aligned} \quad (4.11)$$

где $V = (V_1, V_2, \dots, V_n)$ – вектор переменных состояний, $U = (U_1, U_2, \dots, U_m)$ – вектор внешних воздействий, $Y = (Y_1, Y_2, \dots, Y_s)$ – вектор выходов, $A = \{a_{ij}, i = 1, 2, \dots, n; j = 1, 2, \dots, n\}$, $B = \{b_{ik}, i = 1, 2, \dots, n; k = 1, 2, \dots, m\}$, $C = \{c_{li}, l = 1, 2, \dots, s; i = 1, 2, \dots, n\}$ – матрицы параметров. Вся информация о свойствах проектируемой технической системы содержится в числовых таблицах – матрицах параметров. При анализе системы и синтезе законов управления можно будет опираться на стандартные процедуры преобразования этих таблиц.

Ряд форм представления модели СУ получается при преобразовании ее уравнений на основе формул и требований выбранного численного метода решения.

Алгебраизованная форма представления – результат представления дифференциальных уравнений в полученных после дискретизации точках в алгебраизованном виде с помощью формул численного интегрирования. ММ есть система алгебраических уравнений вида

$$Y_n V_n = B_n,$$

где значения переменных соответствуют значениям в момент времени t_n . Вид матрицы Якоби Y_n и вектора правых частей B_n определяется выбранным методом алгебраизации. Для получения ММ системы в алгебраизованной форме реализуют одну из неявных разностных формул численного интегрирования в общем виде:

$$\left(\frac{dV}{dt} \right)_n = \eta_n V_n + \mu_n, \quad (4.12)$$

где V_n – значение векторов в момент времени t_n , n – номер шага интегрирования, η_n – коэффициент зависящий от выбранного метода интегрирования и значения шага, μ_n – вектор зависящий от значения векторов V на предыдущих шагах.

Алгебраизация и линеаризация могут осуществляться по отношению ко всем или только избранным переменным, уравнениям или их частям, что увеличивает разнообразие возможных форм представления моделей.

Формы (4.8), (4.9) являются достаточно общими для отображения различных устройств СУ. ММ непрерывных процессов в СУ и их устройствах описываются уравнениями (4.8), (4.9) без каких-либо изменений. ММ дискретных устройств СУ получаются из уравнений (4.8), (4.9) как частный случай, когда функции в них определены только при $t_n = nT$, где T – такт дискретности по времени, а $n = 0, 1, 2, \dots$

В отличие от непрерывных моделей ММ дискретной модели устанавливают соотношения между решетчатыми функциями

$$Y(nT) = \Phi[X(nT)],$$

значения которых могут быть определены не в любой момент времени. Для решетчатых функций вводятся понятия квантования по времени (T) и квантования по уровню (Δ). ММ дискретных устройств представляют в форме разностных уравнений. Переход от исходных дифференциальных уравнений к эквивалентным разностным можно сделать по-разному. В частности, полагая в (4.9) $t = nT$, $\frac{dV(t)}{dt} = \frac{\Delta V(nT)}{T}$, получим обобщенные разностные уравнения в векторной форме [4, 13]:

$$V[(n+1)T] = V(nT) + TF(V(nT), nT)$$

В общем случае разностное уравнение n -го порядка записывается при $T = 1$ в виде

$$F(n, V(n), \Delta V(n), \dots, \Delta^k V(n)) = 0, \tag{4.13}$$

Где под конечной разностью $\Delta V(n)$ понимается выражение

$$\Delta V(n) = V(n+1) - V(n),$$

И далее

$$\begin{aligned} \Delta^2 V(n) &= V(n+1) - \Delta V(n), \\ &\dots\dots\dots \\ \Delta^k V(n) &= \Delta^{k-1} V(n+1) - \Delta^{k-1} V(n). \end{aligned}$$

Раскрывая эти выражения, получим

$$\Delta^2 V(n) = V(n+2) - 2V(n+1) + V(n), \tag{4.14}$$

$$\Delta^3 V(n) = V(n+3) - 3V(n+2) + 3V(n+1) - V(n),$$

.....

$$\Delta^k V(n) = \sum_{i=0}^k \frac{k!}{i!(k-i)!} V(n+i) (-1)^{k-i}.$$

После подстановки (4.14) в (4.13) приходим к разностному уравнению k -го порядка в форме

$$F(n, V(n), V(n+1), V(n+2), \dots, V(n+k)) = 0. \quad (4.15)$$

Если это уравнение удастся представить в виде

$$V(n+k) = F_I(n, V(n), V(n+1), \dots, V(n+k-1)) = 0, \quad (4.16)$$

то очевидно, задав начальные значения при $n = 0, V(0), V(1), \dots, V(k-1)$, получим V_n и вообще V_k при любом целом n .

Уравнение (4.16) представляет собой разностный аналог нормальной формы (4.9) и позволяет рекуррентно вычислить все интересующие значения $V(n)$.

Формы представления моделей устройств СУ определяются также используемыми языковыми средствами. Наряду с традиционным математическим языком используются алгоритмические языки, а также те или иные графические изображения, облегчающие восприятие модели устройств СУ и приводящие к представлению моделей в схемной форме.

К схемной форме представления математических моделей относится представление моделей в виде эквивалентных схем и в виде графов. Такое представление модели отличается высокой наглядностью, позволяет сосредоточить внимание на наиболее существенных связях и находить оптимальное решение задач проектирования ССУ.

Использование аппарата теории графов при разработке алгоритмов проектирования даёт удобную систему терминов и операций для описания и решения многих задач. Графы используются при решении задач синтеза, особенно в конструкторском проектировании, при решении задач анализа. Рассмотрим только те понятия и термины теории графов, которые необходимы для решения задач моделирования.

Граф – объект, состоящий из двух множеств (множество узлов и множество ребер), которые находятся между собой в некотором соотношении [7]. Если рёбра графа имеют определённое направление, то такой граф называют ориентированным (орграф). Рёбра орграфа называются дугами. Подграфом является часть графа, образованная некоторым подмножеством ребер графа и всеми инцидентными им вершинами. Если в подграф входят все

вершины графа, то подграф называют суграфом. Маршрутом называют последовательность смежных рёбер, т. е. рёбер инцидентных одной и той же вершине. Цепь – маршрут, в котором все рёбра различны. Замкнутая цепь называется циклом (контуром). Граф является связным, если можно указать маршрут, охватывающий все вершины. Дерево графа – связный подграф, не имеющий циклов. Фундаментальное дерево (остов) – связный суграф не имеющий циклов, т. е. фундаментальное дерево, охватывает все вершины графа и не образует ни одного цикла. Ветви дерева – рёбра графа, вошедшие в дерево. Хорды – рёбра графа, не вошедшие в дерево. Выбор фундаментального дерева графа неоднозначен. Для одного и того же графа их может быть несколько.

При моделировании на макроуровне особый интерес представляет дерево, в которое рёбра включаются согласно некоторому приоритету. Если изображая структуру объекта, за каждым ребром графа закреплять обозначения, заменяемого им элемента, то можно построить нормальное дерево графа. Нормальное дерево графа – фундаментальное дерево, в котором рёбра включаются в такой последовательности: источники разности потенциала типа E , рёбра типа C , рёбра типа R , рёбра типа L , источники переменной типа I . Сечение ветви дерева – множество ребер, пересекаемых линией сечения (при этом выполняются следующие условия: линия сечения является замкнутой и пересекает любое ребро не более одного раза, среди ветвей дерева пересекается единственная).

Граф несёт информацию о связях в объекте, удобную для восприятия человеком, но для алгоритмической обработки нужна информация числового характера. Представить граф в таком виде можно с помощью матрицы инцидентий A , которая кодирует ориентированный граф: каждому узлу графа (кроме одного, называемого базовым) соответствует одна строка, каждому ребру – один столбец. В столбце записывается $+1$ на пересечении со строкой узла, из которого ребро направлено, -1 на пересечении столбца со строкой узла, к которому ребро направлено, остальные элементы этого столбца равны 0 . Базовому узлу в матрице инцидентий никакая строка не соответствует. В качестве базового может быть выбран произвольный узел. Матрица инцидентий является сильно разреженной (содержащей нулевые элементы), причем разреженность возрастает с увеличением размера матрицы.

Граф отображает только такие структурные особенности моделируемого объекта, как способ связи элементов друг с другом. Наиболее полную информацию об объекте содержит эквивалентная схема, отображающая также физическую сущность отдельных элементов. Эквивалентной схемой объекта является графическое представление структуры объекта – взаимосвязей между его элементами. Эквивалентная схема равноценна графу, но в отличие от последнего каждая ветвь имеет условное изображение, отражающее ее физическую суть, т. е. кроме топологической информации присутствует информация о типах ветвей.

Направления потоков в ветвях выбираются произвольно (если реальное направление при моделировании окажется противоположным, то это приведет лишь к отрицательным численным значениям потока).

Представить структуру в виде эквивалентной схемы можно для следующих физических однородных подсистем: электрической, механической поступательной, механической вращательной, гидравлической закрытой, гидравлической открытой, магнитной и тепловой. Как можно видеть, из этих однородных подсистем состоит практическое большинство технических объектов.

В физически однородных устройствах СУ (подсистемах) различают три типа простых элементов: емкостного типа C , индуктивного типа L и резистивного типа R .

В каждой из подсистем используются специфические системы обозначений элементов на эквивалентных схемах. В дальнейшем будем использоваться единую систему обозначений для элементов всех подсистем, обычно применяемую при изображении электрических эквивалентных схем. При этом элементы представляют собой двухполюсники (рис. 4.7) C, R, L, E, I .



R



C



L



E



I

Ветви типа E соответствует компонентное уравнение $E = f(Z)$, где в качестве Z могут фигурировать время, фазовые переменные или константа, E – разность переменных типа потенциалов для узлов этой ветви. Ветви типа I соответствует компонентное уравнение $I = f(Z)$, где в качестве Z могут фигурировать время, фазовые переменные или константа, I – переменная типа потока. Компонентные уравнения ветвей типа R, C, L соответственно:

Рис. 4.7

$$U = IR; I = C \frac{dU_c}{dt}; U = L \frac{dI_L}{dt}.$$

Таким образом, если при моделировании СУ на уровне устройств нужно отобразить только топологические связи элементов устройств СУ, то достаточно использовать графы, если нужно отобразить топологию и функционирование устройств СУ, то целесообразно использовать эквивалентные схемы.

Общий алгоритм составления эквивалентных схем имеет следующие этапы:

- выделение в СУ однородных физических подсистем;
- составление эквивалентных схем однородных подсистем без учета их взаимовлияния;
- установление связей между подсистемами.

Проектировщик в эквивалентной схеме отражает те элементы и свойства реального объекта, которые, по его мнению, оказывают существенное влияние на функционирование объекта. Чем можно пренебречь, ему подска-

зывают опыт и интуиция, так как процедура составления эквивалентной схемы не полностью формализована.

Таким образом, рассмотренные ММ могут служить для решения задач в процессе проектирования устройств, соответствующих различным классам СУ: нелинейным и линейным, нестационарным и стационарным, непрерывным и дискретным.

Математические модели элементов устройств СУ

Элементы СУ при первоначальном их получении описываются на уровне представления физических явлений в сплошных средах и называются ММ объектов с распределенными параметрами. В качестве математического описания используются дифференциальные уравнения или системы дифференциальных уравнений в частных производных (PDE) с краевыми условиями первого и второго рода.

Точное решение краевых задач удается получить лишь для немногих частных случаев. Поэтому общий способ их решения, в том числе и в САПР, заключается в использовании различных приближенных моделей. В настоящее время наиболее широкое распространение получили модели на основе интегральных уравнений и модели на основе различных методов дискретизации пространства.

Обобщенная форма представления ММ элементов СУ как объекта с распределенными параметрами есть дифференциальное уравнение в частных производных (PDE):

$$Lv - f = 0, X \in G, \quad (4.17)$$

с граничными условиями:

$$L_{\Gamma}v = f_{\Gamma}, x \in \Gamma,$$

где L и L_{Γ} – дифференциальные операторы;

$v = v(x)$ – анализируемая физическая величина;

X – точка евклидова пространства EN ;

Γ – граница области G ;

f, f_{Γ} – функции, описывающие распределение внутренних и граничных источников исследуемого поля.

Пусть в N -мерной области G евклидова пространства EN необходимо рассчитать распределение некоторой физической величины. Рассмотрим пространство M функций, определенных на области G , такое, что если $u_1(x) \in M$, $u_2(x) \in M$, то

$$u_3(x) = a_1 u_1(x) + a_2 u_2(x) \in M$$

(любая линейная комбинация функций, принадлежащих пространству M , также принадлежит пространству M). Введём скалярное произведение функций:

$$(u, v) = \int_G u(x)v(x)dG,$$

тогда нормой функции $u \in M$ будем называть неотрицательное число $\|u\| = (u, u)^{1/2}$, расстоянием между функциями u и v будем называть величину, равную норме $\|u - v\|$. Такое пространство M называется гильбертовым пространством [1].

$v = v(x)$ – искомая функция (фазовая переменная), которая может быть напряжением механического элемента, потоком газа или жидкости в пневматическом или гидравлическом элементах СУ, потоком носителей заряда в электронных элементах устройств СУ.

В простейшем случае для однокоординатной системы уравнения (4.17) можно записать параболическое уравнение в виде

$$\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2} + f(x, t), \Gamma : v = \varphi_2(x, t), \quad (4.18)$$

которое может, например, применяться при расчете распространения теплоты в одномерных элементах механических устройств СУ. При этом требуется найти $V(x, t)$ – распространение температуры в зависимости от расположения источника тепла $f(x, t)$ по отношению к измеряемой точке оси x в заданные моменты времени t .

Гиперболическое уравнение (волновое уравнение)

$$\frac{\partial^2 v}{\partial x^2} - \frac{\partial^2 v}{\partial y^2} = f(x, y), \Gamma : v = \varphi_3(x, y)$$

применяется при расчетах пневматических и гидравлических элементов СУ. В частности, распространение потока, обтекающего элементы корпуса ЛА, расчет давления газа в чувствительном элементе гиросtabilизатора, расчет давления пара в турбогенераторе. Результаты расчета в виде функции $v(x, y)$ используются в уравнениях (4.8), (4.9), (4.16) динамики соответствующих устройств СУ как возмущающие воздействия на систему.

Лекция 5. Методы формирования моделей ССУ

Формирование математических моделей элементов устройств СУ. Формирование математических моделей устройств СУ. Формирование математических моделей систем управления.

При получении математического описания СУ исходят из уровней описания СУ, представленных на [рис. 4.3](#) – «система – устройство – элемент». В зависимости от степени детализации описания будут различаться и методы формирования моделей на разных уровнях рассмотрения и проектирования СУ[[10](#)]. Для автоматизированного проектирования СУ основным вопросом является формализация методов формирования математических моделей СУ, устройств и элементов СУ. Именно от степени автоматизации процедур получения математического описания зависит эффективность применения САПР СУ, а значит и качество проектирования в целом. Необходимо помнить и о том, что именно метод получения математической модели, реализованный в САПР, во многом будет определять характеристики полученной модели элемента, устройства и СУ в целом. Методы получения ММ в значительной степени оказывают влияние на выбор методов анализа ССУ. И самое главное, при выборе метода формирования ММ при проектировании САПР как инструментального средства необходимо помнить об одном из основных назначений САПР – создание удобной среды проектирования для инженера – пользователя [[12](#)].

Наибольшие трудности возникают при построении ММ объектов управления, так как учет физических процессов и явлений в этих объектах требует длительных и весьма трудоемких математических операций. Большое количество однообразных и утомительных операций при построении ММ, в том числе дифференцирование, умножение полинома на полином, раскрытие скобок, приведение подобных членов, неизбежно связаны с многочисленными ошибками. Автоматизация этих операций обязательна и на современном этапе выполняется на базе символьных процессоров математических систем Maple, MathCAD, MATLAB.

Формирование математических моделей элементов устройств СУ

В качестве математического описания моделей элементов устройств СУ используются дифференциальные уравнения или системы дифференциальных уравнений в частных производных (PDE) с краевыми условиями первого и второго рода ([4.17](#)). Общий способ их формирования и решения, в том числе и в САПР, заключается в использовании различных приближенных подходов. В настоящее время наиболее широкое распространение получили модели на основе интегральных уравнений и модели на основе различных методов дискретизации пространства [[11](#); [12](#)].

Основная идея построения модели на основе интегральных уравнений заключается в переходе от исходного дифференциального уравнения в частных производных к эквивалентному интегральному уравнению, подлежащему дальнейшим преобразованиям.

Сущность метода состоит в аппроксимации искомой непрерывной функции совокупностью приближенных значений, рассчитанных в некоторых точках области – узлах. Совокупность узлов, соединенных определенным образом, образует сетку. Сетка, в свою очередь, является дискретной моделью области определения искомой функции.

С помощью дискретизации непрерывная математическая модель преобразуется в дискретную модель, состоящую из конечного числа степеней свободы. Методы пространственной дискретизации:

- методы конечных элементов (МКЭ);
- методы граничных элементов (МГЭ);
- методы конечных разностей (МКР);
- методы конечных объемов (МКО);
- спектральный метод;
- метод свободных стенок.

В настоящее время в линейной механике твердого тела метод конечных элементов наиболее распространен, в то время как применение метода граничных элементов для решения данных задач находится на втором месте. Для нелинейных задач метод конечных элементов является наиболее эффективным и доминирующим.

Классический метод конечных разностей почти полностью потерял свое значение при решении практических задач механики твердого тела. Это утверждение, однако, неверно для механики жидкости и газов, где разностные методы до сих пор широко распространены. Метод конечных объемов, основанный на законах сохранения, применяется для решения сильно нелинейных задач механики жидкости и газов. Спектральные методы используются в различных областях механики и основаны на пространственно-временном преобразовании в область, где задача может быть легко решена. Метод свободных сеток – один из новых методов вычислительной математики и основан на конечно-разностном подходе с использованием независимых сеток, полученных в результате применения конечно-элементных технологий.

Оба метода относятся к классу сеточных методов приближенного решения краевых задач. С точки зрения теоретических оценок точности, методы обладают примерно равными возможностями. В зависимости от формы области, краевых условий, коэффициентов исходного уравнения оба метода имеют погрешности аппроксимации от первого до четвертого порядка относительно шага. В силу этого они успешно используются для разработки программных комплексов автоматизированного проектирования технических объектов.

Методы конечных элементов и конечных разностей имеют ряд существенных отличий. Прежде всего, методы различны в том, что в МКР аппроксимируются производные искомых функций, а МКЭ – само решение, т. е. зависимость искомых функций от пространственных координат и времени. Методы сильно отличаются и в способе построения сеток. В МКР строятся, как правило, регулярные сетки, особенности геометрии области учитываются только в околограничных узлах. В связи с этим МКР чаще применяется для анализа задач с прямолинейными границами областей определения функций. К числу традиционных задач, решаемых на основе МКР, относятся исследования течений жидкостей и газов в трубах, каналах с учетом теплообменных процессов и ряд других. В МКЭ разбиение на элементы производится с учетом геометрических особенностей области, процесс разбиения начинается от границы с целью наилучшей аппроксимации её геометрии. Затем разбивают на элементы внутренние области, причем алгоритм разбиения строится так, чтобы элементы удовлетворяли некоторым ограничениям, например стороны треугольников не слишком отличались по длине, и т. д. Поэтому МКЭ наиболее часто используется для решения задач с произвольной областью определения функций: расчет на прочность деталей и узлов строительных конструкций, авиационных и космических аппаратов, тепловой расчет двигателей и т. д.

Общей проблемой методов является высокая размерность результирующей системы алгебраических уравнений (несколько десятков тысяч в реальных задачах). Поэтому реализация МКР и МКЭ в составе САПР требует разработки специальных способов хранения матрицы коэффициентов системы и методов решения последней.

Основная идея МКР заключается в замене частных производных их разностными аналогами. На [рис. 5.1](#) представлена графическая интерпретация некоторых конечно-разностных аппроксимаций для производных.

Существует три способа замены производной первого порядка [2; 3].

Правая схема:

$$\frac{\partial V}{\partial x} = \frac{V(x + \Delta x) - V(x)}{\Delta x}.$$

Левая схема:

$$\frac{\partial V}{\partial x} = \frac{V(x) - V(x - \Delta x)}{\Delta x}.$$

Центральная схема:

$$\frac{\partial V}{\partial x} = \frac{V(x + \Delta x) - V(x - \Delta x)}{2\Delta x}.$$

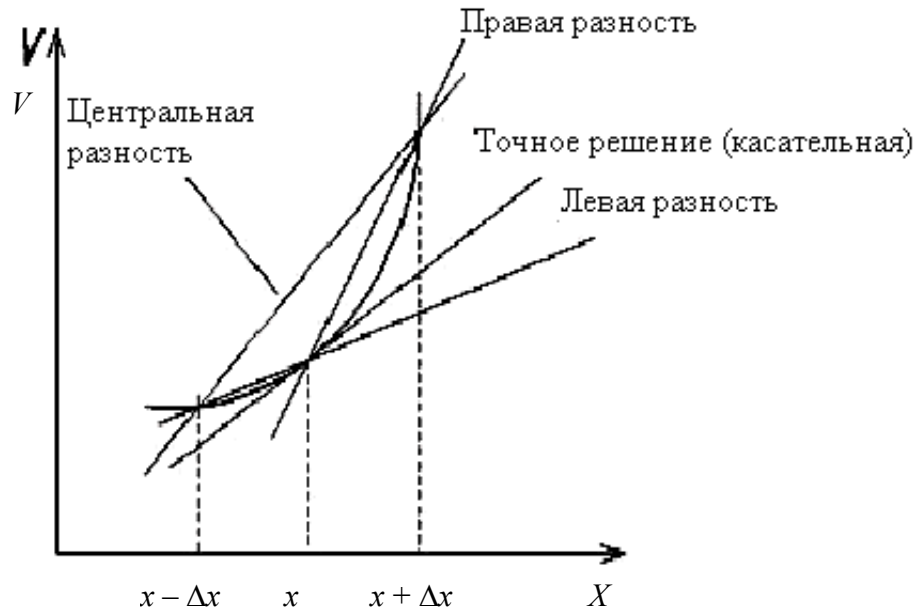


Рис. 5.1

Для получения разностных аналогов производных более высоких порядков рассмотрим разложение функции в ряд Тейлора в окрестности точки x :

$$V(x + \Delta x) = V(x) + \frac{\partial V}{\partial x} \Delta x - \frac{1}{2!} \frac{\partial^2 V}{\partial x^2} \Delta x^2 + \frac{1}{3!} \frac{\partial^3 V}{\partial x^3} \Delta x^3 - \frac{1}{4!} \frac{\partial^4 V}{\partial x^4} \Delta x^4 \dots \quad (5.1)$$

$$V(x - \Delta x) = V(x) - \frac{\partial V}{\partial x} \Delta x + \frac{1}{2!} \frac{\partial^2 V}{\partial x^2} \Delta x^2 - \frac{1}{3!} \frac{\partial^3 V}{\partial x^3} \Delta x^3 + \frac{1}{4!} \frac{\partial^4 V}{\partial x^4} \Delta x^4 \dots \quad (5.2)$$

Для получения центральной разности второго порядка вычтем выражение (5.2) из выражения (5.1), получим

$$\frac{\partial^2 V}{\partial x^2} = \frac{\frac{V(x + \Delta x) - V(x)}{\Delta x} - \frac{V(x) - V(x - \Delta x)}{\Delta x}}{\Delta x} = \frac{V(x + \Delta x) - 2V(x) + V(x - \Delta x)}{\Delta x^2}.$$

В индексированном пространстве узлов сетки формула будет иметь вид

$$\frac{\partial^2 V}{\partial x^2} = \frac{V_{i-1} - 2V_i + V_{i+1}}{\Delta x^2}.$$

Разностный аналог производной третьего порядка получаем аналогично:

$$\frac{\partial^3 V}{\partial x^3} = \frac{\frac{V_{i+2} - 2V_{i+1} + V_i}{\Delta x^2} - \frac{V_{i+1} - 2V_i}{\Delta x^2}}{\Delta x} = \frac{V_{i+2} - 3V_{i+1} + 3V_i - 3V_{i-1}}{\Delta x^3}.$$

Теперь рассмотрим алгоритм метода конечных разностей.

Метод конечных разностей является старейшим методом решения краевых задач [1; 3].

Алгоритм МКР состоит из этапов, традиционных для метода сеток.

1. Построение сетки в заданной области. В МКР используется сетка, задаваемая конечным множеством узлов. В узлах сетки определяются приближенные значения φ_h искомой функции φ . Совокупность узловых значений φ_h называют сеточной функцией.

2. Замена дифференциального оператора $L_h = \frac{\partial \varphi}{\partial u}$ в исходном дифференциальном уравнении разностным аналогом L_h , построенным по одной из схем, рассмотренных ниже. При этом непрерывная функция φ аппроксимируется сеточной функцией φ_h .

3. Если есть граничные условия второго и третьего рода, то для граничного узла с этим условием записывается соответствующая аппроксимация. В результате должна получиться замкнутая система НАУ.

4. Решение полученной нелинейной системы алгебраических уравнений.

Рассмотрим задачу распространения теплового потока в одномерной области. На [рис. 5.2](#) представлена одномерная сетка, состоящая из четырех точек.



Рис. 5.2

Решим уравнение:

$$\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) = 0, \tag{5.3}$$

где $k = k(T) = T^2$, T – функция температуры.

Применив центральную разностную схему, получим систему НАУ (5.4), решив которую с учетом граничных условий получим значение температуры в узлах сетки.

$$\left\{ \begin{array}{l} \frac{\left(\frac{T_2 + T_1}{2}\right)^2 \frac{T_2 - T_1}{\Delta x} - \left(\frac{T_1 - T_0}{2}\right)^2 \frac{T_1 - T_0}{\Delta x}}{\Delta x} = 0, \\ \frac{\left(\frac{T_3 + T_2}{2}\right)^2 \frac{T_3 - T_2}{\Delta x} - \left(\frac{T_2 - T_1}{2}\right)^2 \frac{T_2 - T_1}{\Delta x}}{\Delta x} = 0. \end{array} \right. \quad (5.4)$$

Основа физической концепции МКЭ – это разбиение математической модели конструкции на непересекающиеся компоненты (подобласти) простой геометрии, называемые конечными элементами или просто элементами (для краткости) [1; 2; 3]. Множество элементов, на которые разбита конструкция, называется конечно-элементной сеткой. Механическое поведение каждого элемента выражается с помощью конечного числа степеней свободы или значений искомым функций во множестве узловых точек. Поведение математической модели, таким образом, аппроксимируется поведением дискретной модели, полученной путем сборки или ансамблирования всех элементов. Заметим, что концепция разбиения-сборки естественно возникает при исследовании многих искусственных или живых систем. Например, легко представить мост, здание, двигатель или скелет как сложную систему, составленную из простых компонентов. Заметим также, что в отличие от метода конечных разностей конечные элементы не накладываются друг на друга в пространстве.

Рассмотрим основные типы конечных элементов для одно-, двух- и трехмерных задач механики и их свойства, называемые атрибутами элементов (рис. 5.3).

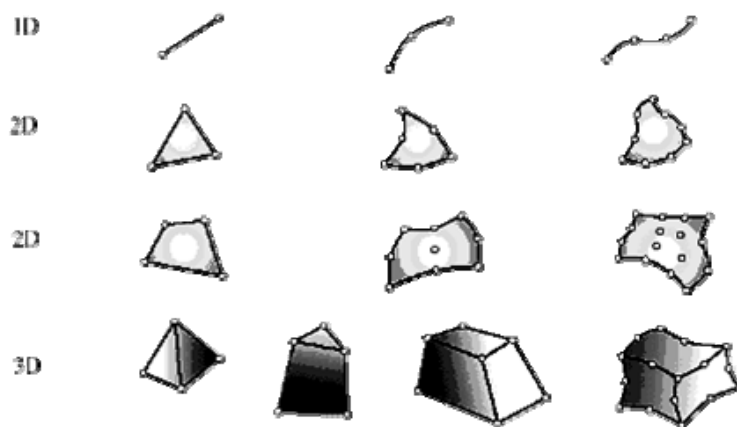


Рис. 5.3

Собственная размерность. Конечные элементы могут описываться одной, двумя или тремя пространственными координатами в зависимости от размерности задачи, для решения которой они предназначены. Соответствующее число внутренних или локальных координат называется собственной размерностью элемента. В динамическом анализе время рассматривается как дополнительная размерность. Отметим, что в расчетах используются также специальные элементы с нулевой размерностью, такие как точечные массы или сосредоточенные упругие элементы (пружины).

Узловые точки. Каждый элемент описывается множеством характерных точек, называемых узловыми точками или узлами (для краткости). Узлы предназначены для описания геометрии элемента и для задания физических степеней свободы (числа неизвестных функций). Узлы обычно находятся в угловых или крайних точках элемента, но могут быть также расположены между угловыми узлами и внутри элемента. Данное различие связано с порядком аппроксимации, который обеспечивает данный конечный элемент. Элементы, имеющие только угловые узлы, называются линейными и обеспечивают линейную интерполяцию геометрии и функций. Элементы, имеющие дополнительные узлы на своих границах между угловыми точками, могут обеспечивать квадратичную или даже кубическую интерполяцию

В первом случае такие элементы называются квадратичными. Отметим также, что существуют элементы, имеющие внутренние узлы. Теоретически такие элементы обеспечивают более точное описание геометрии тела и искомым функций, однако широкого распространения данный тип элементов не получил. При наличии современных автоматических генераторов конечно-элементных сеток часто бывает проще и удобнее разбить конструкцию на большое число линейных элементов простой формы, чем использовать элементы высокого порядка, требующие для построения сетки значительной работы вручную. Элементы, не имеющие внутренних узлов, относятся к так называемому серендипову семейству.

Геометрия элемента. Геометрия элемента определяется расположением узловых точек. Большинство элементов, используемых в расчетах, имеют достаточно простую геометрическую форму. Например, в одномерном случае элементы обычно представляют собой прямолинейные отрезки или сегменты кривых линий; в двумерном случае элементы имеют трехстороннюю или четырехстороннюю форму; в трехмерных задачах наиболее распространены такие геометрические фигуры, как тетраэдры, призмы и гексаэдры (см. [рис. 5.1](#)).

Степени свободы. Степени свободы определяют физическое состояние элемента, т. е. физическое поле, которое описывает данный элемент. Благодаря общим степеням свободы в соседних элементах осуществляется сборка модели и формирование глобальной системы конечно-элементных уравнений. В качестве степеней свободы могут фигурировать как узловые значения неизвестной функции, так и ее производные по пространственным координатам в узлах. В первом случае элементы относятся к типу лагранжевых элементов; во втором – к типу эрмитовых элементов. Например, в простейшей задаче о растяжении стержня неизвестной функцией является продольное пе-

ремещение стержня. Соответственно в качестве степеней свободы выступают узловые значения данной функции и, следовательно, конечный элемент относится к лагранжевому типу. Наоборот, в задаче об изгибе стержня неизвестной функцией является поперечное перемещение центральной оси стержня, а в качестве степеней свободы используются как узловые значения самой функции, так и ее производной по продольной координате. Физический смысл этой производной – угол поворота поперечного сечения стержня. Таким образом, конечный элемент, применяемый в расчетах стержня на изгиб, относится к типу эрмитовых элементов. Заметим также, что данные обозначения происходят от названия полиномов Лагранжа и Эрмита, широко используемых в прикладной математике для интерполяции функций по узловым значениям.

Узловые силы. Система узловых сил полностью соответствует степеням свободы элемента и выражается с помощью глобального вектора узловых сил.

Определяющие соотношения. Для конечных элементов, используемых в механических расчетах, определяющее соотношение задает поведение материала, из которого изготовлена конструкция. Например, в качестве такого соотношения во многих случаях используется обобщенный закон Гука, связывающий тензор деформаций и тензор напряжений в точке. Для линейного упругого стержневого элемента достаточно задать один модуль Юнга E и один коэффициент температурного расширения.

Свойства сечения. К свойствам сечения относятся площади и моменты инерции одномерных и двумерных конечных элементов, таких как балки, стержни, пластины. В эту группу также входит толщина пластин и оболочек. При построении конечного элемента свойства сечений считаются заданными и входят в результирующую матрицу жесткости элемента.

Ансамблирование или сборка представляет собой объединение отдельных элементов в конечно-элементную сетку. С математической точки зрения ансамблирование состоит в объединении матриц жесткости отдельных элементов в одну глобальную матрицу жесткости всей конструкции. При этом существенно используются две системы нумерации узлов элементов: локальная и глобальная. Локальная нумерация представляет собой фиксированную нумерацию узлов для каждого типа конечных элементов в соответствии с введенной локальной системой координат на элементе. Глобальная нумерация узлов всей конструкции может быть совершенно произвольной, как и глобальная нумерация конечных элементов. Однако между локальными номерами и глобальными номерами узлов существует взаимно однозначное соответствие, на основе которого и формируется глобальная система конечно-элементных уравнений.

С позиции метода конечных элементов существенные граничные условия – это такие условия, которые непосредственно влияют на степени свободы модели и накладываются на компоненты глобального вектора неизвестных U . Наоборот, естественные граничные условия – это условия, которые опосредованно влияют на степени свободы через глобальную систему конечно-элементных уравнений и накладываются на правую часть системы – вектор F .

Для решения *PDE* задачи всю расчётную область представляют в виде совокупности неперекрывающихся геометрических фигур достаточно простой формы. Размеры таких фигур, как правило, бывают малы по сравнению с размерами расчётной области. Эти элементарные фигуры называют конечными элементами. Трёхмерные расчётные области обычно разбивают на многогранники, а двумерные – на многоугольники. Простейшие многогранники (прямолинейные четырёхузловые тетраэдры) и простейшие многоугольники (прямолинейные трёхузловые треугольники) называют симплекс-элементами. Вся совокупность конечных элементов в расчётной области называется конечноэлементной сеткой. Вершины этих многогранников или многоугольников называют узлами конечно-элементной сетки.

Пусть в результате расчёта известно узловое распределение некоторой физической величины. Для простоты будем полагать, что эта величина скалярная. Обозначим её латинской буквой u . Узловое распределение этой величины может быть описано столбцовой матрицей, которую обозначим $[u(y)]$. В этой матрице каждой строке соответствует узел конечно-элементной сетки. Распространение узлового распределения на все возможные точки расчётной области называют конечно-элементной аппроксимацией. В общем случае конечно-элементное аппроксимирующее выражение имеет вид [1; 3; 4]:

$$u(Q) = [N](Q) \cdot [u^{(y)}],$$

где Q – точка наблюдения, имеющая свои координаты; $[N](Q)$ – матрица-строка функций формы.

В пределах отдельно взятого конечного элемента

$$u(Q) = [N^{(e)}](Q) \cdot [u^{(e)}],$$

где $[N^{(e)}](Q)$ – матрица-строка функций формы конечного элемента; $[u^{(e)}]$ – узловое распределение физической величины в пределах конечного элемента.

Функции формы – это функции «интерполяционной природы» [1; 3; 4], обладающие следующими свойствами:

$$N_i(Q_j) = 1;$$

$$N_i(Q_j) = 0;$$

$$N_i(Q_k) = 0;$$

$$N_j(Q_k) = 0;$$

где $N_i(Q)$ – в общем случае любое значение, но для симплекс-элементов $N_i(Q) \in [0; 1]$.

Здесь обозначено: i, j – номера узлов некоторого конечного элемента; Q_i, Q_j – узлы конечного элемента; Q_k – точка, не принадлежащая конечному элементу; Q – точка, принадлежащая конечному элементу; N_i – скалярное поле, называемое функцией формы i -го узла; N_j – скалярное поле, называемое функцией формы j -го узла.

Для симплекс-элементов характерны линейные функции формы:

$$[N^{(e)}](Q) = [a] + [a_x] \cdot x + [a_y] \cdot y,$$

где x, y – координаты точки наблюдения Q ; $[a], [a_x], [a_y]$ – матрицы-строки коэффициентов функций формы конечного элемента, которые в соответствии со свойствами 1 – 4 вычисляются по формуле

$$\begin{bmatrix} [a] \\ [a_x] \\ [a_y] \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1},$$

где 1, 2, 3 – локальные номера узлов конечного элемента (далее для краткости – просто «элемент»).

Функции формы позволяют легко определять в пределах каждого элемента пространственные дифференциальные операторы первого порядка от скалярного или векторного поля по известному узловому распределению:

$$\text{grad}u = \text{grad}[N^{(e)}] \cdot [u^{(e)}],$$

$$\text{div}A = \text{grad}[N^{(e)}] \cdot [A^{(e)}],$$

$$\text{rot}A = \left| \text{grad}[N^{(e)}] \cdot [A^{(e)}] \right|,$$

где $[A^{(e)}]$ – узловое распределение векторного поля A в пределах элемента;

$$\text{grad}[N^{(e)}] = [a_x] \cdot l_x + [a_y] \cdot l_y,$$

где l_x, l_y – единичные базисные векторы (орты) декартовой системы координат.

Таким образом, конечно-элементная технология решения задач математической физики сводится к вычислению элементных матриц, соответствующих заданному PDE , сборке из них глобального матричного уравнения, решению этого уравнения и анализу узлового распределения искомой величины.

Формирование математических моделей устройств СУ

При переходе с третьего уровня описания СУ на второй процессы в системе рассматриваются не в сплошной, а в дискретной среде, увеличивается степень абстрагирования и математическое описание устройств СУ есть математические модели объектов с сосредоточенными параметрами. ММ в большинстве случаев является система обыкновенных дифференциальных уравнений (ОДУ) (4.8), (4.9). Поведение устройств можно охарактеризовать с помощью фазовых переменных, которые образуют вектор неизвестных V в системе уравнений математической модели. Фазовыми переменными в устройствах СУ электрической природы являются токи и напряжения, в устройствах СУ механической природы – силы и скорости. Непрерывной, независимой переменной в системе уравнений математической модели объекта с сосредоточенными параметрами остается время [6; 7; 8].

Для получения ММ элементов устройств СУ чаще используют теоретический подход. Этапами моделирования являются: выделение элементов разбиением общей структуры системы на отдельные участки; переход к усредненным значениям параметров и фазовых переменных. Исходными уравнениями для получения ММ элементов являются уравнения предыдущего уровня.

При формировании математического описания на втором уровне принято моделируемую СУ представлять в виде совокупности устройств – физически однородных подсистем. Каждая подсистема описывает процессы определенной физической природы: механические, электрические, тепловые и т. д. Для большинства физически однородных подсистем с сосредоточенными параметрами можно выделить переменные, используемые в математической модели, которые непосредственно характеризуют запасы энергии в подсистеме. Такие переменные называются переменными состояния.

В состав математических моделей объектов с сосредоточенными параметрами входят два типа уравнений: компонентные и топологические. Уравнения, описывающие законы функционирования элементов, называют компонентными. Уравнения, отражающие способ связи элементов между собой в составе объекта, называют топологическими. Фазовые переменные, фигурирующие в этих уравнениях, также можно разделить на два типа: переменные типа потенциала и переменные типа потока. Переменные типа потенциала U (или φ) соответствуют местам соединения (узлам, полюсам) элементов, переменные типа потока I соответствуют двухполюсным элементам, которые в схемах представлены в виде ветвей.

Законы функционирования элемента задаются компонентными уравнениями, связывающими разнородные фазовые переменные, относящиеся к одному элементу, т. е. компонентные уравнения связывают переменные типа потока с переменными типа потенциала. Так, уравнения закона Ома связы-

вают ток и напряжение резистора. Компонентные уравнения составляют основу математической модели элемента. Компонентные уравнения получаются на основе знаний о конкретной предметной области, либо теоретически, либо физическим макетированием, либо математическим моделированием на третьем уровне для каждого элемента моделируемого объекта. Это может быть длительная и трудоемкая процедура, но она выполняется однократно с одновременным накоплением библиотеки элементов.

Топологические уравнения связывают между собой однородные фазовые переменные и представляют собой уравнения равновесия и непрерывности. Примером могут служить уравнения законов Кирхгофа, записываемые относительно либо токов, либо напряжений ветвей. Для формирования топологических уравнений разработаны формальные методы. Топологические уравнения формируются для каждого моделируемого устройства СУ, поскольку их структуры уникальны.

Следует отметить, что топологические уравнения строго справедливы только в установившихся режимах, но их можно использовать и в случаях, когда временем распространения возбуждения по линиям связи объекта можно пренебречь. Время распространения возбуждения зависит от физической природы объекта. Возбуждение – это изменение фазовых переменных. Критическая длина – приближенный предельный размер среды, при повышении которого необходимо учитывать время распространения возбуждения. Критическая длина зависит от временного диапазона моделирования объекта. Например, в объектах электрической природы для переходных процессов в наносекунды критическая длина составляет 30 см, в объектах механической природы 30 см соответствует длительности переходных процессов в несколько миллисекунд.

Форма компонентных и топологических уравнений одинакова для большинства объектов различной физической природы. Это обуславливает наличие аналогий между системами разной физической природы. Аналогии физических однородных подсистем – одинаковый вид компонентных и топологических уравнений в различных физических подсистемах (с точностью до фазовых переменных и коэффициентов).

Установление аналогий между физически однородными подсистемами имеет важное значение при разработке и эффективном использовании универсального математического и программного обеспечения САПР для анализа сложных технических систем.

Топологические уравнения для различных подсистем приведены в [табл. 5.1](#). Можно видеть, что и топологические уравнения имеют одинаковый вид с точностью до фазовых переменных и коэффициентов.

Как видно из [табл. 5.1](#), уравнения одинаковы по отношению к фазовым переменным.

Таблица 5.1

Подсистема	Уравнение равновесия	Уравнение непрерывности
Механическая поступательная	$\sum_{i \in p} F_i = 0$ принцип Даламбера	$\sum_{j \in p} V_j = 0$ принцип сложения скоростей
Механическая вращательная	$\sum_{i \in p} M_i = 0$ принцип Даламбера для вращательных подсистем	$\sum_{j \in p} \omega_j = 0$ принцип сложения скоростей
Электрическая	$\sum_{i \in p} I_i = 0$ первый закон Кирхгофа	$\sum_{j \in q} U_j = 0$ второй закон Кирхгофа
Гидравлическая (пневматическая) закрытая	$\sum_{i \in p} G_i = 0$ сумма расходов в узле равна нулю	$\sum_{j \in q} P_j = 0$ сумма разностей давлений при обходе по замкнутому контуру равна нулю
Гидравлическая открытая	$\sum_{i \in p} G_i = 0$ сумма расходов в узле равна нулю	$\sum_{j \in q} h_j = 0$ сумма разностей напоров при обходе по замкнутому контуру равна нулю
Тепловая	$\sum_{i \in p} \Phi_i = 0$ сумма тепловых потоков в узле равна нулю	$\sum_{j \in q} T_j = 0$ сумма разностей температур при обходе по замкнутому контуру равна нулю
Магнитная	$\sum_{j \in p} \Phi_{M_j} = 0$ сумма магнитных потоков в узле равна нулю	$\sum_{j \in q} U_{M_j} = 0$ сумма падений магнитного напряжения при обходе по замкнутому контуру равна нулю

Наличие аналогий между физическими подсистемами имеет большое методологическое значение, поскольку моделирование различных устройств в рамках всех СУ можно вести с единых позиций. При этом при составлении эквивалентных схем сложных технических систем предпочтительнее привлекать аналогии между электрической системой и другими физическими системами. Эти аналогии позволяют при получении моделей объектов применять достаточно универсальные приемы построения моделей электрических систем, формализованные с использованием законов Кирхгофа и ориентированных графов. Сочетанием простейших элементов типа R, L, C , а также источников фазовых переменных, может быть получена математическая модель объекта любой сложности.

Реальная техническая система, как правило, представляет совокупность разнородных физических подсистем. При получении ММ таких СУ (после составления эквивалентных схем однородных подсистем) следует установить связи между ними, т. е. определить их воздействие друг на друга. Физические элементы, в которых происходит преобразование энергии или иное воздействие переменной одной подсистемы на переменные другой подсистемы, должны иметь описание в более чем одной подсистеме. Компонентные уравнения таких элементов должны содержать соотношения между фазовыми переменными двух разнородных подсистем. Например, элемент резистивного типа R в электрической подсистеме является ветвью с зависимым от температуры параметром «сопротивление», а для тепловой подсистемы – источник теплового потока.

Различают три вида связей подсистем:

- трансформаторный тип связи;
- гираторный тип связи;
- связь через зависимые параметры элементов.

Трансформаторная и гираторная связи выражают соотношения между фазовыми переменными двух подсистем. Этим типам связей соответствуют преобразовательные элементы, представляемые парами источников тока или напряжения. Третий вид связи выражает влияние фазовых переменных одной подсистемы на параметры элементов другой и задается в виде зависимостей C , L или R от фазовых переменных.

Вариант эквивалентной схемы трансформаторного типа связи приведен на [рис. 5.4, а](#). Такая связь характерна при электромеханическом взаимодействии. Варианты эквивалентных схем гираторного типа связи приведены на [рис. 5.4, б](#) и [рис. 5.4, в](#). Такой вид связи характерен при взаимодействии механической и гидравлической или пневматической подсистем. Запись вида $A(B)$ на эквивалентных схемах всех типов связи означает, что фазовая переменная A является функцией фазовой переменной B .

Связь через зависимые параметры элементов характерна при взаимодействии с тепловой подсистемой. В зависимости от значений температуры меняются параметры элементов взаимодействующей подсистемы. Обратное влияние на тепловую подсистему учитывается введением в эквивалентную схему тепловой подсистемы объекта зависимых источников теплового потока. При отсутствии массопереноса значение источника определяется мощностью, выделяемой элементом, при наличии массопереноса – переносимой массой. При моделировании влияние температуры без учета обратного воздействия предполагают, как будет изменяться температура во времени, и переносят эту зависимость на параметры элементов объекта, т. е. делают параметры зависимыми от времени. Аналогично моделируют старение элементов объекта при расчетах надежности.

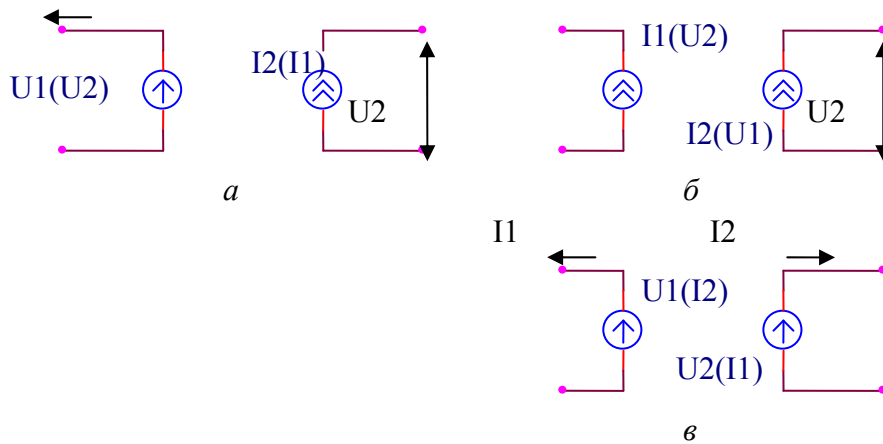


Рис. 5.4

Известен ряд методов формирования математических моделей систем при ее рассмотрении на втором уровне описания: обобщенный, узловой, табличный, метод переменных состояния. Различие между методами заключается в выборе исходных топологических уравнений и способе упорядочивания фазовых переменных объекта. Фазовые переменные делятся на предварительно исключаемые переменные, на этапе формирования ММС и на исключаемые на каждой итерации при решении системы уравнений ММС (4.8). Исключение проводится по методу Гаусса. Первая группа переменных исключается аналитически, а вторая – численно.

Общим для всех методов является исходная совокупность компонентных $F_K(\frac{dV}{dt}, V, t) = 0$ и топологических $F_T(V) = 0$ уравнений, где V – вектор фазовых переменных.

Кроме матрицы инциденций A , как отмечено выше, для описания топологии объекта используется матрица контуров и сечений M (M -матрица). Метод, основанный на использовании информации, заключенной в M -матрице, – наиболее удобный и общий метод получения топологических уравнений.

M -матрица строится на основании ориентированного графа эквивалентной схемы объекта и выбранного для этого графа дерева. Количество столбцов матрицы соответствует числу ветвей дерева, а количество строк – числу хорд. Процедура формирования M -матрицы заключается в следующем: каждая хорда графа поочередно включается в дерево, при этом образуется замкнутый контур (главный контур); выполняется обход этого контура в направлении, заданном направлением хорды; в строке матрицы, соответствующей данной хорде, ставится +1; если направление ветви дерева совпадает с направлением обхода контура, –1; если направление ветви дерева противоположно; 0, если ветвь не входит в данный контур.

Пример некоторой простой эквивалентной схемы устройства СУ и соответствующего ей графа приведен на рис. 5.5.

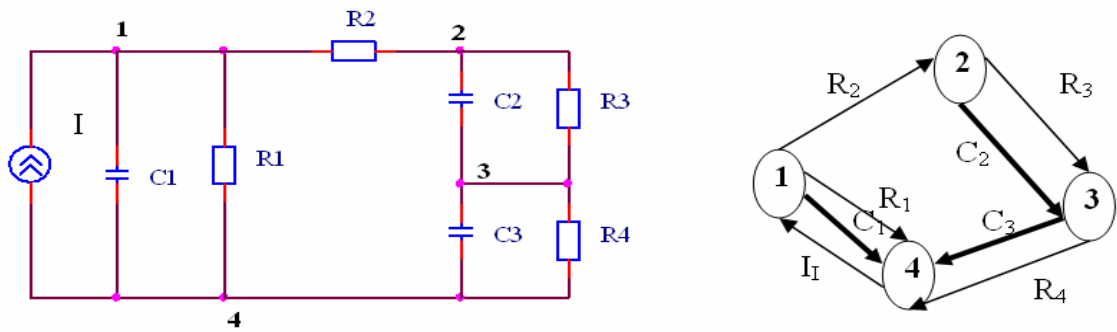


Рис. 5.5

Рассмотрим получение матрицы контуров и сечений для данного графа, начиная с выбора фундаментального дерева. Выбор дерева однозначно определяет векторы $U_{ВД}$, U_X , $I_{ВД}$, I_X ($U_{ВД}$, U_X – векторы переменных типа разностей потенциалов на ветвях дерева и хордах; $I_{ВД}$, I_X – векторы переменных типа потока для ветвей дерева и хорд) и приводит к записи топологических уравнений с использованием M -матрицы в виде

$$M \cdot U_{ВД} + U_X = 0, \quad (5.6)$$

$$I_{ВД} - M_T \cdot I_X = 0. \quad (5.7)$$

Для схемы на [рис. 5.5](#) M -матрица представлена в виде [табл. 5.2](#).

Таблица 5.2

Обозначение	Ветвь C1	Ветвь C2	Ветвь C3
Хорда R1	- 1	0	0
Хорда R2	- 1	+ 1	+ 1
Хорда R3	0	- 1	0
Хорда R4	0	0	- 1
Хорда II	+ 1	0	0

Отметим, что уравнения [\(5.6\)](#) представляют собой уравнения непрерывности (в случае электрических подсистем уравнения второго закона Кирхгофа или ему аналогичные согласно аналогиям топологических уравнений для подсистем другой физической природы) для контуров, образованных поочередным подключением каждой из хорд в отдельности к дереву. А уравнения [\(5.7\)](#) – уравнения равновесия (уравнения первого закона Кирхгофа) для сечений ветвей дерева, т. е. для таких сечений, при которых пересекаются некоторые хорды и единственная ветвь дерева.

Количество топологических уравнений равно количеству ветвей эквивалентной схемы рассматриваемого объекта.

Использование топологических матриц A и M позволяет автоматизировать процесс составления уравнений Кирхгофа, поскольку процесс составле-

ния этих матриц можно автоматизировать. При этом, если уравнения законов Кирхгофа составлены с помощью топологических матриц, то эти уравнения всегда будут линейно независимыми и, образованная с их помощью математическая модель объекта не будет вырожденной, т. е. всегда будет иметь решение. Каждое из уравнений сечений (5.7) будет отличаться от других тем, что содержит ток ветви дерева $I_{ВД}$, которого нет в других уравнениях (в соответствии с принципом построения сечений дерева), а каждое из контурных уравнений (5.6) будет отличаться от других тем, что содержит напряжение на хорде, которого нет в других уравнениях (в соответствии с принципом построения главных контуров).

Базис *метода переменных состояния* составляют переменные состояния – переменные типа потока через элементы типа $C(I_C)$, переменные типа разности потенциалов на элементах типа $L(U_L)$ и производные переменных состояния.

Метод переменных состояния – единственный из методов формирования ММ системы (устройства), который позволяет получить математическую модель в нормальной форме. В общем случае методика получения ММ системы в нормальной форме состоит из двух этапов. Пусть имеется система топологических и компонентных уравнений, причем компонентными уравнениями реактивных элементов служат дифференциальные зависимости

$$I_C = C \frac{dU_C}{dt}; U_L = L \frac{dI_L}{dt}. \quad (5.8)$$

Первый этап заключается в том, что на основе законов Кирхгофа переменные состояния выражаются через токи и напряжения остальных ветвей:

$$I_C = f_1 \text{ (остальные переменные);} \quad (5.9)$$

$$U_L = f_2 \text{ (остальные переменные).}$$

На втором этапе I_C и U_L заменяются производными в соответствии с (5.8), а остальные переменные с помощью комбинирования законов Кирхгофа и компонентных уравнений выражаются через U_C и I_L . В результате вместо системы топологических и компонентных уравнений получим ММ системы в нормальной форме:

$$\frac{dU_C}{dt} = \frac{1}{C} f_1(U_C(t), I_L(t)) \quad (5.10)$$

$$\frac{dI_L}{dt} = \frac{1}{L} f_2(U_C(t), I_L(t))$$

Рассмотрим получение ММС на примере RLC -схемы.

1. Составим эквивалентную схему объекта

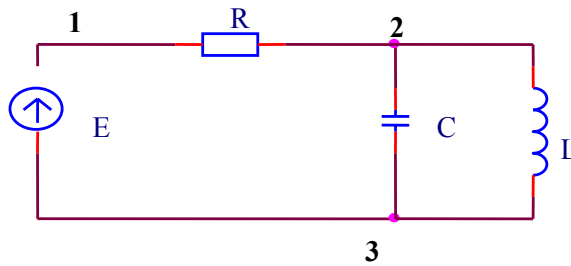


Рис 5.6

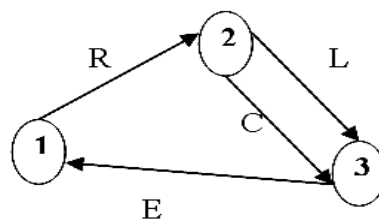


Рис. 5.7

2. Строим граф эквивалентной схемы ([рис. 5.7](#)). Граф практически повторяет эквивалентную схему, но без условных изображений ветвей.

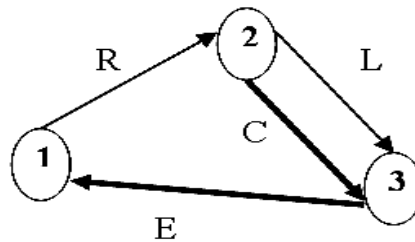


Рис. 5.8

3. Выбираем нормальное дерево графа. Если в схеме имеются контуры из U -элементов или звезды из I -элементов, то необходимо преобразование одного из U -элементов в I -элемент и исключение его из дерева или соответственно преобразование одного из I -элементов в U -элемент и включение его в дерево. На [рис. 5.8](#) ветви дерева представлены утолщенными линиями.

4. Строим матрицу контуров и сечений, где столбцы соответствуют ветвям дерева, а строки – хордам ([табл. 5.3](#)).

Таблица 5.3

Обозначение	Ветвь E	Ветвь C
Хорда R	+ 1	+ 1
Хорда L	0	- 1

5. Для получения топологических уравнений сканируем M -матрицу по строкам и столбцам. При сканировании по строкам получаем уравнения непрерывности (неразрывности), при сканировании по столбцам – уравнения равновесия. При получении уравнений непрерывности знаки элементов матрицы меняются на противоположные:

$$U_E + U_C + U_R = 0,$$

$$U_C - U_L = 0,$$

$$I_E - I_R = 0,$$

$$I_C - I_R + I_L = 0.$$

6. Добавляем компонентные уравнения всех ветвей:

$$U_E = \text{const},$$

$$U_R = I_R R,$$

$$\frac{dU_C}{dt} = \frac{1}{C} I_C,$$

$$\frac{dI_L}{dt} = \frac{1}{L} U_L.$$

7. Получаем нормальную форму Коши, раскрывая правые части последних двух уравнений с комбинированием топологических и оставшихся компонентных уравнений:

$$\frac{dU_C}{dt} = \frac{1}{C} \left(\frac{U_E - U_C(t)}{R} - I_L(t) \right),$$

$$\frac{dI_L}{dt} = \frac{U_C(t)}{L}.$$

Далее, используя численный метод интегрирования, получаем ММС в алгебраизованной форме. Таким образом, в методе переменных состояния

предварительная алгебраизация компонентных уравнений не требуется, поэтому при программной реализации метода библиотека математических моделей элементов не связана с библиотекой методов интегрирования.

Назовем основные недостатки метода переменных состояния:

1. Сложность формирования ММС в виде (5.10). Нормальная форма Коши не может быть получена, если в ветви дерева попадет ветвь типа L или в хорды – ветвь типа C .

2. Ориентация на явные методы решения системы уравнений ММС. Явные методы практически нельзя использовать для расчета схем с большим разбросом постоянных времени, т. е. жестких ММС, из-за ограничений на шаг расчета.

3. Реализация неявных методов для системы (5.10) затруднительна из-за сложности вычисления матрицы Якоби.

Узловой метод формирования ММ системы – метод, базис которого составляют переменные типа потенциала (узловые потенциалы). В основе узлового метода лежит уравнение равновесия, т. е. сумма переменных типа потока в узлах эквивалентной схемы равна нулю

$$I(\varphi) = 0, \quad (5.11)$$

где I – вектор переменных типа потока, φ – вектор переменных типа потенциала, характеризующих состояние узла.

Теоретическая модель-алгоритм, используемая для расчета схемы, обычно формируется в виде, соответствующем решению (5.11) методом Ньютона

$$Y(\varphi^p) \Delta\varphi^p = -I(\varphi^p), \quad (5.12)$$

где $Y = \frac{\partial I}{\partial \varphi}$ – матрица узловых проводимостей, p – индекс ньютоновских итераций, $\Delta\varphi^p = \varphi^{p+1} - \varphi^p$ – вектор поправок, $I(\varphi^p)$ – вектор узловых токов (вектор невязок).

После этого считаем, что $\varphi^{p+1} = \Delta\varphi^p + \varphi^p$, и снова формируем и решаем систему (5.12). Цикл заканчивается при выполнении одного из следующих условий $\|\Delta\varphi^p\| \leq \varepsilon_1$, либо $\|I(\varphi)\| \leq \varepsilon_2$ или по их комбинации, где ε_1 – задаваемая норма вектора приращений, ε_2 – задаваемая норма вектора невязок. Цикл также может быть завершен неудачно по превышению числа итераций.

Таким образом, для получения математической модели объекта в узловом базисе необходимо сформировать матрицу узловых проводимостей и вектор невязок.

Методика формирования ММС состоит в последовательном рассмотрении каждого элемента схемы и определения его вклада в вектор узловых токов и матрицу узловых проводимостей.

Формирование вектора узловых токов состоит в образовании для каждого узла схемы суммы полюсных токов элементов, соединенных с данным узлом. При этом считается, что ток, втекающий в узел, входит в узловой ток данного узла со знаком «-», а вытекающий – со знаком «+».

При формировании матрицы узловых проводимостей используется принцип позиционного суммирования, заключающийся в том, что элемент Y_{ik} матрицы $Y(\varphi)$ получается суммированием всех проводимостей элементов схемы, включенных между узлами i и k , в том числе взаимной проводимости y_{ik} , которая входит в состав матрицы Y_{pj-n} –полюсника, если i и k являются его полюсами. Аналогично в состав элемента Y_{ii} матрицы $Y(\varphi)$ входит как слагаемое собственная проводимость y_{ii-n} –полюсника, если i -й узел схемы одновременно является полюсом n -полюсника. Например, любой биполярный транзистор можно рассматривать как трехполюсник с матрицей проводимостей

$$Y_{\text{ТР}} = \begin{bmatrix} y_{\text{ЭЭ}} & -y_{\text{ЭБ}} & -y_{\text{ЭК}} \\ -y_{\text{ЭБ}} & y_{\text{ББ}} & -y_{\text{КБ}} \\ -y_{\text{ЭК}} & -y_{\text{КБ}} & y_{\text{КК}} \end{bmatrix}.$$

Рассмотрим формирование ММС для схемы, представленной на [рис.5.5, а](#). Размерность математической модели определяется числом узлов схемы минус 1, т. е. в нашем случае она равна трем ([рис. 5.9](#)).

$$\begin{bmatrix} Y_I + Y_{C1} + \frac{1}{R_1} + \frac{1}{R_2} & -\frac{1}{R_2} & 0 \\ -\frac{1}{R_2} & Y_{C2} + \frac{1}{R_2} + \frac{1}{R_3} & -\frac{1}{R_3} - Y_{C2} \\ 0 & -\frac{1}{R_3} - Y_{C2} & Y_{C2} + Y_{C3} + \frac{1}{R_3} + \frac{1}{R_4} \end{bmatrix} \times \begin{bmatrix} \Delta\varphi_1 \\ \Delta\varphi_2 \\ \Delta\varphi_3 \end{bmatrix} =$$

$$= \begin{bmatrix} -I + (\varphi_1 - \varphi_2) \frac{1}{R_2} \\ -(\varphi_1 - \varphi_2) \frac{1}{R_2} + (\varphi_2 - \varphi_3) \left(\frac{1}{R_2} + Y_{C2} \right) \\ -(\varphi_2 - \varphi_3) \left(\frac{1}{R_2} + Y_{C2} \right) + \varphi_3 \left(\frac{1}{R_4} + Y_{C3} \right) \end{bmatrix}$$

Рис. 5.9

Как следует из (5.11), допустимый вид компонентного уравнения $I = I(\varphi)$, т. е. напрямую узловым методом может быть применен к анализу статических состояний. Но одна из основных задач анализа объекта на макроуровне – это анализ объекта во временной области, т. е. анализ динамических характеристик. Динамические процессы в объекте определяются реактивными элементами типа C и L . Если привести компонентные уравнения элементов C и L к виду $I = I(\varphi)$, то можно говорить и об анализе динамики. Компонентное уравнение для элемента типа C дискретизируем с помощью какого-либо метода численного интегрирования, например, с помощью неявного метода Эйлера:

$$\text{Эйлера: } \frac{dU_{Cn}}{dt} = \frac{U_{Cn} - U_{Cn-1}}{h_n}, \quad h_n - \text{ шаг интегрирования в } n\text{-й момент}$$

времени:

$$I_{Cn} = \frac{C}{h_n}(U_{Cn} - U_{Cn-1}) = \frac{C}{h_n}(\varphi_i - \varphi_j) - \frac{C}{h_n}U_{Cn-1}.$$

Таким образом, на одном шаге численного интегрирования мы получили компонентное уравнение в допустимом виде, и модель элемента типа C может быть представлена в виде

$$\begin{bmatrix} \frac{C}{h_n} & -\frac{C}{h_n} \\ -\frac{C}{h_n} & \frac{C}{h_n} \end{bmatrix} \times \begin{bmatrix} \Delta\varphi_i \\ \Delta\varphi_j \end{bmatrix} = - \begin{bmatrix} \frac{C}{h_n}(\varphi_i - \varphi_j) - \frac{C}{h_n}U_{Cn-1} \\ -\frac{C}{h_n}(\varphi_i - \varphi_j) - \frac{C}{h_n}U_{Cn-1} \end{bmatrix}.$$

Аналогично для элемента типа L :

$$U_{Ln} = L \frac{dI_{Ln}}{dt} = \frac{L}{h_n}(I_{Ln} - I_{Ln-1})$$

или

$$I_{Ln} = I_{Ln-1} + \frac{h_n}{L}(\varphi_i - \varphi_j).$$

Математическая модель для элемента L выглядит следующим образом:

$$\begin{bmatrix} \frac{h_n}{L} & -\frac{h_n}{L} \\ -\frac{h_n}{L} & \frac{h_n}{L} \end{bmatrix} \times \begin{bmatrix} \Delta\varphi_i \\ \Delta\varphi_j \end{bmatrix} = - \begin{bmatrix} I_{Ln-1} + \frac{h_n}{L}(\varphi_i - \varphi_j) \\ -I_{Ln-1} - \frac{h_n}{L}(\varphi_i - \varphi_j) \end{bmatrix}.$$

Основные достоинства узлового метода:

1. Малая размерность математической модели объекта;
2. Отсутствие проблемы топологических вырождений, поскольку узловые потенциалы независимо от топологии схемы всегда образуют невырожденный базис;
3. Простота формирования ММ системы (ММС) в виде алгоритма, которому соответствует ММС в неявной форме и, как следствие, простота реализации неявных методов решения ОДУ;
4. Простые алгоритмы работы с многополюсными элементами, что позволяет разрабатывать библиотеки ММЭ с вложенными элементами.

Недостатки узлового метода:

1. Ограничение на вид компонентного уравнения;
2. Методы численного интегрирования ОДУ при формировании ММС встраиваются в компонентные уравнения реактивных ветвей.

Современные автоматизированные *методы формирования дискретных устройств СУ* в САПР основаны на использовании входных языков логического уровня, которые чаще всего создаются как непроцедурные. Логический уровень характеризуется использованием моделей, составляемых из моделей отдельных логических элементов (триггеров, элементов И-НЕ, ИЛИ-НЕ) Эти модели могут отражать выполнение логических функций с учетом временных задержек. Модель, отражающая протекающие в схеме процессы, называется асинхронной. Модель, отражающая отдельные состояния схемы (чаще всего установившиеся), называется синхронной. Другими словами, асинхронная модель – это имитационная модель, представленная средствами функционально-логического уровня.

Описание схемы на входном языке задаются в виде списка элементов моделируемой схемы. В каждой строке списка указывают: тип очередного элемента, его имя или имена его выходов, связи входов элементов с выходами других элементов, возможно указание задержки. Все связи имеют свои имена, обычно совпадающие с именами соответствующих выходов элементов. Отдельные строки служат для перечисления выходов элементов, являющихся выходами всей схемы, и задания входных последовательностей, реакция на которые должна быть получена как результат моделирования.

Примером языка моделирования дискретных электронных устройств на логическом уровне может служить язык VHDL (Very high-speed integrated circuits Hardware Design Language), утвержденный в качестве международного стандарта IEEE 1076 в 1987 году. Язык VHDL успешно используется и при синтезе устройств [2].

В дальнейшем стандарт корректировался и расширялся, новые версии приняты в 1993 и 1999 годах. Так, версия 1999 году, получившая индекс IEEE 1076.1, содержит средства описания аналоговых и смешанных моделей.

Эти описания вместе с базовым вариантом языка VHDL образуют язык VHDL-AMS (VHDL – Analog and Mixed Signal).

VHDL – не единственный язык проектирования дискретных устройств СУ. Так, для проектирования интегральных схем широко применяется Verilog, близкий к C++ язык SystemC, язык описания цепей EDIF, языки проектирования устройств на ПЛИС и др., но для сквозного проектирования на всех требуемых для СБИС уровнях функционального проектирования (выше схемотехнического) наибольшее распространение получили VHDL и Verilog.

В VHDL имеются средства для поведенческих и структурных описаний. Описание схемы на языке VHDL включает две части.

Первая часть – описание схемы как компонента некоторой надсистемы, т. е., прежде всего, описание интерфейсов схемы с внешней средой. Вид этой части, называемой сущностью, – *entity*:

```
entity имя_сущности is
```

```
<декларации>
```

```
...
```

```
end имя_сущности;
```

Назначение второй части – описание внутренних свойств схемы (структуры или функций), она называется архитектурным телом:

```
architecture имя_архитектурного_тела of имя_сущности is
```

```
<декларации>
```

```
...
```

```
begin
```

```
описание_схемы_или_алгоритма
```

```
end имя_архитектурного_тела;
```

У одной схемы (сущности) может быть несколько архитектурных тел, так как могут быть описаны разные аспекты (структура или алгоритмы) и версии объекта, причем на разных иерархических уровнях.

В декларациях архитектурного тела объявляются типы, фигурирующие в данном архитектурном теле. Ими могут быть используемые типы компонентов (объекты), параметры и сигналы. В декларации могут входить также описания процедур, функций, типов данных.

Различают структурное и поведенческое описание сущностей. Структурные описания служат для представления структурных схем, а поведенческие описания служат для представления функций и алгоритмов, выполняемых устройством.

Формирование математических моделей систем управления

При рассмотрении системы управления на верхнем (первом при использовании нисходящего проектирования) уровне описания (рис. 4.3) происходит дальнейшее абстрагирование от физических особенностей функционирования устройств и элементов системы [6; 12]. На этом уровне описываются в основном информационные процессы в СУ при прохождении фазовой переменной типа сигнал от входа к выходу. Математическое описание СУ зависит от применяемого математического аппарата, который в свою очередь определяет и методы формирования ММ СУ. Необходимо отметить, что получение математических моделей на первом уровне описания СУ наиболее трудноформализуемый процесс, по сравнению с подходами, принятыми на низших уровнях описания: уровень элементов и уровень устройств системы [10].

При формировании описания СУ на первом уровне используют ряд упрощений и допущений, главные из которых формулируются следующим образом:

- 1) однонаправленность в передаче сигналов, т. е. использование моделей, в которых отсутствует влияние выходных переменных на состояние входных переменных;
- 2) отсутствие влияния нагрузки на параметры и состояние моделируемого объекта;
- 3) использование вместо фазовых переменных двух типов U и I , переменных одного типа, называемых сигналом;
- 4) линейность моделей инерционных элементов.

Для математических моделей первого уровня характерны те же типы уравнений, что и для ММ уровня устройств СУ, но эти уравнения включают фазовые переменные, описывающие состояние укрупненных элементов системы. Если определен закон непрерывного перехода объекта из одного состояния в другое, то для анализа ММ СУ используют аппарат передаточных функций, а при рассмотрении состояний объекта в дискретные моменты времени ОДУ и их системы переходят в разностные уравнения относительно значений фазовых переменных в эти моменты времени. В случае дискретного множества состояний объектов применяют также аппарат математической логики и конечных автоматов.

Иногда для сложных информационных систем удается перейти к дискретному представлению фазовых переменных. Тогда ММ СУ становится системой логических соотношений (СЛС), описывающей процессы преобразования сигналов. Использование СЛС применительно к таким сложным системам более экономично, чем описание изменения в электрических цепях

информационной системы напряжений и токов как непрерывных функций времени при помощи ОДУ или их систем.

Значит, для формирования математических моделей СУ на первом уровне описания используются те же методы, что и для ММ второго уровня описания СУ, где в качестве базовых будут рассматриваться укрупненные элементы системы.

Получение имитационных ММ и ММ массового обслуживания, описывающие функционирование сложных вычислительных и информационных систем, производственных участков, линий, цехов, предприятий и их объединений, возможно на основе универсальных языков программирования, но получающаяся при этом программная реализация имитационных моделей СМО оказывается громоздкой, длительность разработки может превысить разумные пределы. Поэтому применяют проблемно ориентированные языки имитационного моделирования: общецелевые предназначенные для моделирования различных объектов, представляемых моделями СМО (GPSS, СТРУМ и др.); специализированные языки, ориентированные на анализ узкого класса объектов, например, для моделирования вычислительных систем (язык МПЛ/ВС). При этом увеличивается эффективность в разработке и исполнении программ.

При разработке сложных моделей перед кодированием имитационной модели объекта на выбранном языке моделирования целесообразно увеличивать наглядность и представлять структуру модели с помощью сетей Петри. Графическое изображение сети Петри удобно для восприятия и легко преобразуется в исходную программу имитационного моделирования. Для кодирования применяются специальные языки сетей Петри и общецелевые языки имитационного моделирования.

ТЕМА 4. МЕТОДЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ: МЕТОДЫ АНАЛИЗА ССУ

Лекция 6. Методы анализа ССУ в САПР и требования к ним

Методы анализа систем управления в САПР. Виды анализа как проектной процедуры при автоматизированном проектировании ССУ. Требования к методам анализа в САПР.

Методы анализа систем управления в САПР

Задачи анализа средств и систем управления связаны с исследованием характеристик в зависимости от условий функционирования системы. Результаты анализа дают ответ на вопрос, какими свойствами обладает исследуемый объект, удовлетворяет ли он требованиям ТЗ, но не объясняют, как улучшить свойства объекта [7].

Анализ технических систем в САПР основан на математическом моделировании, т. е. на исследовании проектируемых СУ путем оперирования их ММ. В САПР СУ инструменты анализа непосредственно связаны с инструментами синтеза. В частности, задачи синтеза часто удается свести к многократному решению соответствующих задач анализа. Инженерный анализ СУ традиционно проводится алгебраическими, частотными и корневыми методами с привлечением методов гармонической линеаризации, вариантов метода малого параметра. Применение традиционных «ручных» методов к сложным, особенно нелинейным, СУ при решении практических задач в большинстве случаев оказывается затруднительным. Применение САПР позволяет найти проектное решение [8].

Традиционные методы анализа СУ изложены в многочисленных работах по теории управления и, как правило, являются скалярными, предназначенными для простых систем невысокого порядка. Такой подход был оправдан при неавтоматизированном проектировании, когда анализ и расчет осуществлялись путем приближенных теоретических исследований в сочетании с макетированием и испытаниями, которые составляли большую часть инструментария проектировщика.

Машинная ориентация традиционных методов с целью применения их в САПР состоит в том, чтобы распространить их на многомерные системы высокого порядка, качество СУ определять не по одному, а по многим критериям, упростить и ускорить процедуру получения конечных результатов, улучшить представление результатов (графиков, таблиц и т. д.) с помощью средств САПР.

Традиционно анализ СУ сводится к анализу устойчивости, качества и точности, что соответствует расчету статических и динамических характеристик СУ. Реализация данных методов анализа СУ в САПР требует переделки и видоизменения исходных классических алгоритмов к программной специфике. Автоматизация проектной процедуры анализа технических систем сводится к оперированию их ММ при проведении вычислительного эксперимента и означает решение соответствующих уравнений.

Виды анализа как проектной процедуры при автоматизированном проектировании ССУ

При анализе переходных процессов определяются зависимости фазовых переменных от времени при заданных значениях внутренних и внешних параметров технического объекта. Анализ во временной области (динамических характеристик) позволяет получить картину переходных процессов, оценить динамические свойства системы, он является важной процедурой при исследовании как линейных, так и нелинейных систем. Математическая модель технической системы, описывающая ее динамические свойства, представляет собой систему обыкновенных дифференциальных уравнений (ОДУ) вида [5]:

$$F\left(\frac{dV}{dt}, V, t\right) = 0. \quad (6.1)$$

Следовательно, методы анализа переходных процессов технических систем, используемые в универсальных программах анализа САПР, есть методы решения систем ОДУ, т. е. методы численного интегрирования. Другими словами, это методы алгебраизации дифференциальных уравнений. Формулы интегрирования систем ОДУ могут входить в ММ независимо от компонентных уравнений или быть интегрированными в ММ компонентов.

От выбора метода решения систем ОДУ существенно зависят такие характеристики анализа, как точность и вычислительная эффективность. Эти характеристики определяются, прежде всего, типом и порядком выбранного метода интегрирования систем ОДУ. Инженер-пользователь САПР должен представлять возможности методов интегрирования, заложенные в программу анализа, уметь оценить вычислительные затраты, сформулировать требования к точности интегрирования, диагностировать причины возможных отказов при работе программ анализа. В универсальных САПР постоянные интегрирования (минимальный и максимальный шаг интегрирования, точность расчета) задаются обычно по умолчанию. Причем задаваемая точность намного выше требуемой для практических задач. Изменяя постоянные интегрирования, можно значительно повысить эффективность анализа, которая в свою очередь зависит от конкретной программной и алгоритмической реализации.

Для большинства проектируемых динамических систем переходные процессы носят асимптотически устойчивый характер, т.е. при $t \rightarrow \infty$ СУ переходит в определенное устойчивое состояние. Расчет значений фазовых переменных системы в устойчивом состоянии есть анализ установившихся процессов или расчет статических характеристик. Статический режим можно рассчитать, интегрируя исходную систему ОДУ (6.1) на достаточно большом интервале времени – метод установления или «расчет статики через динамику». Достоинства такого подхода является единство алгоритмов расчета для анализа переходных и установившихся процессов. Недостатком – увеличение временной емкости программы анализа.

При анализе статического режима производные фазовых переменных в ММ СУ (6.1) $\frac{dV}{dt}$ равны нулю и нет внешних воздействий, меняющихся во времени. Следовательно, ММ системы для анализа статических характеристик может быть получена из ММ (6.1) и представляет собой систему алгебраических уравнений (АУ) относительно соответствующих фазовых переменных вида

$$F(V) = 0. \quad (6.2)$$

Значит, методы анализа установившихся процессов технических систем, используемые в универсальных программах анализа в САПР, есть методы решения систем АУ. Необходимо отметить, что многие методы решения ОДУ и АУ сводятся к решению систем линейных алгебраических уравнений (ЛАУ).

При анализе частотных характеристик технической системы определяются зависимости фазовых переменных от частоты при заданных значениях внутренних и внешних параметров СУ. Анализ в частотной области более специфичен, его применяют, как правило, к объектам с линеаризуемыми ММ при исследовании колебательных стационарных процессов, анализе устойчивости, расчете искажений информации, представляемой спектральными составляющими сигналов, и т. п.

ММ технической системы для анализа частотных характеристик получается из (6.1) при переходе из временной в частотную область с помощью преобразования Фурье, что приводит к получению системы линейных алгебраических уравнений (ЛАУ) с комплексными коэффициентами

$$AV = B, \quad (6.3)$$

где A – матрица коэффициентов размерностью $n \times n$; B – вектор правых частей. Следовательно, методы анализа частотных характеристик СУ, используемые в универсальных программах анализа в САПР, есть методы решения систем ЛАУ.

Количество методов и алгоритмов для решения систем линейных и нелинейных алгебраических уравнений, ОДУ в вычислительной математике велико, однако использование большинства из них в САПР оказывается неэффективным из-за особенностей ММ проектируемых объектов.

Наиболее существенные особенности используемых ММ технических систем, которые проявляются более чем на одном уровне абстрагирования:

– Высокая размерность модели (так как одной из причин автоматизации проектирования является увеличение сложности СУ) обусловлена применением ММ, обеспечивающих высокоточные результаты. Размерность ММ чаще всего выражается порядком соответствующей системы уравнений.

– Разреженность матриц в ММ (т. е. матрицы с большим количеством нулевых элементов) обусловлена тем, что каждый элемент проектируемого объекта имеет непосредственные связи со сравнительно небольшим числом других элементов. Разреженность оценивается отношением числа нулевых элементов к общему числу элементов матрицы (основная причина распространения матричных представлений связана с удобством программирования матричных процедур, использованием операторов цикла, следовательно, матричные формулировки ведут к экономии времени программирования).

– Плохая обусловленность модели (модели, при использовании которых малые погрешности исходных данных приводят к значительным погрешностям результата) называют плохо обусловленными. Из-за конечности разрядной сетки при решении систем уравнений неизбежны ошибки из-за округлений промежуточных результатов. Эти ошибки в процессе решения увеличиваются и приводят к возникновению несходимости или неустойчивости вычислений.

Поэтому при автоматизированном проектировании СУ очень важно знать и правильно выбрать численный метод решения систем уравнений ММ системы.

Требования к методам анализа в САПР

При создании математического обеспечения САПР усилия направлены не только на разработку ММ, но и на развитие численных методов и алгоритмов анализа. Так как эффективность метода зависит от особенностей решаемой задачи, то в САПР целесообразна реализация более чем одного метода для каждого класса решаемых уравнений [6].

К методам и алгоритмам анализа в САПР предъявляют следующие требования:

– Экономичность метода характеризуется затратами вычислительных ресурсов (машинного времени T и памяти Π). На экономичность влияет, прежде всего, размерность ММ (N). При сравнении методов исследуют характер зависимости $T(N)$, $\Pi(N)$. Наиболее эффективные методы имеют линейную зависимость показателей экономичности от сложности решаемой задачи. Для многих численных методов характерна полиномиальная зависимость от $T(N)$.

– Надежность метода оценивается как вероятность получения правильных результатов при использовании метода для решения задач заданного класса. Недостатки высоконадежных методов – низкая экономичность. В этом случае целесообразно комбинирование методов с переходом к трудоемким, но надежным методам только в результате автоматического распознавания ситуаций несходимости или неустойчивости вычислений

– Точность. Погрешности решения определяются особенностями использования ММ, численных методов, разрядностью сетки ЭВМ. Каждый источник погрешности должен контролироваться, чтобы погрешности не превысили предельно допустимые.

В математическом обеспечении САПР целесообразно иметь не один, а несколько методов одинакового целевого назначения с разными возможностями для удовлетворения требований точности и экономичности.

Ошибки при анализе могут быть и при ошибках в задании исходных данных. Примерами таких ошибок могут быть ошибки в задании численных значений параметров или в задании соединений элементов в анализируемой схеме.

Актуальной задачей создания и совершенствования математического обеспечения САПР ССУ является повышение экономичности методов анализа при соблюдении требований точности. Основными способами повышения экономичности, применяемыми на разных уровнях проектирования, являются:

– Учет разреженности матриц – организация операций над разреженными матрицами. Отказ от хранения нулевых элементов, реализация алгоритмов, игнорирующих действия над нулевыми элементами, что уменьшает затраты времени и памяти;

– Использование диакоптических и декомпозиционных методов, т. е. методов исследования сложных систем по частям (основа блочно-иерархического подхода к проектированию). Экономичность увеличивается за счет минимизации числа обменов информацией между ОЗУ и ВЗУ, распараллеливанием вычислений, использование для каждой части системы подходящих алгоритмов анализа;

– Учет событийности, т. е. на каждом шаге вычислительного процесса обрабатывается информация, относящаяся только к активизированным элементам общей модели (т. е. к элементам, у которых изменение фазовой переменной превышает пороговое значение). В имитационных моделях, в логических схемах доля активизированных элементов уменьшается, следовательно, повышается эффективность учета событийности;

– Многоуровневое адаптивное моделирование – в общей модели системы для различных фрагментов используют модели для разных иерархических уровней (например, логические и электрические) и возможность автоматической смены фрагментов в процессе вычислений, что позволяет использовать модели, близкие к оптимальным с точки зрения удовлетворения требований точности и экономичности.

Лекция 7. Алгоритмы и методы анализа статических режимов ССУ в интегрированных САПР

Методы анализа в частотной области, их основные характеристики. Методы анализа статических характеристик. Оценка эффективности методов анализа СУ при их использовании в САПР

Методы анализа в частотной области, их основные характеристики

Процедура решения системы ЛАУ (6.3) $AV = B$ занимает важное место в математическом обеспечении интегрированных САПР. Решение системы ЛАУ является не только самостоятельной задачей (анализ частотных характеристик), но и одной из наиболее частой повторяющихся процедур при решении других задач (например, систем нелинейных АУ, при анализе динамики). При этом эффективность решения задач анализа в значительной мере зависит от эффективности использования методов решения ЛАУ [13].

Методы решения систем ЛАУ делятся на две группы: прямые и итерационные. Прямые методы используют конечные соотношения для вычисления неизвестных. Их достоинства: просты, универсальны, дают решение после выполнения заранее известного числа операций. Недостатки: требуют хранения всей матрицы A в ОЗУ, следовательно, при увеличении n увеличивается объем занимаемой памяти; не учитывают структуру матрицы A ; накапливают погрешность вычисления, так как на любом этапе используются результаты предыдущих операций. Прямые методы используют для ММ с $n < 200$ [6; 10].

Итерационные методы – методы последовательных приближений. В них необходимо задать некоторое приближенное решение – начальное приближение. После этого с помощью некоторого алгоритма проводится один цикл вычислений, называемый итерацией. В результате итерации находят новое приближение. Итерации проводятся до получения решения с требуемой точностью. Недостатки: сложные алгоритмы решения; заранее трудно определить объем вычислений; сходимость итераций может быть медленной. Достоинства: не нужно хранить матрицу всей системы ЛАУ, можно вычислять элементы матрицы по мере необходимости; погрешности результатов не накапливаются, так как точность вычислений в каждой итерации определяется лишь результатами предыдущей, следовательно, итерационные методы можно использовать для систем ЛАУ с большой размерностью матрицы A и для плохо обусловленных систем. Итерационные методы могут использоваться для уточнения решений, полученных с использованием прямых методов.

В программах автоматизации проектирования итерационные методы решения используют редко, только в специальных случаях, когда гарантиру-

ется быстрая сходимость итераций. В большинстве случаев используются прямые методы, обеспечивающие решения за конечное число операций.

При решении систем ЛАУ прямыми методами на ЭВМ основными проблемами являются: высокая разреженность матрицы A ; ошибки округлений из-за конечности разрядной сетки. Основные прямые методы решения систем ЛАУ можно разделить на три группы.

1) Методы разложения матрицы A на сомножители – решение системы ЛАУ сводится к решению аналогичной задачи, но с матрицами простого вида. Основные затраты времени и памяти при решении систем ЛАУ с использованием методов разложения матрицы A на сомножители определяются на подготовительном этапе получения соответствующих сомножителей (прямой ход), затраты же на само решение (обратный ход) оказываются на порядок меньше;

а) для матрицы A предполагается, что найдены некоторые матрицы сомножители C, S, G , удовлетворяющие соотношению $CAS = G$, тогда справедливы соотношения $V = SU$, где U – решение системы $GU = D$, а правая часть $D = CB$. Решение сводится к расчёту D по $D = CB$, U по $GU = D$, V по $V = SU$. В программах анализа в САПР чаще всего применяют метод Гаусса или его разновидности. Метод Гаусса – метод последовательного исключения неизвестных из системы уравнений. При исключении k -й неизвестной v_k из системы уравнений $AV = B$ все коэффициенты a_{ij} при $i > k$ и $j > k$ пересчитывают по формуле

$$a_{ij} = a_{ij} - a_{ik} a_{kj} / a_{kk}. \quad (7.1)$$

Исключение $n - 1$ неизвестных, где n – порядок системы (6.3), называют прямым ходом, в процессе которого матрица коэффициентов приобретает треугольный вид. При обратном ходе последовательно вычисляют неизвестные, начиная с v_n . В общем случае число арифметических операций для решения (6.3) по Гауссу пропорционально n^3 . Это приводит к значительным затратам машинного времени, поскольку система ЛАУ решается многократно в процессе одновариантного анализа и существенно ограничивает сложность анализируемых объектов. Заметно повысить вычислительную эффективность анализа можно, если использовать характерное практически для всех приложений свойство высокой разреженности матрицы A в модели (6.3).

Матрицу называют разреженной, если большинство ее элементов равно нулю. Эффективность обработки разреженных матриц велика, потому что, во-первых, не нужно делать пересчета по формуле (7.1), если хотя бы один из элементов a_{ik} или a_{kj} оказывается нулевым; во-вторых, не требуются затраты памяти для хранения нулевых элементов. Хотя алгоритмы обработки разреженных матриц более сложны, но в результате удается получить затраты машинного времени, близкие к линейным, например, затраты оказываются пропорциональными n^2 .

При использовании методов разреженных матриц нужно учитывать зависимость вычислительной эффективности от того, как представлена матрица коэффициентов A , точнее, от того, в каком порядке записаны ее строки и столбцы. Для пояснения этой зависимости рассмотрим два варианта представления одной и той же системы ЛАУ.

В первом случае система уравнений имеет вид

$$\begin{aligned}a_{11}v_1 + a_{12}v_2 + a_{13}v_3 + a_{14}v_4 + a_{15}v_5 &= b_1, \\a_{21}v_1 + a_{22}v_2 &= b_2, \\a_{31}v_1 + a_{33}v_3 &= b_3, \\a_{41}v_1 + a_{44}v_4 &= b_4, \\a_{51}v_1 + a_{55}v_5 &= b_5.\end{aligned}$$

При прямом ходе в соответствии с формулой (7.1) все элементы матрицы, которые первоначально были нулевыми, становятся ненулевыми, а матрица оказывается полностью насыщенной. Элементы, становящиеся ненулевыми в процессе гауссовых исключений, называют вторичными ненулями. Во втором случае меняются местами первое и пятое уравнения. Теперь вторичные нули не появляются, матрица остается разреженной, высокая вычислительная эффективность сохраняется.

Таким образом, методы разреженных матриц должны включать в себя способы оптимального упорядочения строк и столбцов матриц. Используют несколько критериев оптимальности упорядочения. Простейшим из них является критерий расположения строк в порядке увеличения числа первичных нулей, более сложные критерии учитывают не только первичные нули, но и появляющиеся вторичные нули.

Методом разреженных матриц называют метод решения системы ЛАУ на основе метода Гаусса с учетом разреженности (первичной и вторичной) матрицы коэффициентов. Метод разреженных матриц можно реализовать путем интерпретации и компиляции. В обоих случаях создаются массивы ненулевых коэффициентов матрицы (с учетом вторичных нулей) и массивы координат этих ненулевых элементов. При этом выигрыш в затратах памяти довольно значителен. Так, при матрице умеренного размера 200×200 без учета разреженности потребуются 320 кбайт.

В случае интерпретации моделирующая программа для каждой операции по (7.1) при $a_{ik} \neq 0$ и $a_{jk} \neq 0$ находит, используя указатели, нужные коэффициенты и выполняет арифметические операции по (7.1). Поскольку система ЛАУ в процессе анализа решается многократно, то и операции поиска нужных коэффициентов также повторяются многократно, на что, естественно, тратится машинное время.

Способ компиляции более экономичен по затратам времени, но уступает способу интерпретации по затратам памяти. При компиляции поиск нужных для (7.1) коэффициентов выполняется однократно перед численным решением задачи. Вместо непосредственного выполнения арифметических

операций для каждой из них компилируется команда с найденными адресами ненулевых коэффициентов. Такие команды образуют рабочую программу решения системы ЛАУ, которая и будет решаться многократно. Очевидно, что теперь в рабочей программе будет выполняться минимально необходимое число арифметических операций.

б) Матрица A представляется в виде $A = LU$, затем система $AV = B$ решается как две подсистемы $Lq = B$, $UV = q$. Матрицы L и U должны быть найдены явно и иметь вид, обеспечивающий удобство решения. L – нижнетреугольная матрица; U – верхнетреугольная, все диагональные элементы которой равны 1. Вектор B не изменяется.

При $n = 4$

$$A_{LU} = \begin{vmatrix} l_{11} & u_{12} & u_{13} & u_{14} \\ l_{21} & l_{22} & u_{23} & u_{24} \\ l_{31} & l_{32} & l_{33} & u_{34} \\ l_{41} & l_{42} & l_{43} & l_{44} \end{vmatrix}.$$

Матрица A в методе LU -разложения может быть представлена в виде произведения треугольных матриц двумя вариантами:

Метод Краута – Холецкого

$$\begin{vmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{vmatrix} = \begin{vmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{n1} & \dots & l_{nn} \end{vmatrix} \times \begin{vmatrix} 1 & \dots & u_{1n} \\ 0 & 1 & u_{2n} \\ 0 & 0 & 1 \end{vmatrix}.$$

Метод Дулитла

$$\begin{vmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{vmatrix} = \begin{vmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ l_{31} & l_{32} & \dots & \dots \\ l_{n1} & \dots & \dots & 1 \end{vmatrix} \times \begin{vmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ 0 & 0 & \dots & \dots \\ 0 & 0 & \dots & u_{nn} \end{vmatrix}.$$

На первом этапе происходит разложение матрицы A

$$l_{ij} = a_{ij}^{(i-1)},$$

$$u_{ii} = 1, u_{ij} = \frac{a_{ij}^{(i-1)}}{l_{ii}} \quad (j > i).$$

Затем пересчитываются элементы a_{ij} очередного j -го столбца

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} \cdot u_{kj} \quad \text{при } i = j, j+1, \dots, n$$

j -й строки

$$u_{ji} = \frac{a_{ji} - \sum_{k=1}^{j-1} l_{ik} \cdot u_{kj}}{l_{ii}} \quad \text{при } j = i+1, \dots, n.$$

На втором этапе происходит получение решения системы (6.3) – вектора V .

Сначала решается система с матрицей L , $Lq = B$ с помощью алгоритма, аналогичного обратному ходу метода Гаусса (только сверху вниз)

$$q_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} \cdot q_j}{l_{ii}} \quad \text{при } i = 2, 3, \dots, n,$$

затем решается система $UV = q$ по формулам обратного хода метода Гаусса. Элементы $b_{ii} \neq 0$.

Наиболее часто в современных программах автоматизации проектирования при анализе статических режимов используют методы разложения матрицы A на сомножители. Опыт показал, что наиболее приемлемыми из них являются компактные схемы Гаусса – одна из разновидностей метода LU -разложения (под L и U понимается нижняя и верхняя треугольные матрицы) и метод Гаусса.

Оба метода в случае полностью заполненных матриц примерно равноценны, однако при использовании разреженных матриц предпочтение отдаётся LU -разложению, обеспечивающему большие удобства организации программы учёта разреженности.

2) Методы обращения матрицы – решение системы ЛАУ отыскивается в виде $V=A^{-1}B$. В практике автоматизации схмотехнического проектирования методы обращения матриц не используются, так как для вычисления матрицы A^{-1} нужно больше операций, чем при решении системы ЛАУ методами первой группы.

3) Методы решения систем ЛАУ с матрицей A специального вида используют тогда, когда этот вид может заранее гарантирован.

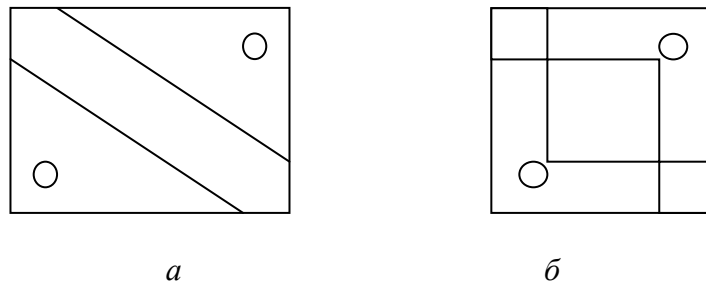


Рис.7.1. *a* – ленточно-диагональная; *б* – блочно-диагональная

Например, электронные схемы с цепочной структурой без ОС имеют ленточную структуру матрицы. Для решения систем с ленточными матрицами наиболее эффективным является метод прогонки при решении трехдиагональных систем

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & a_{23} & 0 & \dots & 0 \\ 0 & a_{32} & a_{33} & a_{34} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}.$$

Метод прогонки состоит из двух этапов: прямая прогонка, заключающаяся в расчёте прогоночных коэффициентов

$$l_{i+1} = \frac{-a_{i, i+1}}{a_{i, i} + a_{i, i-1} \cdot l_i},$$

$$p_{i+1} = \frac{b_i - a_{i, i-1} \cdot p_i}{a_{i, i} + a_{i, i-1} \cdot l_i}.$$

$$l_1 = p_1 = 0 \text{ при } i = 1, 2, \dots, n.$$

Обратная прогонка состоит в расчёте V_i с $V_n = p_n + 1$,

$$V_i = l_{i+1} V_{i+1} + p_{i+1} \text{ при } i = n-1, n-2, \dots, 1.$$

Достоинство: вместо хранения в памяти трёх диагоналей матрицы A и вектора B можно организовать хранение только двух векторов L и P , необходимых для обратной прогонки. Эта задача решается на этапе формирования ММ, причем модель формируется не в виде $AV = B$, а в виде векторов L и P . Такой подход позволяет увеличить предельные возможности программы анализа.

Основными методами решения систем ЛАУ второй группы – итерационными методами являются линейные стационарные методы первого порядка, формула которых имеет вид:

$$V_{i+1} = HV_i + C,$$

где $i = 0, 1, 2, \dots$ – номер итерации; H – итерационная матрица, выбор которой является решающим; C – соответствующий известный вектор.

Метод линейный и стационарный, так как H и C не зависят от номера итерации и от V_i . Порядок метода определяется количеством предыдущих значений V , использованных в данном методе. Основные характеристики данного метода – сходимость итераций и скорость сходимости к точному решению V^* .

Выражение $V_{i+1} = HV_i + C$ сходится, если для любого начального приближения V_0 последовательность V_1, V_2, \dots , определяемая с помощью линейного стационарного метода, сходится к V^* . Необходимое и достаточное условие сходимости:

$$\rho(H) < 1,$$

где $\rho(H)$ – спектральный радиус матрицы H .

В зависимости от выбора H и C получаются различные итерационные методы: простой итерации, Зейделя, релаксационные. Для практической реализации итерационных методов необходимо выбрать способ ускорения сходимости и установить критерий окончания итерационного процесса.

Введем вектор ошибки на k -ой итерации $E_k = V_k - V^*$. Можно показать, что $\|E_k\| \leq \|H^k\| \times \|E_0\|$, где $\|E\|, \|H\|$ – соответствующие векторные и матричные нормы. Следовательно, $\|H_k\|$ определяет, во сколько раз была уменьшена норма ошибки после k -итераций, т. е. скорость сходимости. Способы ускорения сходимости весьма разнообразны, но часто основываются на оценке максимального $M(H)$ и минимального $m(H)$ по модулю собственных значений матрицы H . Идеальным критерием окончания итераций является норма вектора ошибки E_k , но непосредственно ее определить невозможно, так как нет точного решения V^* . Поэтому для итерационного процесса при решении системы (6.3) вводится вектор приращений $\Delta V_k = V_{k+1} - V_k$, связанный с вектором ошибки следующим равенством

$$\Delta V_k = (H - I)E_k,$$

где I – единичная матрица. Переходим к оценке по нормам $\|E_k\| \leq \|(H - I)^{-1}\| \times \|\Delta V_k\|$. Это соотношение можно представить в виде

$$\|E_k\| \leq \|(1 - \rho(H))^{-1}\| \times \|\Delta V_k\|. \quad (7.2)$$

Из (7.2) следует, что вектор приращений можно использовать в качестве критерия окончания итераций, но при $\rho(H) \rightarrow 1$ применяют соотношение и оценивают $M(H)$.

Недостатками итерационных методов являются малая скорость сходимости, низкая точность. Достоинствами – экономичность с точки зрения затрат времени и памяти на одну итерацию.

Рассмотрим применение метода Гаусса – Зейделя (его преимущества: простота, лёгкость программирования; линейная скорость сходимости) на примере решения системы

$$a_{11}V_1 + a_{12}V_2 + a_{13}V_3 = b_1,$$

$$a_{21}V_1 + a_{22}V_2 + a_{23}V_3 = b_2,$$

$$a_{31}V_1 + a_{32}V_2 + a_{33}V_3 = b_3.$$

Предположим, что диагональные элементы отличны от нуля (в противном случае можно переставить уравнения). Выразим неизвестные из первого, второго и третьего уравнений системы

$$v_1 = \frac{1}{a_{11}}(b_1 - a_{12}v_2 - a_{13}v_3),$$

$$v_2 = \frac{1}{a_{22}}(b_2 - a_{21}v_1 - a_{23}v_3),$$

$$v_3 = \frac{1}{a_{33}}(b_3 - a_{31}v_1 - a_{32}v_2).$$

Зададим некоторые начальные (нулевые) приближения значений неизвестных $V_1 = V_1^{(0)}, V_2 = V_2^{(0)}, V_3 = V_3^{(0)}$. Подставляя эти значения в правую часть приведенных выше выражений, получаем новое приближение для неизвестных:

$$v_1^{(1)} = \frac{1}{a_{11}}(b_1 - a_{12}v_2^{(0)} - a_{13}v_3^{(0)}),$$

$$v_2^{(1)} = \frac{1}{a_{22}}(b_2 - a_{21}v_1^{(1)} - a_{23}v_3^{(0)}),$$

$$v_3^{(1)} = \frac{1}{a_{33}}(b_3 - a_{31}v_1^{(1)} - a_{32}v_2^{(1)}).$$

Итерационный процесс продолжается, пока значения $V_1^{(k)}, V_2^{(k)}, V_3^{(k)}$ не станут близкими с заданной погрешностью к значениям $V_1^{(k-1)}, V_2^{(k-1)}, V_3^{(k-1)}$.

Развитие метода Зейделя – метод релаксации.

$$V_i^{(k+1)} = V_i^{(k)} + (\omega V_i - V_i^{(k)}),$$

где ω – релаксационный параметр.

Условие сходимости (10.6) выполняется, если $0 < \omega < 2$. Оптимальным значением релаксационного параметра называется значение $\omega_{\text{опт}}$, при котором достигается минимум $\rho(H_\omega)$. На практике для многих случаев

$$\omega_{\text{опт}} = \frac{2}{(1 + \sqrt{1 - M(H)})},$$

так как $\rho(H) < 1$, то $1 < \omega_{\text{опт}} < 2$.

Методы анализа статических характеристик

Математическая модель технической системы для анализа статических режимов, как указывалось выше, есть система нелинейных алгебраических уравнений (НАУ) вида [6]:

$$F(V) = 0, \quad (7.3)$$

где V – вектор фазовых переменных.

В отличие от систем ЛАУ для решения систем НАУ не существует прямых методов решения, так как не удается получить аналитическое решение. Поэтому основными методами решения систем НАУ являются итерационные методы, в основе которых лежит принцип последовательных приближений, выполняющихся по формуле [7, 10]:

$$V^{k+1} = \psi(V^k),$$

где $\psi(V^k)$ – некоторый функционал исходной системы, который строится так, чтобы обеспечить выполнение условия $\lim_{k \rightarrow \infty} V^k = V^*$, где $V^* = (V_1^*, V_2^*, \dots, V_n^*)$ – решение системы.

Основными характеристиками методов решения систем НАУ являются скорость сходимости и область сходимости, определяемая условиями сходимости [12].

Скорость сходимости оценивается по изменению расстояния до точки решения в двух последовательных итерациях. Различают линейную скорость сходимости, если выполняется условие $\|V^* - V^{k+1}\| < a\|V^* - V^k\|$ и квадратичную скорость сходимости $\|V^* - V^{k+1}\| < a\|V^* - V^k\|^2$, где $a \leq 1$, V^* – точка решения, $\|V^* - V^k\|$ – какая-либо норма вектора. Теоретические условия сходимости различны для различных методов и не пригодны в практических расчетах схем. Исходя из этого, на практике, учитывая, что V^* неизвестна, проверку сходимости метода выполняют на каждой итерации по условию

$$\|V^{k+1} - V^k\| < \|V^k - V^{k+1}\|. \quad (7.4)$$

Условие (7.4) не априорное условие сходимости, а лишь средство контроля сходимости. Область значений фазовых переменных, в которой выполняется условие (7.4), называется областью сходимости. Сходящийся вычислительный процесс заканчивается при выполнении условия

$$\|V^{k+1} - V^k\| < \varepsilon_1 \quad \text{или} \quad |F(V_i)| < \varepsilon_2, \quad \varepsilon_1 > 0, \varepsilon_2 > 0.$$

При расчете электронных устройств СУ эти условия обычно неизвестны и гарантии сходимости нет, следовательно, первой проблемой является обеспечение высокой алгоритмической надежности решения, т. е. сходимости в возможно большей области начальных приближений и в более широком диапазоне численных значений параметров функции $F(V) = 0$. При расчете больших схем ($n > 100$) кроме алгоритмической надежности метода добавляется требование минимизации вычислительных затрат, т. е. уменьшения объема вычислений на одну итерацию, что характеризуется временем выполнения одной итерации и объемом памяти необходимого для реализации метода. Численных методов, удовлетворяющих этим требованиям, нет, поэтому при принятии решения о включении методов решения систем НАУ в состав САПР необходимо обеспечивать компромисс.

Рассмотрим основные классы методов решения систем НАУ, используемых в САПР при анализе технических систем.

Первая группа – прямые или итерационные методы решения систем НАУ:

- метод простых итераций;
- метод Ньютона;

метод Ньютона – Брайдена.

При решении системы уравнений ММ технического объекта методом простых итераций ее представляют в виде:

$$V_1 = F_1(V_1, V_2, \dots, V_n),$$

$$V_2 = F_2(V_1, V_2, \dots, V_n),$$

.....

$$V_n = F_n(V_1, V_2, \dots, V_n).$$

Алгоритм напоминает метод Гаусса – Зейделя. Пусть в результате предыдущей итерации получены значения неизвестных $V_1 = a_1, V_2 = a_2, \dots, V_n = a_n$, тогда выражение для следующих итераций:

$$V_1 = F_1(a_1, a_2, \dots, a_n),$$

$$V_2 = F_2(V_1, a_2, \dots, a_n),$$

.....

$$V_n = F_n(V_1, V_2, \dots, V_{n-1}, a_n).$$

Итерационный процесс продолжается, пока изменения неизвестных в двух последовательных итерациях не станут малыми: $|V_i^k - V_i^{k-1}| < \delta$. По скорости сходимости метод простых итераций относится к методам с линейной скоростью сходимости. Условия сходимости для метода простых итераций:

$$\max_j \sum_i \left| \frac{\partial f_i}{\partial v_j} \right| < 1$$

или

$$\max_i |\lambda_i| < 1,$$

где λ – собственные значения матрицы Якоби. Количественные формулировки условий сходимости для других методов решения систем НАУ сложны и на практике не используются.

При включении метода простых итераций в состав САПР нужно учитывать следующие недостатки: начальные приближения должны быть близкими к истинному решению, иначе итерации могут не сходиться; начальное приближение V_0 задается пользователем на основе знания им приближенного

решения системы (предметной области исследования объекта проектирования), либо выбирается автоматически, как в методе движущейся в области сходимости.

Среди итерационных методов решения систем НАУ в САПР часто используют метод Ньютона. Его главное достоинство – высокая скорость сходимости. В основе лежит использование разложения функции $F_i(V_1, V_2, \dots, V_n) = 0$ в ряд Тейлора (члены со вторыми и выше производными отбрасываются). Решение ищут в виде $V_{i+1} = V_i + \Delta V_i$, где ΔV_i – поправка на $(i+1)$ -итерацию.

Приближение к корню V^* производится с выбранного исходного приближения V_0 . В методе Ньютона поправка определяется как $\Delta V_i = -Y_i^{-1} \cdot F(V_i)$, где Y_i^{-1} – обратная матрица Якоби, вычисленная на i -ой итерации. Однако вычисление Y_i^{-1} неэффективно, поэтому используют решение системы ЛАУ $Y_i \cdot \Delta V_i = -F(V_i)$.

Пусть известны приближенные значения неизвестных a_1, a_2, \dots, a_n . Задача состоит в нахождении приращений к этим значениям $\Delta v_1, \Delta v_2, \dots, \Delta v_n$, благодаря которым решение системы запишется в виде

$$\begin{aligned} v_1 &= a_1 + \Delta v_1, \\ v_2 &= a_2 + \Delta v_2, \\ &\dots\dots\dots \\ v_n &= a_n + \Delta v_n. \end{aligned} \tag{7.5}$$

Разложим (7.3) с учётом (7.5) в ряд Тейлора

$$\begin{aligned} F_1(v_1, \dots, v_n) &\approx F_1(a_1, \dots, a_n) + \frac{\partial F_1}{\partial v_1} \cdot \Delta v_1 + \dots + \frac{\partial F_1}{\partial v_n} \cdot \Delta v_n, \\ &\dots\dots\dots \\ F_n(v_1, \dots, v_n) &\approx F_n(a_1, \dots, a_n) + \frac{\partial F_n}{\partial v_1} \cdot \Delta v_1 + \dots + \frac{\partial F_n}{\partial v_n} \cdot \Delta v_n. \end{aligned} \tag{7.6}$$

Левые части системы (7.6) равны нулю, отсюда получим систему ЛАУ относительно приращений

$$\begin{cases} \frac{\partial F_1}{\partial v_1} \cdot \Delta v_1 + \dots + \frac{\partial F_1}{\partial v_n} \cdot \Delta v_n = -F_1 \\ \dots\dots\dots \\ \frac{\partial F_n}{\partial v_1} \cdot \Delta v_1 + \dots + \frac{\partial F_n}{\partial v_n} \cdot \Delta v_n = -F_n. \end{cases} \tag{7.7}$$

Значения F_1, F_2, \dots, F_n и их производные вычисляются при $v_1 = a_1, v_2 = a_2, \dots$ и т. д. Определителем системы (7.7) является Якобиан. Для существования единственного решения $Y \neq 0$ на каждой итерации.

Процесс решения системы НАУ методом Ньютона состоит в определении приращений ΔV_i к значениям V_i на каждой итерации. При $\max |\Delta V_i| < \varepsilon$ решение прекращается. Сходимость ухудшается с увеличением n .

Достоинства применения метода Ньютона при решении систем НАУ для анализа статических характеристик объектов проектирования, по сравнению с другими методами: увеличивается скорость сходимости, так как у метода Ньютона квадратичная скорость сходимости (по сравнению с линейной скоростью метода простых итераций); всегда имеет область сходимости для реальных схем. Недостатками являются малый размер области сходимости (меньше чем для метода простых итераций) и V_0 нужно выбирать близко к V^* ; увеличиваются затраты памяти для размещения матрицы Якоби. Метод Ньютона эффективен, если $F(V)$ – выпуклая функция, решение с $\varepsilon = 10^{-3} \div 10^{-4}$ достигается за 3–5 итераций.

Первый недостаток метода Ньютона привел к появлению второй группы методов решения систем НАУ, методов, повышающих вероятность сходимости метода Ньютона – методы продолжения решения по параметру. Эта группа методов делится на две подгруппы:

- 1) с ограниченным параметром:
 - метод с одной итерацией;
 - метод движущейся области сходимости;
- 2) с неограниченным параметром (методы установления):
 - явные;
 - неявные.

В методах с ограниченным параметром в решаемой системе уравнений выделяют параметр, влияющий на положение точки корня в пространстве фазовых переменных. Например, напряжение источника питания $U_{ип}$. Система уравнений ММ проектируемого объекта решается методом Ньютона многократно при ступенчатом изменении параметра.

Пусть при $U_{ип}$, стремящемся к нулю, точное решение V^* также стремится к нулю. Тогда при первом решении выбираем $V_0 = 0$ и находим значение V_1^* , соответствующее $U_{ип1}$. Затем увеличиваем $U_{ип}$ и решаем систему НАУ при $V_0 = V_1^*$ и так далее до решения системы уравнений при истинном значении $U_{ип}$.

Методы установления – методы анализа, основанные на интегрировании дифференциальных уравнений ММ технического объекта на большом интервале времени, т. е. в качестве неограниченного параметра выбирается время. Как уже отмечалось, достоинством такого подхода является единство алгоритмов расчета для анализа переходных и установившихся процессов. Недостатком – увеличение временной емкости программы анализа.

Третья группа методов – методы оптимизации. Эти методы основаны на эквивалентности результатов решения системы $F(V) = 0$ итерационными методами и минимизации какого-либо из функционалов

$$\Phi_1(V) = \sum_i F_i^2(V), \quad \Phi_2(V) = \sum_i |F_i(V)|$$

хорошо отработанными методами оптимизации. Количество решений системы $F(V) = 0$ равно количеству минимумов. Таким образом, локальные методы оптимизации можно использовать при расчёте статического режима схем с одним состоянием равновесия (ключи, усилители). Если схема имеет несколько состояний равновесия (триггеры и т. д.), то локальные методы могут и не дать нужного решения. Недостаток: уступает методу Ньютона по скорости сходимости. В целом методы оптимизации не нашли широкого использования при расчете схем.

Оценка эффективности методов анализа СУ при их использовании в САПР

Решение систем АУ как математического описания СУ имеет место во многих процедурах функционального проектирования. От решения системы АУ зависит эффективность выполнения проектной процедуры анализа СУ, поэтому при разработке проектирующей подсистемы нужно правильно выбрать метод решения системы АУ для его включения в состав САПР. При этом нужно учитывать, что численный метод еще не является программируемым алгоритмом, т. е. совокупностью отдельных операций, протекающих в определенной последовательности с определенным началом и достижимым концом. Выбор определенного метода и его преобразование в алгоритм и программу – попытка оптимизировать исходные положения и поставленные дополнительные требования [10].

К исходным положениям отнесем следующие положения:

- постановка задачи и общие предположения решения;
- дополнительная информация об исходных данных (специальный тип матриц, вид числового материала и др.);
- вычислительные ресурсы, язык программирования.

Дополнительными требованиями являются:

- специальные требования к выходным данным (например, точность);
- степень универсальности (должна решаться единичная задача или разрабатывается универсальное математическое и программное обеспечение САПР);

- минимизация времени счета, ограничение памяти.

Эти условия выбора метода анализа технического объекта или метода решения системы уравнений ММ проектируемого объекта противоречивы,

поэтому применяют эмпирические правила выбора методов при решении об их включении в состав САПР [12]:

- основной принцип выбора метода есть принцип непосредственного решения, т. е. следует выбирать метод, который решает именно поставленную задачу, а не ведет к решению через решаемые подзадачи (например, собственные значения вычисляются методом Якоби, а не как корни характеристического многочлена);
- «математически элегантные» решения часто являются непросматриваемыми относительно распространения ошибок и неустойчивости и численно неблагоприятными;
- теоретическая сходимость при численной реализации не будет гарантией сходимости (теоретически расходящиеся методы могут быть численно пригодны).

Ко всему перечисленному: ход процесса решения системы уравнений ММ СУ определяется параметрами управления, значения которых вычисляются в ходе реализации программы. Так, условиями останова, как правило, являются неравенства

$$\left| V^{k+1} - V^k \right| < \varepsilon_1 \quad \text{или} \quad \left| F(V_i) \right| < \varepsilon_2, \quad \varepsilon_1 > 0, \varepsilon_2 > 0 .$$

В первом случае возникает опасность потери точности при образовании разности, во втором случае при выборе ε_2 малым ошибка округления может исказить обрыв или ему помешать.

На эффективность используемого метода оказывают влияние не только особенности самого метода, но и особенности решаемой задачи: размерность системы уравнений (n); число обусловленности (z); разрешенность матрицы Якоби (s). Кроме этого, полезно учитывать особенности архитектурной платформы, в частности, быстродействие (Б).

При выборе метода анализа необходимо ориентироваться на некоторые диапазоны значений этих факторов, характерных для моделей проектируемых объектов в соответствующей предметной области.

Выбор метода решения систем АУ основан на взаимосвязи $T_m = n\gamma I / Б$, где T_m – затраты машинного времени; n – порядок системы АУ, принимаемый за оценку сложности задачи; γ – среднее число арифметических операций, приходящихся на единицу сложности задач на одной итерации; I – среднее число итераций.

Для метода Гаусса $I = 1$ при $n < 50$ $\gamma \approx 2 \cdot \left(n \cdot \frac{2}{3} + 2 \cdot n \right)$, при учете разреженности ($n > 50$) $\gamma = \text{const}$. Подсчитано, что $\gamma \approx 25$ для ключевых электронных схем; $\gamma \approx 8$ для математических моделей с трехдиагональной матрицей.

Для итерационных методов решения систем ЛАУ $I \gg 1$, $\gamma = 2K$, где K – среднее арифметическое для числа ненулевых элементов в одной строке матрицы A . Так, для моделей переключательных электронных схем по результатам статистических исследований $\gamma = 7,8$, т. е. одна итерация выполняется быстрее, чем по методу Гаусса. Однако из-за того, что $I \gg 1$ итерационные методы по показателю T_m практически всегда проигрывают прямым методам.

Решение систем НАУ выполняется итерационными методами, при этом на требуемое число итераций I в методе Ньютона решающее влияние оказывает выбор начального приближения; в других итерационных методах – z (число обусловленности) матрицы Якоби решаемой системы уравнений. В методе Ньютона, применяемом в рамках методов установления или продолжения решения по параметру, обычно I не превышает трех ($I \leq 3$). В случаях, если I превышает некоторый порог $I_{пр}$ (например, $I_{пр} = 7$), лучше уменьшать значения коэффициентов, управляющих процессом установления, чем продолжать итерации при $I > I_{пр}$. При решении НАУ величина γ растет, так как при ее подсчете должны быть учтены затраты на вычисление элементов матрицы Якоби.

В методе простых итераций I может достигать неприемлемо больших значений, поэтому на I вводится ограничение $I_{гр} = 1,5 \cdot 10^4$. Из соотношения $I_{гр} = -0,5 \cdot z \cdot \lg \varepsilon$, при $\varepsilon = 10^{-3}$ получаем, что метод простых итераций можно использовать только при решении системы НАУ, у которых матрица Якоби имеет $z < 10^4$. Методы Зейделя, Якоби, последовательной верхней релаксации имеют аналогичный характер зависимости I от z , хотя скорость сходимости у них часто оказывается несколько выше, чем в методе простых итераций.

Экономичность метода решения системы АУ определяется затратами ОЗУ. На хранение матрицы Якоби нужно n^2 ячеек памяти. В задачах анализа распределенных моделей, в которых $n > 10^4$, экономичность становится важной характеристикой.

Точность решения определяется обусловленностью и значениями погрешностей ε_1 и ε_2 , задаваемых пользователем САПР. Задаваемые значения ε_1 и ε_2 могут оказаться недостижимыми или из-за несходимости, или из-за слишком медленной сходимости вычислительного процесса. Поэтому если создаваемая САПР ориентирована на решение систем уравнений с широким диапазоном значений z , то разработчику системы нужно принимать специальные меры по обеспечению точности решения.

Надежность метода определяется не только фактом сходимости к корню, но и тем, каковы затраты времени T_m на получение решения с требуемой точностью. Ненадежность итерационных методов проявляется либо при неудачном выборе начального приближения к точному решению (метод Ньютона), либо при плохой обусловленности задачи (метод простых итераций), либо при повышенных требованиях к точности решения (метод простых итераций), либо при высокой размерности задачи (метод Гаусса при неучете

разреженности). Поэтому при создании специализированных САПР необходимы предварительный анализ особенностей ММ заданного класса задач (значений n , z , допустимых погрешностей) и соответствующий выбор конкретного метода. При создании универсальных САПР необходима реализация средств адаптации метода решения к конкретным условиям. Такая адаптация чаще всего применяется в рамках методов установления или продолжения решения по параметру.

Таким образом, алгоритмы и методы анализа статических режимов СУ в современных САПР есть прежде всего алгоритмы и методы решений линейных и нелинейных алгебраических уравнений математической модели СУ для разных уровней описания систем. При этом применение традиционных частотных методов анализа устойчивости и качества трансформируется в обработку результатов численного решения систем ЛАУ. Например, в широко применяемом формате Spice проектировщику для получения наглядных характеристик системы ЛАХ и ФЧХ достаточно ввести эквивалентную схему замещения устройства или СУ, выполнить анализ частотных характеристик (расчет системы ЛАУ ММ устройства или СУ) и вывести графические результаты моделирования с использованием правил языка проектирования.

Лекция 8. Алгоритмы и методы анализа ССУ во временной области в интегрированных САПР

Методы анализа СУ во временной области. Методы анализа СУ с «жесткими» системами уравнений. Основные характеристики методов анализа СУ во временной области. Алгоритмы автоматического выбора шага. Выбор эффективных методов анализа переходных процессов СУ.

Методы анализа СУ во временной области

Анализ временных оценок качества СУ осуществляется численными методами решения системы уравнений ММ, которая может быть представлена: в виде системы дифференциальных уравнений в неявной форме

$F\left(\frac{dV}{dt}, V, t\right) = 0$; в виде системы дифференциальных уравнений в явной форме

ме $\frac{dV}{dt} = F(V, t)$, где V – вектор фазовых переменных размерностью n .

По исходной ММ СУ строится переходный процесс в системе, по которому и делаются нужные оценки [5; 6].

Анализ переходных процессов проектируемого объекта сводится к численному интегрированию системы ОДУ и нахождению $V(t)$ на заданном интервале времени $[0, T_{\text{кон}}]$, при заданных начальных условиях $V(0) = V_0$. При

решении этой задачи на интервале интегрирования выделяется конечное число точек t_k , в которых определяются значения V , $k = 1, 2, \dots, \text{III}$, где III – число шагов интегрирования [7].

В каждой точке t_k система ОДУ представляется в виде системы n алгебраических уравнений с $2n$ неизвестными. Система уравнений дополняется уравнениями, задаваемыми выбранным методом численного интегрирования.

Обобщенная формула методов интегрирования используется для решения систем ОДУ.

$$\alpha \cdot \dot{V}_k = G(V_k, V_{k-1}, \dots, V_{k-p}, \dot{V}_{k-1}, \dot{V}_{k-2}, \dots, \dot{V}_{k-p}), \quad (8.1)$$

где G – некоторая функция, определяемая способом построения метода интегрирования (чаще всего линейная); p – количество предыдущих точек, использованных в формуле интегрирования (порядок метода); α – коэффициент.

Если уравнения составляются для текущего k -го момента времени ($\alpha = 0$), то метод явный (иначе, экстраполяционный метод, основанный на формулах интегрирования вперед).

Если уравнения составляются для следующего $(k+1)$ -го момента времени ($\alpha = 1$), то метод неявный (интерполяционный, основанный на формулах интегрирования назад).

Различия между ними удобно показать на примере простейших методов первого порядка – методов Эйлера. Формула явного метода Эйлера представляет собой следующую формулу замены производных в точке t_n :

$$dV / dt | n = (V_{n+1} - V_n) / h_n,$$

где индекс равен номеру шага интегрирования; $h_n = t_{n+1} - t_n$ – размер шага интегрирования (обычно h_n называют просто шагом интегрирования). В формуле неявного метода Эйлера использовано дифференцирование назад:

$$dV / dt | n = (V_n - V_{n-1}) / h_n,$$

где $h_n = t_n - t_{n-1}$.

В зависимости от значения p методы интегрирования распадаются на два класса: 1) одношаговые, если $p = 1$; 2) многошаговые, если $p > 1$, причем p называют порядком многошагового метода (в формуле интегрирования используют результаты предыдущих шагов).

Классификация основных численных методов решения систем ОДУ

- 1) Явные:
 - блочные,
 - одношаговые,
 - многоэтапные методы Рунге – Кутты,
 - одноэтапные:
 - метод Эйлера,
 - метод разложения в ряд Тейлора,
 - многошаговые:
 - метод Адамса – Башфорта,
 - методы с сериями шагов.
- 2) Неявные:
 - одно- и двушаговые:
 - метод Эйлера,
 - метод трапеций.
 - Рунге – Кутты
 - многошаговые:
 - ФДН,
 - Адамса – Мултона,
 - блочные.

В современных САПР для решения систем ОДУ применяются наиболее распространенные методы ФДН (формулы дифференцирования назад), Адамса и Рунге – Кутты.

Общий вид формул дифференцирования назад:

$$\frac{dV_k}{dt} = \frac{\sum_{i=0}^p a_i \cdot V_{k-i}}{h_k},$$

$h_k = t_k - t_{k-1}$ – величина k -го шага интегрирования; a_i – коэффициент (значение зависит от порядка метода p).

Аппроксимация производных в точке t_k производится с помощью значений, относящихся к данному и предыдущему моментам времени. При $p = 1$ формула метода ФДН совпадает с неявной формулой Эйлера

$$\dot{V}_k = \frac{(V_k - V_{k-1})}{h_k}.$$

Часто используют формулу Гира второго порядка, называемую формулой Шихмана при $h = \text{const}$

$$\dot{V}_k = \left(\frac{3}{2} \cdot V_k - 2 \cdot V_{k-1} + \frac{1}{2} \cdot V_{k-2} \right) / h.$$

Формулы интегрирования при $p > 1$ в явных методах Адамса имеют вид

$$\dot{V}_{k-1} = \frac{a_1 \cdot (V_k - V_{k-1})}{h_k} + \sum_{i=2}^p a_i \cdot \dot{V}_{k-i},$$

неявных методах Адамса:

$$\dot{V}_k = \frac{a_0 \cdot (V_k - V_{k-1})}{h_k} + \sum_{i=1}^{p-1} a_i \cdot \dot{V}_{k-i},$$

явная формула Адамса при $p = 1$ называется явной формулой Эйлера:

$$\dot{V}_{k-1} = \frac{(V_k - V_{k-1})}{h_k},$$

при $p = 2$ метод Адамса называют методом трапеций:

$$\dot{V}_k = 2 \cdot \frac{(V_k - V_{k-1})}{h_k} - \dot{V}_{k-1}.$$

Методы ФДН являются наиболее распространенными современными методами интегрирования, основанными на представлении производной $\frac{dV}{dt}$ в точке t_k через значения V_k и i -ю предысторию, т. е. в точках $V_{k-1}, V_{k-2}, \dots, V_{k-i}$. Их основное преимущество перед другими методами – возможность смены не только шага интегрирования h , но и порядка метода p , что обеспечивает высокую точность расчетов характеристик.

Явные методы интегрирования целесообразно использовать при решении систем ОДУ в форме Коши. Достоинства явных методов: малый объем вычислений на одном шаге; уменьшение затрат памяти. Явные методы наиболее легко реализуются, приводят к сравнительно небольшому объему вычислений на одном шаге интегрирования. Однако для соблюдения условий устойчивости приходится уменьшать шаг настолько, что увеличившееся число шагов может сделать недопустимо большими общие затраты машинного времени. Поэтому явные методы, к которым относятся известные методы Адамса – Башфорта и явные варианты метода Рунге – Кутты, оказываются малонадежными и в САПР находят ограниченное применение.

Чаще всего используется формула, предложенная Рунге, основанная на вычислении приближенного решения y_1 в узле $x_0 + h$ в виде линейной комбинации с постоянными коэффициентами:

$$y_1 = y_0 + p_{q1}k_1(h) + p_{q2}k_2(h) + \dots + p_{qq}k_q(h), \quad (8.2)$$

$$k_1(h) = f(x_0, y_0),$$

$$k_2(h) = hf(x_0 + \alpha h, y_0 + \beta_{21}k_1(h)),$$

.....

$$k_q(h) = hf(x_0 + \alpha_q h, y_0 + \beta_{q1}k_1(h) + \dots + \beta_{q,q-1}k_{q-1}(h)).$$

Существует семейство формул типа Рунге – Кутты (8.2) четвертого порядка точности, для которых $q = s = 4$. Формула классического метода Рунге – Кутты

$$y_1 = y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$k_1 = hf(x_0, y_0),$$

$$k_2 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1),$$

$$k_3 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_2),$$

$$k_4 = hf(x_0 + h, y_0 + k_3).$$

Методы анализа СУ с «жесткими» системами уравнений

Особая группа – задачи с «жесткими» системами уравнений (чаще всего это системы дифференциальных уравнений высокого порядка, описывающие динамические системы). Для описания процесса нужно использовать функции двух типов: на малых отрезках – функции с быстро меняющимися параметрами (большие производные) и на остальной части – функции с малыми производными (медленные движения). Часто в системе ОДУ не удастся явно выделить члены с малыми параметрами, однако решение также имеет участки быстрых и медленных движений.

В частности, система ОДУ будет плохо обусловленной (жесткой), если выполняется условие

$$z = \frac{\max(\lambda_i)}{\min(\lambda_i)} \gg 1, \quad (8.3)$$

где z – число обусловленности матрицы A . Обычно математическую модель считают плохо обусловленной (а систему уравнений жесткой), если $z > 10^5$.

Метод Гира для жестких систем является многошаговым предсказывающе-исправляющим ФДН методом. При этом формула-предиктор и формула-корректор имеют один и тот же порядок.

Когда система уравнений является жесткой, интегрирование осуществляется специальным методом на основе формул дифференцирования назад (ФДН) вида

$$x_{m+1} = \sum_{i=1}^k a_i \cdot x_{m+1-i} + h b f_{m+1}.$$

В методе Гира эти формулы реализованы с помощью вектора Нордсика, позволяющего использовать эффективный алгоритм автоматического изменения порядка и величины шага. Последние компоненты вектора Нордсика позволяют получить оценку ошибки. При $k = 1$ (тогда $a_1 = b = 1$) получим неявный Эйлера – метод 1-го порядка.

Многошаговые формулы основаны на численном интегрировании, т. е. интеграл аппроксимируется с помощью какой-либо квадратурной формулы. Существуют многошаговые методы, воплотившие совсем другую идею – численного дифференцирования искомой функции. Рассмотрим вывод формул численного интегрирования для жесткой системы. Метод Гира строится на основе формул дифференцирования назад

$$\sum_{i=0}^k \alpha_i y_{n+i} - h \beta_k f(x_{n+k}, y_{n+k}) = 0, \alpha_k = 1. \quad (8.4)$$

Предположим, что начальное приближение для решения уравнения (8.4) находится по явной формуле вида

$$y_{n+k} = - \sum_{i=0}^{k-1} \alpha_i y_{n+k-i} + h \beta_{k-1} f(x_{n+k-1}, y_{n+k-1}). \quad (8.5)$$

Обозначим $m = n + k$ и перепишем формулу (8.5) в следующем виде:

$$y_m = - \sum_{i=0}^k \alpha_{k-i} y_{m-i} + h \beta_{k-1} f(x_{m-1}, y_{m-1}). \quad (8.6)$$

Обозначим $A_i = -\alpha_{k-i}$, $B_1 = \beta_{k-1}$ и заменим m на n . Тогда

$$y_n = \sum_{i=0}^k A_i y_{n-i} + h B_1 f(x_{n-1}, y_{n-1}).$$

Аналогично преобразуем формулу (8.4):

$$y_m = -\sum_{i=0}^k \alpha_{k-i} y_{m-i} + h\beta_k f(x_m, y_m).$$

Обозначим $\alpha_i = -\alpha_{k-i}$, $b_0 = \beta_k$ и заменим m на n . Тогда

$$y_n = -\sum_{i=0}^k \alpha_i y_{n-i} + hb_0 f(x_n, y_n).$$

Рассмотрим следующий итерационный процесс:

$$y_n^{(0)} = \sum_{i=0}^k A_i y_{n-i} + hB_1 f_{n-1}, \quad (8.7)$$

$$y_n^{(v+1)} = -\sum_{i=0}^k \alpha_i y_{n-i} + hb_0 f(x_n, y_n^{(v)}), \quad v = 0, 1, 2, \dots \quad (8.8)$$

Из (8.8) получаем

$$y_n^{(v+1)} = y_n^{(v)} + b_0 (hf(x_n, y_n^{(v)}) - hf(x_n, y_n^{(v-1)})), \quad v = 1, 2, \dots$$

Определим v -е приближение для производной $hy'(x_n)$ по формуле

$$hy_n^{(v)} = hf(x_n, y_n^{(v-1)}), \quad (8.9)$$

погрешность этого приближения для $hy'(x_n)$ имеет порядок $O(h^{k+2})$. Тогда

$$y_n^{(v+1)} = y_n^{(v)} + b_0 (hf(x_n, y_n^{(v)}) - hy_n^{(v)}), \quad v = 1, 2, \dots$$

Вычтем (8.7) из (8.8) при $v = 0$:

$$\begin{aligned} y_n^{(1)} &= y_n^{(0)} + \sum_{i=1}^k (\alpha_i - A_i) y_{n-i} + hb_0 f(x_n, y_n^{(0)}) - hB_1 f_{n-1} = \\ &= y_n^{(0)} + b_0 \left(hf(x_n, y_n^{(0)}) - \sum_{i=1}^k \frac{(A_i - \alpha_i)}{b_0} y_{n-i} - h \frac{B_1}{b_0} f_{n-1} \right). \end{aligned} \quad (8.10)$$

Обозначим $\gamma_i = (A_i - \alpha_i)/b_0$, $\delta_1 = B_1/b_0$ и определим начальное приближение для производной $hy'(x_n)$ по формуле

$$hy_n^{(0)} = \sum_{i=1}^k \gamma_i y_{n-i} + h\delta_1 f_{n-1}. \quad (8.11)$$

Погрешность этого приближения к $hy'(x_n)$ имеет порядок $O(h^{k+1})$. Тогда (8.10) примет вид

$$y_n^{(1)} = y_n^{(0)} + b_0(hf(x_n, y_n^{(0)}) - hy_n'^{(0)}). \quad (8.12)$$

Из (8.12) следует, что формула $y_n^{(v+1)} = y_n^{(v)} + b_0(hf(x_n, y_n^{(v)}) - hy_n'^{(v)})$ справедлива также для $v = 0$. При этом $hy_n'^{(0)}$ определяется с помощью (8.11).

Введем в рассмотрение векторы

$$Y_n = (y_n, hy_n', y_{n-1}, \dots, y_{n-k+1})^T$$

и

$$Y(x_n) = (y(x_n), hy'(x_n), y(x_{n-1}), \dots, y(x_{n-k+1}))^T.$$

Первая компонента y_n вектора Y_n аппроксимирует $y(x_n)$, вторая компонента hy_n' вектора Y_n аппроксимирует $hy'(x_n)$. Остальные компоненты y_{n-l} аппроксимируют значения точного решения $y(x_{n-l})$, $l = 1, 2, \dots, k-1$. Определим также вектор

$$Y_n^{(v)} = (y_n^{(v)}, hy_n'^{(v)}, y_{n-1}, \dots, y_{n-k+1})^T$$

и матрицу

$$D = \begin{bmatrix} A_1 & B_1 & A_2 & \dots & A_{k-1} & A_k \\ \gamma_1 & \delta_1 & \gamma_2 & \dots & \gamma_{k-1} & \gamma_k \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}.$$

Тогда предсказывающие формулы (8.7) и (8.11) для решения $y(x_n)$ и производной $hy'(x_n)$ могут быть представлены в виде следующего векторного соотношения:

$$Y_n^{(0)} = DY_{n-1}. \quad (8.13)$$

Погрешность первой и второй компоненты вектора $\bar{Y}_n^{(0)}$ имеет порядок $O(h^{k+1})$, а для остальных равна нулю, если $y_{n-i} = y(x_{n-i})$, $i = 1, \dots, k-1$, и $O(h^{k+2})$, если $y(x_{n-i}) - y_{n-i} = O(h^{k+2})$. Значит,

$$Y(x_n) - Y_n^{(0)} = O(h^{k+1}). \quad (8.14)$$

Из (8.9) следует, что

$$hy_n^{(v+1)} = hy_n^{(v)} + hf(x_n, y_n^{(v)}) - hy_n^{(v)}. \quad (8.15)$$

Введем вектор $c = (b_0, 1, 0, \dots, 0)^T$ и функцию невязки

$$F(Y_n^{(v)}) = hf(x_n, y_n^{(v)}) - hy_n^{(v)}. \quad (8.16)$$

Тогда исправляющие формулы (8.9) и (8.15) для решения $y(x_n)$ и производной $hy'(x_n)$ могут быть представлены в виде

$$Y_n^{(v+1)} = Y_n^{(v)} + cF(Y_n^{(v)}). \quad (8.17)$$

Погрешность первой компоненты вектора $Y_n^{(v+1)}$ имеет порядок $O(h^{k+1})$, погрешность второй компоненты – $O(h^{k+2})$, а для остальных компонент погрешность такая же, как для $Y_n^{(0)}$. Значит,

$$Y(x_n) - Y_n^{(v+1)} = O(h^{k+1}). \quad (8.18)$$

Итерационный процесс (8.7), (8.8) сходится к y_n , если h достаточно мало. Следовательно, будет сходиться итерационный процесс (8.13), (8.17):

$$Y_n^{(v)} \rightarrow Y_n$$

и

$$Y(x_n) - Y_n = O(h^{k+1}).$$

Определим линейное преобразование Q , которое переводит вектор $Y(x_n)$ в вектор $Z(x_n) + O(h^{k+1})$, где

$$Z(x_n) = \left(y(x_n), hy'(x_n), \frac{h^2 y''(x_n)}{2}, \dots, \frac{h^k y^{(k)}(x_n)}{k!} \right)^T,$$

т. е.

$$QY(x_n) = Z(x_n) + O(h^{k+1}),$$

или записывается покомпонентно,

$$\begin{pmatrix} y(x_n) \\ hy'(x_n) \\ y(x_{n-1}) \\ y(x_{n-2}) \\ \dots \\ y(x_{n-k+1}) \end{pmatrix} \xrightarrow{Q} \begin{pmatrix} y(x_n) \\ hy'(x_n) \\ \frac{h^2 y''(x_n)}{2} \\ \frac{h^2 y'''(x_n)}{6} \\ \dots \\ \frac{h^k y^{(k)}(x_n)}{k!} \end{pmatrix}.$$

Вектор $Z(x_n)$ называется *вектором Нордсика*.

Преобразование Q переводит вектор $Y(x_n)$ в вектор $Z(x_n)$

$$Z_n = QY_n = Q(Y(x_n) + O(h^{k+1})) = QY(x_n) + O(h^{k+1}) = Z(x_n) + O(h^{k+1}).$$

Матрицу Q такого преобразования можно найти следующим образом. Так как преобразование Q не меняет первые две компоненты вектора, то первые две строки матрицы Q имеют вид

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \end{pmatrix}$$

Чтобы найти l -ю строку матрицы Q

$$q_{l1}, q_{l2}, \dots, q_{l,k+1} \quad (l \geq 3),$$

надо разложить все слагаемые в левой части равенства

$$q_{l1}y(x_n) + q_{l2}hy'(x_n) + q_{l3}y(x_{n-1}) + \dots + q_{l,k+1}y(x_{n-k+1}) = \frac{h^{l-1}y^{(l-1)}(x_n)}{(l-1)!} + O(h^{k+1})$$

по степеням h , а затем приравнять коэффициенты при одинаковых степенях h справа и слева. Получим $k + 1$ уравнений с $k + 1$ неизвестными. Решая эту систему, найдем элементы l -й строки. Таким способом находятся строки третья, четвертая, ..., $(k+1)$ -я.

Если записать покомпонентно, то имеем

$$\begin{pmatrix} y(x_n) \\ hy'(x_n) \\ y(x_{n-1}) \\ y(x_{n-2}) \\ \dots \\ y(x_{n-k+1}) \end{pmatrix} \xrightarrow{Q} \begin{pmatrix} y(x_n) \\ hy'(x_n) \\ \frac{h^2 y''(x_n)}{2} \\ \frac{h^2 y'''(x_n)}{6} \\ \dots \\ \frac{h^k y^{(k)}(x_n)}{k!} \end{pmatrix} + O(h^{k+1}).$$

Из (8.18) следует, что

$$Z_n^{(v)} = QY_n^{(v)} = Q(Y(x_n) + O(h^{k+1})) = QY(x_n) + O(h^{k+1}) = Z(x_n) + O(h^{k+1}). \quad (8.19)$$

Применим преобразование Q к (8.13). Тогда

$$QY_n^{(v+1)} = QY_n^{(v)} + QcF(Y_n^{(v)}) = Z_n^{(v)} + IF(Y_n^{(v)}),$$

где $I = Q_c = (l_0, l_1, l_2, \dots, l_k)^T$. Так как функция невязки $F(Y_n^{(v)})$ зависит только от первых двух компонент вектора $Y_n^{(v)}$, а преобразование Q первые две компоненты не меняет, то эти компоненты вектора $Y_n^{(v)}$ совпадают с первыми двумя компонентами вектора $Z_n^{(v)}$. Поэтому

$$Z_n^{(v+1)} = Z_n^{(v)} + IF(Z_n^{(v)}).$$

Из сходимости $Y_n^{(v)}$ к Y_n следует сходимость $Z_n^{(v)}$ к Z_n . Таким образом, приходим к такому сходящемуся итерационному процессу:

$$Z_n^{(0)} = PZ_{n-1},$$

$$Z_n^{(v+1)} = Z_n^{(v)} + IF(Z_n^{(v)}), \quad v = 0, 1, \dots \quad (8.20)$$

Из (8.20) следует, что

$$Z_n^{(v+1)} = Z_n^{(0)} + I(F(Z_n^{(0)}) + F(Z_n^{(1)}) + \dots + F(Z_n^{(v)})). \quad (8.21)$$

Из сходимости (8.21) вытекает, что $Z_n^{(v+1)}$ сходится к

$$Z_n = Z_n^{(0)} + Iw, \quad (8.22)$$

где $w = \lim_{v \rightarrow \infty} \sum_{j=0}^v F(Z_n^j)$, причем, как следует из (8.21),

$$F(Z_n) = 0. \quad (8.23)$$

Подставляя (8.22) в (8.23), получаем уравнение

$$F(Z_n^{(0)} + Iw) = 0. \quad (8.24)$$

Решение уравнения (8.24) относительно w методом Ньютона приводит:

$$w^{v+1} = w^v - \left(\frac{\partial F(Z_n^{(0)} + Iw^{(v)})}{\partial Z} I \right)^{-1} F(Z_n^{(0)} + Iw^v).$$

Умножим обе части равенства на I :

$$Iw^{v+1} = Iw^v - I \left(\frac{\partial F(Z_n^{(0)} + Iw^{(v)})}{\partial Z} I \right)^{-1} F(Z_n^{(0)} + Iw^v).$$

Прибавим к обеим частям равенства $Z_n^{(0)}$ и обозначим $Z_n^{(v)} = Z_n^{(0)} + Iw^{(v)}$. Тогда имеем следующий итерационный процесс:

$$Z_n^{(v+1)} = Z_n^{(v)} - I \left(\frac{\partial F(Z_n^{(v)})}{\partial Z} I \right)^{-1} F(Z_n^{(v)}).$$

Начальное приближение для этого процесса определяется с помощью $Z_n^{(0)} = PZ_{n-1}$. Из (8.16) находим

$$\frac{\partial F}{\partial Z} = \left(-\frac{\partial F}{\partial Z}, \frac{\partial F}{\partial Z^2}, \dots, \frac{\partial F}{\partial Z^{k+1}} \right) = \left(h \frac{\partial f}{\partial y}, -1, 0, \dots, 0 \right).$$

Обозначим

$$W = \left(\frac{\partial F(Z_n^{(v)})}{\partial Z} I \right)^{-1} = \left(hl_0 \frac{\partial f(x_n, y_n^{(v)})}{\partial y} - l_1 \right)^{-1}.$$

Тогда

$$Z_n^{(v+1)} = Z_n^{(v)} - IWF(Z_n^{(v)}), \quad (8.25)$$

или

$$Z_n^{(v+1)} = Z_n^{(v)} - I \left(hl_0 \frac{\partial f(x_n, y_n^{(v)})}{\partial y} - l_1 \right)^{-1} (hf(x_n, y_n^{(v)}) - hy_n'^{(v)}).$$

Итерационный процесс (8.25) отличается от итерационного процесса (8.20) тем, что сложению вектора $IFcZ_n^{(v)}$ в (8.20) предшествует умножение F на матрицу W в (8.25) и только после этого производится коррекция значения $Z_n^{(v)}$.

Если правая часть дифференциального уравнения $f(x, y)$ линейна по y , то итерационный процесс (8.25) сойдется за одну итерацию.

Компоненты вектора $I = (l_0, l_1, \dots, l_k)^T$ для методов разного порядка аппроксимации приведены в табл. 8.1.

Таблица 8.1

Коэффициенты метода Гира $I = (l_0, l_1, \dots, l_k)^T$

Коэффициенты	Порядок метода				
	2	3	4	5	6
l_0	$\frac{2}{3}$	$\frac{6}{11}$	$\frac{12}{25}$	$\frac{60}{137}$	$\frac{20}{49}$
l_1	1	1	1	1	1
l_2	$\frac{1}{3}$	$\frac{6}{11}$	$\frac{7}{10}$	$\frac{225}{274}$	$\frac{58}{63}$
l_3		$\frac{1}{11}$	$\frac{1}{5}$	$\frac{85}{274}$	$\frac{5}{12}$
l_4			$\frac{1}{50}$	$\frac{15}{274}$	$\frac{25}{252}$
l_5				$\frac{1}{274}$	$\frac{1}{84}$
l_6					$\frac{1}{1764}$

Рассмотренный метод решения жестких систем на основе формул дифференцирования назад (метод Гира) позволяет получать эффективные алгоритмы расчетов за счет реализации процедур автоматической смены порядка и шага метода, по сравнению с другими методами решения жестких систем.

Лекция 9. Параметры оценки эффективности методов анализа во временной области

Контроль точности в алгоритмах анализа СУ. Оценка устойчивости методов интегрирования. Алгоритмы автоматического выбора шага. Выбор эффективных методов анализа переходных процессов СУ.

Контроль точности в алгоритмах анализа СУ

Основные характеристики методов интегрирования, от которых зависит их эффективность, – точность и устойчивость, а также выбор величины шага интегрирования [6; 10]. Выполнить все требования в рамках одного метода невозможно, поэтому разработан ряд базовых методов различной степени точности и экономичности, используемых в САПР в зависимости от решаемой задачи. При этом алгоритмическая реализация решения систем ОДУ отличается большим разнообразием, так как алгоритмы содержат значительное количество параметров и процедур, выбор и реализация которых не зависит от конкретного численного метода и может быть произвольной. Поэтому эффективность программной реализации методов анализа переходных процессов в САПР зависит не только от характеристик самих методов интегрирования, но и от особенностей их алгоритмической реализации. К особенностям алгоритмов относятся: способ прогноза в неявных методах; контроль ошибки решения; контроль потери устойчивости в явных методах; способ выбора очередного шага; ограничения на максимальный шаг интегрирования; способ начала решения в многошаговых методах.

Рассмотрим основные характеристики методов интегрирования и особенности их алгоритмической реализации.

Точность интегрирования можно оценить, проанализировав полную ошибку ε_k^{Pi} на каждом шаге интегрирования. Однако на практике точное решение системы ОДУ неизвестно, следовательно, оцениваются основные составляющие полной ошибки:

– ошибка аппроксимации (методическая ошибка) ε_k^{ai} – погрешность собственно метода интегрирования (из-за замены производных конечно-разностными выражениями в формуле интегрирования);

– ошибка вычислений ε_k^{Bi} – связана с ошибками округлений чисел в ЭВМ и заменой неарифметических операций и функций арифметическими. Ошибки ε_k^{Bi} изменяются сложным образом так, что их можно рассматривать в качестве случайных величин. Статистический анализ ε_k^{Bi} весьма сложен, но практика расчетов разных систем ОДУ показывает, что при работе с точностью 10^{-9} этой составляющей ошибки можно пренебречь;

– ошибка накоплений ε_k^{Hi} равна полной ошибке на предыдущем шаге ε_{k-1}^{Pi} , а так как точное решение неизвестно, то оценить эту ошибку невозможно, поэтому можно говорить только о характере поведения ε_k^{Hi} . Эта оценка связана с исследованием устойчивости численных методов интегрирования. Если метод интегрирования устойчив, то полную ошибку ε_{k-1}^{Pi} можно определить с достаточной степенью точности, определяя ε_{k-1}^{ai} на $k-1$ -м шаге интегрирования, так как ε_{k-1}^{Bi} будет намного меньше, чем ε_{k-1}^{ai} , а ε_{k-1}^{Hi} не будет возрастать в ходе решения;

– ошибка аппроксимации ε_k^{ai} равна разности между точным решением и числовым решением, полученным по формуле интегрирования (для одной точки). Для малых шагов интегрирования точное решение можно всегда определить, разлагая $V_{i(t_k)}$ в ряд Тейлора в точке t_k , при этом формула интегрирования будет аппроксимировать только первые члены ряда Тейлора. Порядком точности (порядком аппроксимации) s метода интегрирования называется порядок производной в том члене ряда Тейлора, до которого формула интегрирования аппроксимирует ряд при условии $h_k \rightarrow 0$. Ошибку ε_k^{ai} можно определить путем оценки суммы оставшихся членов ряда Тейлора.

Для линейных многошаговых методов интегрирования порядок точности равен числу предыдущих точек, используемых в формуле интегрирования.

Для явных одношаговых методов интегрирования используются различные способы апостериорной оценки точности (оценки ε_k^{ai}), например, по правилу Рунге; использование комбинации формул разных порядков точности; с помощью нелинейного контрольного члена.

Наиболее универсальным является правило Рунге. Обозначим через $V_{k+1}^i(h/2)$ значение фазовой переменной, полученное в точке t_{k+1} повторным расчетом данным методом интегрирования от точки t_k за два половинных шага $h_k/2$. Тогда оценка точности

$$\varepsilon_k^{ai} = c \left| \frac{V_{k+1}^i(h/2) - V_{k+1}^i(h)}{2^s - 1} \right|,$$

где c – некоторая константа. Такой способ оценки точности требует существенных дополнительных затрат машинного времени, поэтому на практике для анализа переходных процессов разрабатывают более экономичные способы без проведения повторных расчетов на каждом шаге интегрирования.

В неявных методах для оценки точности можно, например, использовать процесс решения ОДУ относительно V_{k+1} по разности между спрогнозированным и окончательным решением $\varepsilon_{k+1} = V_{k+1}^{(0)} - V_{k+1}$. Прогноз обычно осуществляется на основе предшествующих точке V_{k+1} значений фазовой переменной или оценок ее производных. Для одно- и двухшаговых методов прогноз выполняют по формулам: $V_{k+1}^{(0)} = V_k$ – постоянный прогноз, $V_{k+1}^{(0)} = V_k + (V_{k+1} - V_{k-1})/h$ – линейный прогноз.

Использование для повышения точности прогноза точек V_{k-2}, V_{k-3} и т. д. в этих методах нецелесообразно, так как требует дополнительной памяти. В многошаговых методах эти точки необходимы для реализации метода, поэтому их используют также и для прогноза, который обычно строится в виде итерполяционного полинома. Однако увеличение степени прогнозирующего полинома не всегда ведет к повышению точности прогноза.

Оценка устойчивости методов интегрирования

Поведение ошибок определяет такие характеристики численных методов решения системы ОДУ, как сходимость и устойчивость.

Сходимость – неограниченное сближение кривых точного и приближенного решений при уменьшении шага h , т. е. $\varepsilon_k^{Pi} \rightarrow 0$ при $h \rightarrow 0, k \rightarrow \infty$. Уменьшение ε_k^{Pi} требует выполнения условия $\varepsilon_k^{ai} \rightarrow 0$ при $h \rightarrow 0$ и $\varepsilon_k^{Hi} \rightarrow 0$.

Если при $k \rightarrow \infty$ полная ошибка $\varepsilon_k^{Pi} \rightarrow \infty$, то метод называется неустойчивым. Если ε_k^{Pi} растёт, но не быстрее, чем само решение V_{ki} , то метод – относительно устойчивый (численно устойчивый).

Устойчивость численного интегрирования связана не только с методом интегрирования, но и с характером решаемой задачи, в частности с обусловленностью математической модели технического объекта – системы ОДУ.

Для количественного определения устойчивости методов интегрирования используется линейная система ОДУ

$$\dot{U} = A \cdot U, U_0 \neq 0, \quad (9.1)$$

где $U \subset V$, U – вектор переменных состояния размерностью n ; V – вектор фазовых переменных; A – постоянная действительная матрица размером $(n \times n)$.

Если метод интегрирования будет неустойчив для системы (9.1), то он, как правило, будет неустойчив и для общей системы уравнений ММ технического объекта

$$\frac{dV}{dt} = F(V, t). \quad (9.2)$$

В пределах шага интегрирования h_k матрицу Якоби Y_k для вектор-функции $F(V)$ системы (9.2) можно считать квазипостоянной, т. е. $Y_k \cong A$, следовательно, любую нелинейную систему ОДУ можно считать квазилинейной. Поэтому результаты исследования устойчивости, полученные для (9.1), будут хорошо сочетаться и с большинством нелинейных систем ОДУ.

Используя известное в теории матриц преобразование подобия

$$A = T \text{diag}(\lambda_i) T^{-1},$$

где T – некоторая преобразующая матрица; $\text{diag}(\lambda_i)$ – диагональная матрица с элементами λ_i на диагонали; λ_i – собственные значения матрицы A ; $i = 1, 2, \dots, n$, произведем замену переменных $U = TY$. Система (9.2) преобразуется в систему несвязанных линейных уравнений вида $\dot{y}_i = \lambda_i y_i, i = 1, 2, \dots, n$; где λ_i – в общем случае комплексная величина. Тогда анализ устойчивости и сравнение методов интегрирования по точности и устойчивости выполняется с помощью модельного уравнения первого порядка

$$\dot{y} = \lambda \cdot y \quad y_0 = I, \quad (9.3)$$

где $\lambda_i = \text{Re}(\lambda_i) + j \text{Im}(\lambda_i)$.

Аналитическое решение уравнения (9.3): $y(t) = e^{\lambda \cdot t}$. В зависимости от положения λ_i на комплексной плоскости аналитическое решение может быть сходящимся, т. е. система ОДУ будет устойчивой; расходящимся – система ОДУ неустойчива; колебательным – система ОДУ на границе устойчивости. Таким образом, собственные значения матрицы A или Y_k характеризуют переходные процессы моделируемого объекта (устойчивость, наличие колебаний и т. п.), в то же время они определяют и математические свойства самой модели. В частности, система ОДУ будет плохо обусловленной (жесткой), если выполняется условие

$$z = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|} \gg 1,$$

где z – число обусловленности матрицы A . Обычно математическую модель считают плохо обусловленной (а систему уравнений жесткой), если $z > 10^5$.

Для оценки точности и устойчивости метода интегрирования модельное уравнение решается с постоянным шагом h и сравнивается с аналитическим. Оценка точности производится по результатам первого шага, а оценка устойчивости – при $t \rightarrow \infty$.

Аналитическое решение модельного уравнения на первом шаге можно представить в виде ряда:

$$e^{\lambda \cdot h} = 1 + \lambda \cdot h + \frac{(\lambda \cdot h)^2}{2!} + \frac{(\lambda \cdot h)^3}{3!} + \dots$$

Формулу интегрирования (8.1) на первом шаге можно разложить в аналогичный ряд (для многошаговых формул первый шаг начинается с точки $(ph, e^{\lambda ph})$), который должен совпадать с (9.3) для первых m членов. Формула интегрирования абсолютно устойчива для заданного $h\lambda$, если численное решение (9.3) при $t \rightarrow \infty$ равно нулю.

Любое комплексное число $h\lambda$ можно изобразить на комплексной плоскости Oxy в виде точки с координатами $Re(h\lambda)$ по оси Ox и $Im(h\lambda)$ по оси Oy . Областью абсолютной устойчивости метода интегрирования называется область R комплексной плоскости $(Re(h\lambda), Im(h\lambda))$, в которой формула интегрирования абсолютно устойчива для всех $h\lambda \in R$ (рис. 9.1).

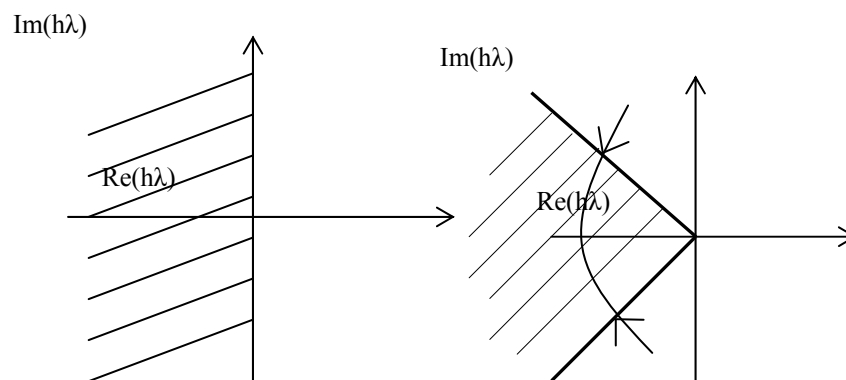


Рис. 9.1

Большинству практических задач соответствуют системы ОДУ, для которых $Re(\lambda_i) < 0$. В этом случае идеальным будет такой метод интегрирования, у которого область абсолютной устойчивости включает всю левую полуплоскость $Re(h\lambda) < 0$. Такие методы называют A -устойчивыми методами. Если область лежит внутри угла $s(\alpha, -\alpha)$ ($0 < \alpha < \frac{\pi}{2}$), то метод называется $A(\alpha)$ – устойчивым. Для каждого метода интегрирования на комплексной плоскости $h\lambda$ можно получить область относительной устойчивости (рис. 9.2).

На практике возможны случаи, когда нелинейная система ОДУ становится локально неустойчивой, т. е. появляются λ_i , для которых $Re(\lambda_i \cdot h) > 0$. В этом случае метод интегрирования должен быть также неустойчив.

Помимо $A(\alpha)$ -устойчивости для точности приближения к аналитическому решению системы $\dot{U} = A \cdot U$ используют понятие жесткой устойчивости, при котором метод должен быть A -устойчив в области R_1 и точным в области R_2 .

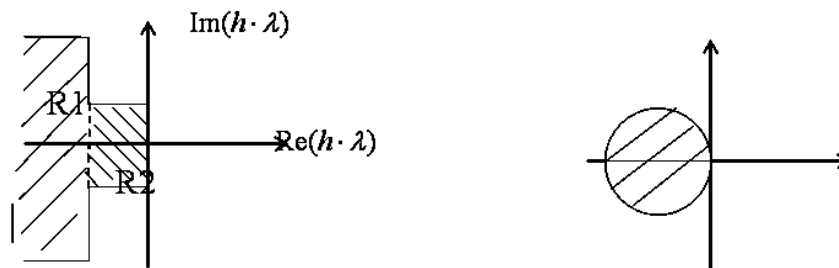


Рис. 9.2

Дальквист доказал, что:

- а) никакой явный многошаговый не является A -устойчивым;
- б) явные методы типа Рунге – Кутта также не A -устойчивы;
- в) неявные многошаговые методы порядка выше второго, не A -устойчивы.

Все это справедливо только для линейных ОДУ с постоянным шагом.

Теория устойчивости для нелинейных ОДУ с переменным шагом разработана для частных случаев.

Контроль потери устойчивости решения ОДУ осуществляется методами, аналогичными или близкими методам контроля ошибки. Очевидно, поскольку рост локальной ошибки предшествует потере устойчивости, то контроль ошибки предотвращает потерю устойчивости. В неявных A -устойчивых методах под потерей устойчивости понимается расходимость итераций при расчете V_{k+1} . Формальным признаком потери устойчивости служит несходимость итерационного процесса за заданное число итераций с заданной ошибкой ϵ . В явных методах потеря устойчивости означает превышение допустимого шага. Формальным признаком потери устойчивости обычно служит появление периодических осцилляций значений V_k , V_{k+1} и т. д. В обоих случаях для сохранения устойчивости нужно уменьшать шаг.

Следовательно, на основе результатов контроля точности и устойчивости осуществляется выбор величины шага интегрирования. Если для метода найдены ϵ_k^{ai} и область устойчивости, то эффективность метода интегрирования будет во многом определяться стратегией *выбора величины шага интегрирования*, поскольку выполнять интегрирование с постоянным шагом нецелесообразно.

При увеличении шага увеличивается погрешность аппроксимации ε_k^{ai} , кроме того, метод может выйти за границу устойчивости. При уменьшении шага увеличивается время расчёта. Оптимальную величину шага интегрирования нельзя определить априорно, при этом имеется хороший способ для выбора величины шага, используемый в любой практической программе анализа динамических режимов. Шаг интегрирования выбирается так, чтобы суммарная погрешность была распределена примерно одинаково по шагам. Исходя из этого, шаг интегрирования выбирают в зависимости от заданной погрешности аппроксимации ε_k^{ai} , допустимой на один шаг интегрирования (при этом нужно гарантировать устойчивость вычислений). Для ограниченно устойчивых методов интегрирования шаг выбирается исходя из условия сохранения устойчивости (при этом нужно гарантировать заданную точность интегрирования). Следовательно, минимизация затрат машинного времени при соблюдении точностных ограничений возможна только в условиях интегрирования с переменным шагом.

Большинство алгоритмов автоматического выбора шага основано на контроле полной ошибки интегрирования, который может быть осуществлён методом двойного счета, т. е. сравнением результатов V_{k+1}^h и $V_{k+1}^{h/2}$, полученных одним полным шагом h и двумя половинными $h/2$. Если $\varepsilon_{k+1}^{П1} = \left| V_{k+1}^h - V_{k+1}^{h/2} \right| < \varepsilon_{доп}^a$, то требования точности удовлетворены. Недостатком является увеличение времени расчёта в два раза. Используется лишь в явных методах.

Для неявных методов для контроля ошибки используют процесс решения ОДУ относительно V_{k+1} по разности $\varepsilon_{k+1}^{П} = V_{k+1}^{(0)} - V_{k+1}$ между спрогнозированным и окончательным решением.

По количеству итераций p , необходимо для получения $V_{k+1}(p)$ такого, что

$$\varepsilon_{k+1}^{П} = V_{k+1}^{(p)} - V_{k+1}^{(p-1)} < \varepsilon_{доп}^a.$$

В неявных методах чаще всего используют следующий алгоритм изменения шага. Назначаются оценки ε_{\min} и ε_{\max} , тогда

$$h_{\text{нов}} = h_{\text{ст}} \text{ при } \varepsilon_{\min} < \varepsilon < \varepsilon_{\max},$$

$$h_{\text{нов}} > h_{\text{ст}} \text{ при } \varepsilon < \varepsilon_{\min},$$

$$h_{\text{нов}} > h_{\text{ст}} \text{ при } \varepsilon > \varepsilon_{\max}.$$

Конкретное соотношение между $h_{\text{нов}}$ и $h_{\text{ст}}$ зависит от способа контроля полной ошибки. Если контроль выполняется вторым способом – прямым, где ошибка шага характеризуется непосредственно, то новый шаг вычисляется по формуле

$$h_{\text{нов}} = n \sqrt{\frac{\varepsilon_{\text{доп}}}{\varepsilon_{n,k+1}}} \cdot h_{\text{ст}},$$

$$\varepsilon_{\text{доп}} = \varepsilon_{\text{min}}, \text{ если } \varepsilon_{k+1}^{\text{П}} < \varepsilon_{\text{min}},$$

$$\varepsilon_{\text{доп}} = \varepsilon_{\text{max}}, \text{ если } \varepsilon_{k+1}^{\text{П}} > \varepsilon_{\text{max}}.$$

Если контроль ошибки выполняется косвенным способом, например, по числу итераций, то используется правило

$$h_{\text{нов}} = h_{\text{ст}} \text{ при } p_{\text{min}} \leq p \leq p_{\text{max}},$$

$$h_{\text{нов}} = c \cdot h_{\text{ст}} \text{ при } \lim_{t \rightarrow \infty} q_i(t) = 0,$$

$$h_{\text{нов}} = \frac{h_{\text{ст}}}{c} \text{ при } p > p_{\text{max}}.$$

c – коэффициент изменения шага, обычно $c = 2$.

Значение p_{min} , p_{max} зависят от алгоритма расчета V_{k+1} . Для алгоритма Ньютона $p_{\text{min}} = 2 \div 3$; $p_{\text{max}} = 5 \div 6$.

Ограничения на максимальный шаг интегрирования может задаваться, исходя из особенностей решения конкретной задачи, так, чтобы не пропустить быстрых изменений решения $V(t)$. В неявных методах вводится ограничение на минимальный шаг, чтобы в случае невозможности обеспечить заданную локальную точность ограничить процесс уменьшения шага и при $h < h_{\text{min}}$ выдать соответствующую информацию.

Алгоритмы автоматического выбора шага

Рассмотрим алгоритмы автоматического выбора шага для одношаговых методов [5; 12].

1. Алгоритм выбора с помощью удвоения и деления шага пополам

Пусть ρ_{n+1} – оценка локальной погрешности метода на шаге h , допущенной при вычислении приближенного значения решения y_{n+1}^h в точке $x_n + h$. Если оценка превосходит некоторую наперед заданную границу ε

$$|\rho_{n+1}| > \varepsilon,$$

то считается, что значение y_{n+1}^h решения не удовлетворяет предписанной точности и шаг h объявляется неприемлемым. Полученная точка $x_n + h$ и значение y_{n+1}^h исключаются из рассмотрения. Выбирается новое значение шага

$$h^{(1)} = h/2$$

и вновь по той же формуле с шагом $h^{(1)}$ вычисляется новое значение решения $y_{n+1}^{h^{(1)}}$ в новой точке $x_n + h^{(1)}$.

Пусть $\rho_{n+1}^{(1)}$ – оценка локальной погрешности метода на данном шаге $h^{(1)}$. Если оценка опять превосходит заданную границу ε

$$|\rho_{n+1}^{(1)}| > \varepsilon,$$

то точка $x_n + h$ и значение $y_{n+1}^{h^{(1)}}$ опять исключаются из рассмотрения, шаг снова делится пополам:

$$h^{(2)} = h^{(1)} / 2$$

и вычисления повторяются. Так происходит до тех пор, пока при какой-то величине шага (обозначим ее через h_n) оценка локальной погрешности не станет меньше ε

$$|\rho_{n+1}| \leq \varepsilon.$$

После этого считается, что решение дифференциального уравнения продолжено до точки $x_{n+1} = x_n + h_n$. Дальнейшее интегрирование уравнения производится из точки x_{n+1} с шагом h_{n+1} , который выбирается описанным ниже способом.

Если оценка локальной погрешности на шаге $h_n = x_{n+1} - x_n$ удовлетворяет неравенству

$$|\rho_{n+1}| < \varepsilon / K,$$

где K – некоторая константа, то считается, что достигнута точность, превышающая заданную, и шаг интегрирования удваивается:

$$h_{n+1} = 2h_n.$$

Если выполняется неравенство

$$\varepsilon / K \leq |\rho_{n+1}| \leq \varepsilon,$$

то считается, что полученное в точке x_{n+1} решение удовлетворяет заданной точности и шаг интегрирования остается без изменения $h_{n+1} = h_n$.

Таким образом, на тех участках изменения независимой переменной, где достигается высокая точность приближенного решения, шаг интегрирования возрастает, а там, где точность не достигается, шаг интегрирования сокращается до необходимых для ее достижения значений. Тем самым обеспечивается выбор величины шага в зависимости от характера поведения решения дифференциального уравнения. Константа K обычно полагается равной 2ν , где ν – порядок используемой оценки локальной погрешности метода.

Иногда для того, чтобы сократить число неприемлемых шагов, в изложенный здесь алгоритм выбора шага вносится изменение, которое заключается в следующем. Если при продолжении решения из точки x_n в точку $x_{n+1} = x_n + h_n$ шаг интегрирования сокращался хотя бы один раз, то при выборе следующего значения шага h_{n+1} удвоения предыдущего шага h_n не происходит.

2. Алгоритм выбора максимальной для заданной точности длины шага

Так как оценка ρ_{n+1} локальной погрешности метода равна с точностью до членов более высокого порядка малости главному члену локальной погрешности метода, то

$$\rho_{n+1} \cong \psi(x_n, y_n) h^{s+1}. \quad (9.4)$$

Соотношение (9.4) справедливо для всех оценок. Если оценка ρ_{n+1} погрешности превосходит заданную границу ε .

$$|\rho_{n+1}| > \varepsilon,$$

то считается, что на данном шаге h метод не достигает требуемой точности и вычисленное значение y_{n+1} решения вместе с точкой x_{n+1} исключается из рассмотрения. В этом случае выбирается новый размер шага, но не последовательным делением пополам, как в описанном выше способе, а с помощью соотношения

$$h_\varepsilon = \alpha h, \quad (9.5)$$

где α находится из условия выполнения равенства

$$|\psi(x_n, y_n) h_\varepsilon^{s+1}| = \varepsilon. \quad (9.6)$$

Из (9.4), (9.5) получаем, что $\alpha^{s+1} = \frac{\varepsilon}{|\rho_{n+1}|}$ и $\alpha = \sqrt[s+1]{\frac{\varepsilon}{|\rho_{n+1}|}}$. Здесь $\alpha < 1$

и новое значение шага

$$h_\varepsilon = \sqrt[s+1]{\frac{\varepsilon}{|\rho_{n+1}|}} \cdot h$$

меньше предыдущего. Далее по формуле Рунге – Кутты из точки x_n выполняется один шаг h_ε и вычисляется приближение $y_{n+1}^{h_\varepsilon}$ к решению дифференциального уравнения в точке $x_n + h_\varepsilon$.

Если первоначальная оценка ρ_{n+1} локальной погрешности метода не превосходит заданную границу ε :

$$|\rho_{n+1}| \leq \varepsilon,$$

то считается, что полученное приближение y_{n+1} к решению удовлетворяет требуемой точности и значение x_{n+h} независимой переменной принимается в качестве следующего узла интервала интегрирования. Дальнейшее интегрирование уравнения осуществляется из точки x_{n+1} с шагом h_ε . Теперь $\alpha \geq 1$ и $h_\varepsilon \geq h$.

Преимущество данного алгоритма выбора шага заключается в большей гибкости по сравнению с описанным в предыдущем разделе способом. В нем при достижении требуемой точности абсолютная величина шага интегрирования либо увеличивается в два раза, либо не изменяется в зависимости от того, выполняется или нет неравенство $|\rho_{n+1}| < \varepsilon / K$. Если это неравенство не выполняется из-за незначительного превышения оценки $|\rho_{n+1}|$ над величиной ε / K , то шаг интегрирования не увеличивается и остается прежним. В алгоритме, основанном на использовании формулы (9.5), имеется возможность увеличения шага в любое число α раз даже тогда, когда данное число меньше двух. Это приводит к более сглаженному изменению шага интегрирования и, как следствие, к сокращению общего количества шагов и снижению вычислительных затрат.

В действительности берется несколько меньшее, чем определяемое значение α , например,

$$\alpha^* = 0,9 \alpha = 0,9 \sqrt[s+1]{\frac{\varepsilon}{|\rho_{n+1}|}}.$$

И, соответственно, меньшее по сравнению с (9.6) значение шага интегрирования

$$h_{\varepsilon}^* = \alpha^* h. \quad (9.7)$$

Это делается для того, чтобы избежать тех шагов, для которых не достигается требуемая точность.

Как и для одношаговых методов, вопрос о *выборе шага интегрирования для многошаговых методов* имеет существенное значение, поскольку от него зависит не только точность вычисления решения, но и время, необходимое для решения задачи. Имея оценки для локальной погрешности, можно организовать для многошаговых методов автоматический выбор шага интегрирования, руководствуясь теми же соображениями, что и для одношаговых методов. Однако использование конечно-разностных схем вносит свою особенность в этот процесс.

Применение переменного шага интегрирования позволяет учитывать характер поведения решения и уменьшить общее число шагов, сохранив при этом требуемую точность приближенного решения. Тем самым могут быть снижены объем работы и машинное время и замедлен рост вычислительной погрешности.

Оценка погрешности и автоматический выбор шага и порядка для метода Гира решения жестких систем. Первые две компоненты вектора Y_n равны первым двум компонентам вектора Z_n , так как преобразование Q не меняет первые две компоненты. Следовательно, для оценки погрешности первой компоненты y_n решения Z_n уравнения можно воспользоваться оценкой локальной погрешности формулы

$$\rho_n = C_{k+1} h^{k+1} y^{(k+1)}(x_n) + O(h^{k+2}), \quad (9.8)$$

где $C_{k+1} = -\frac{\beta_k}{k+1}$.

Неизвестной величиной в (9.8) является $y^{(k+1)}(x_n)$. Если заменить производную $hy^{(k+1)}(x_n)$ разностью назад $\nabla y^{(k)}(x_n)$, то при этом допустим погрешность порядка $O(h^2)$. Если заменить $h^{k+1}y^{(k+1)}(x_n)$ на разность $\nabla(h^k y^{(k)}(x_n))$, то допущена ошибка порядка $O(h^{k+2})$:

$$h^{k+1}y^{(k+1)}(x_n) = \nabla(h^k y^{(k)}(x_n)) + O(h^{k+2}).$$

Конечную разность $\nabla(h^k y^{(k)}(x_n))$ можно получить, если из последней компоненты $Z_{n,k+1}$ вектора Z_n вычесть последнюю компоненту $Z_{n-1,k+1}$ вектора Z_{n-1} и полученную разность умножить на $k!$. Следовательно,

$$h^{k+1}y^{(k+1)}(x_n) \cong \nabla Z_{n,k+1} \cdot k!.$$

В результате имеем

$$\rho_n \cong C_{k+1} \nabla Z_{n,k+1} \cdot k! \quad (9.9)$$

или, расписывая покомпонентно,

$$\rho_n^i \cong C_{k+1} \nabla Z_{n,k+1}^i \cdot k!, \quad i = 1, 2, \dots, M.$$

Последняя компонента вектора Z_{n-1} совпадает с последней компонентой вектора $Z_{n-1}^{(0)}$. ρ_n^i – абсолютная погрешность y_n^i . Введены также другие характеристики точности:

относительная погрешность

$$\rho_n^i / |y_n^i|, \quad y_n^i \neq 0,$$

стандартная погрешность

$$\rho_n^i / \max_{0 \leq j \leq n} |y_j^i|,$$

мера погрешности

$$\mu = \begin{cases} \rho_n^i / |y_n^i|, & \text{если } |y_n^i| > p_i \\ \rho_n^i, & \text{если } |y_n^i| \leq p_i. \end{cases}$$

Все перечисленные характеристики точности можно записывать в общем виде

$$\rho_n^i / P_n^i,$$

где

$$P_n^i = \begin{cases} 1 & \text{для абсолютной погрешности} \\ |y_n^i| & \text{для относительной погрешности} \\ \max_{0 \leq j \leq n} |y_j^i| & \text{для стандартной погрешности} \\ \text{если } |y_n^i| \leq p_i, \text{ то } 1, \text{ иначе } |y_n^i| & \text{для меры погрешности.} \end{cases}$$

Контроль точности ведется по норме

$$\sum_{i=1}^M \left(\frac{C_{k+1} \nabla Z_{n,k+1}^i \cdot k!}{P_n^i} \right)^2 = (C_{k+1} \cdot k!)^2 \sum_{i=1}^M \left(\frac{\nabla Z_{n,k+1}^i \cdot k!}{P_n^i} \right)^2. \quad (9.10)$$

Тогда

$$E = \left(\frac{\varepsilon}{C_{k+1} k!} \right)^2,$$

где ε – заданная точность вычисления приближенного решения, и

$$V = \sum_{i=1}^M \left(\frac{\nabla Z_{n,k+1}^i}{P_n^i} \right)^2.$$

Тогда, если

$$V > E,$$

то считается, что на данном шаге метод не достигает требуемой точности и вычисленное значение вместе с точкой исключаются из рассмотрения. Новый размер шага:

$$h^* = \xi h,$$

где величина ξ выбирается так, чтобы на шаге h^* достигалась требуемая точность. Это приводит к следующему значению для ξ :

$$\xi = \left(\frac{E}{V} \right)^{\frac{1}{2(k+1)}}. \quad (9.11)$$

Вместо (9.11) можно взять несколько меньшее значение

$$\xi = \frac{1}{1.2} \left(\frac{E}{V} \right)^{\frac{1}{2(k+1)}}.$$

Теперь вычисление можно повторить по формулам (8.24), исходя из точки x_{n-1} . Для этого необходимо пересчитать вектор Z_{n-1} . Новые значения компонент этого вектора Z_{n-1}^* вычисляются по простым формулам:

$$Z_{n-1,j}^* = \xi^{j-1} Z_{n-1,j}, \quad j = 1, \dots, k+1.$$

Гир вместе с автоматическим выбором шага решает вопрос об автоматическом выборе порядка точности метода. Решение в точке x_n было получено не методом порядка k , а методом порядка на единицу меньше $k-1$. Тогда локальная погрешность решения равна

$$\rho_n \cong C_k h^k y^{(k)}(x_n) + O(h^{k+1}),$$

или

$$\rho_n \cong C_k \nabla Z_{n,k+1}^i \cdot k!.$$

Норма для погрешности запишется в следующем виде:

$$\sum_{i=1}^M \left(\frac{C_k \nabla Z_{n,k+1}^i \cdot k!}{P_n^i} \right)^2 = (C_k \cdot k!)^2 \sum_{i=1}^M \left(\frac{\nabla Z_{n,k+1}^i \cdot k!}{P_n^i} \right)^2.$$

Обозначим

$$\check{E} = \left(\frac{\varepsilon}{C_{k+1} k!} \right)^2, \quad \check{V} = \sum_{i=1}^M \left(\frac{\nabla Z_{n,k+1}^i}{P_n^i} \right)^2.$$

Если провести все рассуждения аналогично тому, как это было сделано для метода k -го порядка, то получим аналогичное выражение для константы изменения шага в методе порядка $k-1$:

$$\check{\xi} = \left(\frac{\check{E}}{\check{V}} \right)^{\frac{1}{2k}}. \quad (9.12)$$

Вместо (9.12) можно взять несколько меньшее значение

$$\xi = \frac{1}{1.3} \left(\frac{\check{E}}{\check{V}} \right)^{\frac{1}{2k}}.$$

Теперь рассмотрим, как следовало бы изменить шаг интегрирования, если бы решение в точке было получено методом порядка на единицу больше. В этом случае погрешность решения была бы равна

$$\rho_n = C_{k+2} h^{k+2} y^{(k+2)}(x_n) + O(h^{k+3}).$$

Выразим $h^{k+2} y^{(k+2)}(x_n)$ через разность

$$h^{k+2} y^{(k+2)}(x_n) = \nabla^2 (h^k y^{(k)}(x_n)) + O(h^{k+3}).$$

Далее

$$\nabla^2 (h^k y^{(k)}(x_n)) \cong \nabla^2 (Z_{n,k+1} \cdot k!) = \nabla (\nabla Z_{n,k+1} \cdot k!) = k! \nabla (\nabla Z_{n,k+1}).$$

В результате имеем

$$\rho_n \cong C_{k+2} \nabla(\nabla Z_{n,k+1}) k!. \quad (9.13)$$

Норма для погрешности запишется в виде

$$\sum_{i=1}^M \left(\frac{C_{n+2} \nabla Z_{n,k+1}^i \cdot k!}{P_n^i} \right)^2 = (C_{k+2} \cdot k!)^2 \sum_{i=1}^M \left(\frac{\nabla Z_{n,k+1}^i \cdot k!}{P_n^i} \right)^2.$$

Обозначим

$$\widehat{E} = \left(\frac{\varepsilon}{C_{k+1} k!} \right)^2, \quad \widehat{V} = \sum_{i=1}^M \left(\frac{\nabla Z_{n,k+1}^i}{P_n^i} \right)^2.$$

Аналогично предыдущему приходим y_k значению константы изменения шага интегрирования в методе порядка $k+1$

$$\widehat{\xi} = \left(\frac{\widehat{E}}{\widehat{V}} \right)^{\frac{1}{2(k+2)}}. \quad (9.14)$$

Вместо (9.14) можно взять несколько меньшее значение

$$\widehat{\xi} = \frac{1}{1.4} \left(\frac{\widehat{E}}{\widehat{V}} \right)^{\frac{1}{2(k+2)}}. \quad (9.15)$$

Выбором дополнительных множителей в (9.15), (9.12) отдается предпочтение методу порядка k или $k-1$.

Для эффективного использования оценки (9.15) следует при переходе от точки x_{m-1} к точке x_m сохранять значение $\nabla Z_{m-1,k+1}$, которое используется в (9.13) и V .

После вычисления $\xi, \xi^{\checkmark}, \widehat{\xi}$ выбирается тот метод, для которого соответствующая константа изменения шага максимальна.

Преимущество метода Гира по сравнению с другими многошаговыми методами численного интегрирования состоит в легкости изменения шага интегрирования.

Описанная процедура выбора шага и порядка в действительности несколько видоизменяется. Во-первых, если новая предполагаемая длина шага h^* интегрирования увеличивается менее чем в 1,1 раза по сравнению со старым значением h , то шаг не изменяется. Во-вторых, изменение шага и по-

рядка не допускается в течение $k + 1$ шагов после последнего изменения, если только используемое значение шага обеспечивает заданную точность. Причина этого заключается в том, что более частые частые изменения шага интегрирования могут вызвать дополнительный рост погрешности. В-третьих, после $k + 1$ шагов увеличение шага невозможно, то дальнейшая проверка

Обобщенно, выбор шага интегрирования включает два этапа:

- 1) прогноз величины шага по предыдущим вычислениям;
- 2) выполнение шага и подготовка данных для прогноза следующего шага либо отмена шага и его корректировка, если превышена ε_k^{ai} .

Выбор эффективных методов анализа переходных процессов СУ

Каждый метод интегрирования имеет свои достоинства и недостатки и не может справиться со всем многообразием задач анализа. Разработчик САПР, принимая решение о включении в состав проектирующей подсистемы того или иного метода интегрирования, руководствуется прежде всего функциональным назначением системы проектирования и характеристиками методов. Предпочтение должно отдаваться методам, обеспечивающим максимальную скорость решения при достаточной точности и безусловном выполнении требований устойчивости при анализе объектов заданного класса. Опытный пользователь САПР, решая свою конкретную задачу, выбирает подходящий метод интегрирования из библиотеки методов. Такая возможность пользователю предоставлена при моделировании в подсистеме Simulink системы Matlab. В САПР, базирующихся на формате Spice, таких как MicroCAP, OrCAD и др., пользователь лишен возможности выбора метода и работает с единственным встроенным методом интегрирования. Следовательно, при создании САПР разработчик должен включать в состав системы такие методы интегрирования, которые позволят инженеру-проектировщику получить интересующие его результаты с максимальной эффективностью.

Выбрать метод анализа переходных процессов для заданного класса объектов можно, оценив его качество по таким количественным характеристикам, как область устойчивости и точность при приемлемых вычислительных затратах. Следует отметить, что с повышением порядка метода, т. е. точности, его устойчивость по сравнению с методом того же класса, но более низкого порядка ухудшается.

Существует три группы задач, которые требуют использования различных методов интегрирования.

- 1) Расчет систем НАУ методом установления: внешние воздействия отсутствуют; $T_{\text{кон}}$ выбирается из условия завершения переходных процессов;

требования по точности не предъявляются; должен быть A -устойчивым методом. Например, неявный метод Эйлера первого порядка

$$h_k = h_{k-1} \text{ при } 2 < L < 6,$$

$$h_k = 2 \cdot h_{k-1} \text{ при } L < 2,$$

$$h_k = h_{k-1}^{1/2} \text{ при } L > 6,$$

где L – количество итераций.

2) Расчет систем ОДУ с преобразованием апериодических процессов $|\operatorname{Re} \lambda_i| \gg |\operatorname{Im} \lambda_i|$. Метод должен быть $A(\alpha)$ -устойчив; к точности предъявляются невысокие требования, так как ММ технического объекта имеют значительно большую погрешность; используют методы ФДН.

3) Система ОДУ с преобразованием осциллирующих процессов $|\operatorname{Im} \lambda_i| \gg |\operatorname{Re} \lambda_i|$ при наличии гармонических воздействий. Задачи этой группы наиболее сложны, так как в ряде случаев система ОДУ может быть локально неустойчивой. Основное требование к методам интегрирования $A(\pi/2)$ – устойчивость, следовательно, используют метод трапеций, т. е. одношаговые неявные методы.

Особая группа – задачи с «жесткими» системами уравнений (чаще всего это системы дифференциальных уравнений высокого порядка, описывающие динамические системы). Для описания процесса нужно использовать функции двух типов: на малых отрезках – функции с быстро меняющимися параметрами (большие производные) и на остальной части – функции с малыми производными (медленные движения). Часто в системе ОДУ не удается явно выделить члены с малыми параметрами, однако решение также имеет участки быстрых и медленных движений.

При больших значениях времени t основное значение для решения системы ОДУ имеет компонента, соответствующая собственному значению λ_n . Численное решение обычными методами затруднительно. Поэтому используют специально разработанные методы решения жестких систем: методы Гира, экспоненциальный неявный Рунге – Кутта, системные методы Ракитского.

Явные методы интегрирования целесообразно использовать при решении систем ОДУ в форме Коши. Достоинства явных методов: малый объем вычислений на одном шаге; уменьшение затрат памяти.

В отношении устойчивости методов можно отметить, что с увеличением порядка $A(\alpha)$ -устойчивого метода область его устойчивости уменьшается. Например, для метода типа ФДН для методов первого и второго порядков

$\alpha = \frac{\pi}{2}$; при $p = 6$ $\alpha = 0,328_{\text{рад}}$. Не все неявные методы обладают $A(\alpha)$ -устойчивостью, например, его нет у неявных методов Рунге – Кутты выше второго порядка, методов Адамса – Мултона. Все явные методы не имеют области $A(\alpha)$ -устойчивости и относятся к ограниченно устойчивым методам.

В современных САПР используют в основном неявные методы, так как у них нет ограничений, связанных с плохой обусловленностью ММ объектов проектирования, что очень важно при проектировании таких сложных объектов, как СУ. В целом сравнения явных и неявных методов интегрирования ОДУ свидетельствует о большой универсальности последних. Явные методы могут давать лучшие результаты только в отдельных случаях анализа объектов с хорошо обусловленными ММ и играют в САПР СУ вспомогательную роль. Наиболее перспективными для использования в САПР СУ являются методы решения жестких систем ОДУ.

Лекция 10. Алгоритмы и методы анализа чувствительности и статистических испытаний ССУ в САПР

Анализ чувствительности ССУ. Методы анализа чувствительности СУ при их использовании в САПР. Статистический анализ СУ в САПР.

Анализ чувствительности ССУ

Верификация проектных решений СУ требует выполнения многовариантного анализа. Многовариантный анализ заключается в многократном решении задач одновариантного анализа (многократное решение систем уравнений ММ объекта) при изменении внутренних параметров СУ или внешних параметров. Выходные параметры объекта (вектор Y) являются функцией внутренних (вектор X) и внешних (вектор Q) параметров $Y = F(X, Q)$ [6; 7; 10].

Типовыми процедурами многовариантного анализа, реализуемыми в САПР, являются процедуры анализа чувствительности и статистического анализа.

Параметры СУ в процессе работы не остаются равным расчетным значениям, что объясняется изменением внешних условий, неточностью изготовления отдельных устройств системы, старением элементов и т. д. Зависимость характеристик системы от изменения каких-либо ее параметров оценивают *чувствительностью*. Под чувствительностью понимают свойство системы изменять режим работы вследствие отклонения каких-либо параметров от номинальных значений. Для числовой оценки чувствительности исполь-

зуют функции чувствительности, определяемые как частные производные от координат системы или показателей качества процессов управления по вариациям параметров: $u_{ji} = \left(\frac{\partial y_i}{\partial x_j} \right)^0$, где y_i – координаты системы; x_j – параметры системы.

Индекс 0 означает, что функция u_{ij} вычисляется при номинальных значениях параметров.

Система, значения параметров которой равны номинальным и не имеют вариаций, называется исходной системой, а движение в ней – основным движением. Система, в которой имеют место вариации параметров, называется варьированной системой, а движение в ней – варьированным. Разность между варьированным и основным движениями называют дополнительным движением.

Пусть исходная система описывается системой нелинейных дифференциальных уравнений

$$\frac{dy_i}{dt} = f_i(y_1, \dots, y_n; x_1, \dots, x_m). \quad (10.1)$$

Пусть в некоторый момент времени в системе произошли вариации параметров Δx_j , где $j = 1, 2, \dots, m$; тогда параметры станут равными $x_j + \Delta x_j$. Если вариации параметров не вызывают изменения порядка уравнения, то варьированное движение описывается новой системой n уравнений первого порядка

$$\frac{d\tilde{y}_i}{dt} = \tilde{f}_i(\tilde{y}_1, \dots, \tilde{y}_n; x_1 + \Delta x_1, \dots, x_m + \Delta x_m). \quad (10.2)$$

Разность решений уравнений (10.1) и (10.2) определяет дополнительное движение

$$\Delta x_i(t) = \tilde{x}_i(t) - x_i(t). \quad (10.3)$$

Если Δy_i и y_i дифференцируемы по x_j , то дополнительное движение (10.3) можно разложить в ряд Тейлора по степеням Δx_j . При малых вариациях параметров ограничимся в разложении лишь линейными членами. Нужно отметить, что в случае конечных вариаций такое приближение недопустимо. Итак, можно записать уравнения первого приближения для дополнительного движения:

$$\Delta x_i(t, \Delta x_1, \dots, \Delta x_m) = \sum_{j=0}^m \left(\frac{dy_i}{d\Delta x_j} \right) \Delta x_j. \quad (10.4)$$

Учитывая формулу (10.1), можно записать

$$\Delta x_i(t, \Delta x_1, \dots, \Delta x_m) = \sum_{j=0}^m u_{ij} \Delta x_j. \quad (10.5)$$

Следовательно, располагая функциями чувствительности и задаваясь вариациями параметров, можно определить первое приближение для дополнительного движения.

Продифференцируем уравнения исходной системы (10.2) по Δx_j :

$$\frac{\partial}{\partial x_j} \left(\frac{dy_i}{dt} \right) = \frac{d}{dt} \left(\frac{\partial y_i}{\partial x_j} \right) = \frac{du_{ij}}{dt} = \sum_{k=1}^n \frac{\partial f_i}{\partial y_k} u_{kj} + \frac{\partial f_i}{\partial x_j}, i = 1, 2, \dots, n; j = 1, 2, \dots, m. \quad (10.6)$$

Полученные линейные дифференциальные уравнения (10.6) называют уравнениями чувствительности. Решение их дает функции чувствительности u_{ij} . В силу сложности уравнений (10.6) их решение весьма затруднительно. Поэтому для автоматизации метода анализа чувствительности разработаны и используются в САПР ряд численных методов.

Анализ чувствительности позволяет оценить степень влияния каждого параметра x_j и q_k на выходные параметры объекта, т. е. оценить чувствительность выходных параметров к изменению отдельных внутренних или внешних параметров. В процессе проектирования СУ результаты анализа чувствительности имеют большое значение, так как появляется возможность оценить стабильность выходных параметров при наличии дестабилизирующих факторов; определить наихудшие режимы работы объекта с точки зрения выполнения ТЗ. Анализ чувствительности способствует правильному назначению допусков на внутренние параметры.

Анализ чувствительности применяется по отношению к объектам с непрерывными ММ. Результаты анализа чувствительности находят применение при решении задач параметрической оптимизации внутренних параметров объекта с целью улучшения его выходных параметров, так как позволяют определить, какие внутренние параметры и в каком направлении следует изменять в первую очередь.

Количественно степень влияния внутренних и внешних параметров на выходные оценивается с помощью коэффициентов влияния (чувствительности) – частных производных.

$$a_{ij} = \frac{\partial y_i}{\partial x_j}$$

– абсолютный коэффициент чувствительности

$$b_{ij} = \frac{a_{ij} \cdot x_{j \text{ ном}}}{y_{i \text{ ном}}}$$

– относительный коэффициент чувствительности.

$x_{j\text{ном}}, y_{i\text{ном}}$ – номинальное значение параметров.

Значения a_{ij}, b_{ij} для всех выходных и изменяемых внутренних параметров составляют матрицы чувствительности A и B , размера $(m \times n)$ при $x = (x_1, x_2, \dots, x_n)$ и $y = (y_1, y_2, \dots, y_m)$.

Каждая строка матрицы A является вектором – градиентом одного из выходных параметров в пространстве внутренних параметров

$$A_j = \text{grad } y_i(x) = \left(\frac{\partial y_i}{\partial x_1}, \frac{\partial y_i}{\partial x_2}, \dots, \frac{\partial y_i}{\partial x_n} \right).$$

Каждый столбец матрицы A характеризует влияние одного из внутренних параметров на все выходные параметры. В частных случаях может потребоваться вычисление только части матрицы чувствительности (например, градиент одного из выходных параметров). Формулы, явно выражающие коэффициент влияния через известные параметры объекта, могут быть получены в сравнительно редких случаях, когда используют аналитические модели объектов в виде $Y = F(X, Q)$. Тогда коэффициенты влияния определяются обычным дифференцированием выражений обычной модели.

Методы анализа чувствительности СУ при их использовании в САПР

В общем случае формулы $Y = F(X, Q)$ неизвестны и для анализа чувствительности используются численные методы (рассматриваются методы применительно к внутренним параметрам, так как они в равной степени применимы и к внешним параметрам) [13]:

Метод приращений – основан на численном дифференцировании зависимостей $Y(X)$. Выполняется $(n + 1)$ вариант анализа. В первом варианте принимается $X = X_{\text{ном}}$, результат анализа есть $Y_{\text{ном}}$. В каждом последующем $(i + 1)$ варианте, $i = 1, 2, \dots, n$, задаётся отклонение Δx_i от $x_{i\text{ном}}$ только одному из внутренних параметров. Вычисляется значение Y_i вектора Y и рассчитывается очередной i -й столбец A_i матрицы чувствительности $A_i = \frac{Y_i - Y_{\text{ном}}}{\Delta X_i}$, ΔX_i – выбирают обычно в пределах 5–10 % от $X_{i\text{ном}}$.

Достоинства метода – универсальность, возможность распараллеливания вычислительного процесса. Недостатки – увеличение трудоёмкости, уменьшение точности.

Прямой метод – используется, когда ММ объекта есть система ОДУ, а выходные параметры – функционалы результатов интегрирования (определяются по результатам анализа переходных процессов). Пусть ММ объекта представлены в виде системы ОДУ $F(\dot{V}, V, X, t) = 0$ с вектором НУ V_0 ,

где V – вектор фазовых переменных размерности n ; X – вектор внутренних параметров (изменяемых) размерности m .

Эта система уравнений называется основной. Продифференцируем её по i -му элементу вектора X

$$\frac{\partial F}{\partial \dot{V}} \cdot \frac{\partial \dot{V}}{\partial X_i} + \frac{\partial F}{\partial V} \cdot \frac{\partial V}{\partial X_i} + \frac{\partial F}{\partial X_i} = 0.$$

Обозначив $\frac{\partial V}{\partial X_i} = z_i$, получим вспомогательную систему из n линейных дифференциальных уравнений, называемых моделью чувствительности

$$\frac{\partial F}{\partial \dot{V}} \cdot \dot{z}_i + \frac{\partial F}{\partial V} \cdot z_i + \frac{\partial F}{\partial X_i} = 0.$$

Вектор z_i – вектор чувствительности фазовых переменных к изменению i -го параметра.

Преобразование матрицы Z , составленной из столбцов z_i , в матрицу чувствительности выполняется по алгоритмам, зависящим от способа определения выходных параметров Y как функционалов зависимости $V(t)$. Если y_j – значение V_k вектора V в момент времени T , то $A_j = z_k(T)$, где A_j – j -я строка матрицы A , z_k – k -я строка матрицы Z .

Если

$$y_j = \int_0^T \psi[V(x,t), x, t] dt,$$

то

$$A_j = \int_0^T \left[\left(\frac{\partial \psi}{\partial V} \right) \cdot z + \left(\frac{\partial \psi}{\partial x} \right) \right] dt.$$

Прямой метод точнее и экономичнее метода приращений, однако используется лишь при указанных ограничениях, имеет высокий объем вычислений при небольших m . Этот недостаток устранен в вариационном методе.

Метод сопряжённых уравнений – для определения строки матрицы чувствительности интегрируется дополнительная система ОДУ, называемая сопряжённой.

Статистический анализ СУ в САПР

При учете производственного разброса внутренних параметров относительно номинальных значений компоненты вектора X следует рассматривать как случайные величины. Выходные параметры также будут иметь разброс относительно номинальных значений. Цель *статистического анализа* – по-

лучение информации о рассеянии выходных параметров Y и расчёт вероятности выполнения условий работоспособности. На основе статистического анализа прогнозируется возможный процент брака из-за невыполнения условий работоспособности, а при учёте старения внутренних параметров возможна оценка надёжности проектируемого объекта [5; 6; 10].

Статистический анализ сводится к определению основных статистических характеристик выходных параметров проектируемого объекта: плотности распределения этих параметров (гистограмм), математического ожидания (номинальных значений), средних квадратических отклонений (дисперсий), коэффициента корреляции и т. д. Исходными данными для статистического анализа являются технические требования на выходные параметры, предельно допустимые отклонения внешних параметров, сведения о законах распределения внутренних параметров. В большинстве случаев точная статистическая информация о внутренних параметрах отсутствует, но и при наличии приближённых исходных данных об их разбросе статистический анализ даёт полезную для проектирования информацию.

Для определения разброса выходных параметров объекта относительно номинальных значений используют метод наихудшего случая и вероятностные методы. Метод наихудшего случая используют для оценки влияния изменений внешних параметров на разброс выходных, так как правильное функционирование проектируемого объекта должно обеспечиваться при любых значениях внешних параметров внутри заданных диапазонов.

С помощью вероятностных методов оценивается влияние случайного разброса внутренних параметров на разброс выходных параметров. В отношении внутренних параметров ориентация на наихудший случай приводит к неоправданно жестким требованиям к диапазонам их изменения. Это связано с тем, что вероятность возникновения наихудшего случая мала и реальный разброс выходных параметров намного меньше, чем предсказанный для наихудшего сочетания внутренних параметров с учетом их разброса.

Таким образом, сначала определяют неблагоприятные значения внешних параметров по методу наихудшего случая, при которых далее проводят более достоверный анализ выходных параметров по методу статистических испытаний.

Метод наихудшего случая. По этому методу вычисляются значения выходных параметров в наихудших случаях. Наихудшим случаем для некоторого выходного параметра y_i называют случай, когда внутренние параметры объекта X и внешние параметры объекта Q принимают самые неблагоприятные из допустимых с точки зрения выполнения условий работоспособности на параметр y_i . В качестве исходных данных указываются максимально возможные отклонения Δq_i внешних параметров, которые обычно известны из ТЗ на проектирование.

Наихудший случай по параметру y_i соответствует максимальным отклонением Δq_k от номинальных значений $q_{kном}$ в сторону ухудшения выход-

ного параметра y_i . Направление отклонения параметров q_k определяется знаком коэффициента чувствительности

$$\text{sign}\left(\frac{\partial y_i}{\partial q_k}\right) = \text{sign}(a_{ik}).$$

Алгоритм метода наихудшего случая:

- Анализ чувствительности выходного параметра y_i к изменению внешнего параметра q_k и определение знака коэффициента чувствительности $\text{sign}(a_{ik})$ для всех i и k ;
- Задание значения параметров q_k для наихудшего случая $q_{k\text{нс}} = q_{k\text{ном}} \pm \Delta q_k \cdot \text{sign}(a_{ik})$, где знак «+» соответствует условию $y_i < \text{ТТ}_i$, знак «-» соответствует условию $y_i > \text{ТТ}_i$, ТТ_i – технические требования на y_i ;
- Выполнение одновариантного анализа объекта с расчётом выходного параметра y_i в наихудшем случае и определение возможного разброса этого параметра относительно номинального значения;
- Для каждого из выходных параметров имеется свой наихудший случай, следовательно, этапы 2 и 3 должны выполняться один раз.

Данный алгоритм предполагает неизменность знаков $\text{sign}\left(\frac{\partial y_i}{\partial q_k}\right)$ в допустимой области изменения вектора Q .

Влияние внешних параметров на разброс выходных параметров должно оцениваться по методу наихудшего случая, поскольку правильное функционирование проектируемого объекта должно обеспечиваться при любых значениях внешних параметров внутри заданного диапазона.

Метод статических испытаний (Монте-Карло) позволяет строить простые алгоритмы численного анализа стохастических систем. Случайные функции, входящие в описание исходной системы, воспроизводятся по их вероятностным характеристикам. Для каждой выборки случайных значений интегрируется исходная система уравнений ММ СУ, причем такое интегрирование производится многократно. Выборки значений случайных величин задаются программно, например, с помощью генератора случайных чисел.

При реализации метода статистических испытаний в САПР применительно к СУ исходными данными являются сведения о законах распределения внутренних параметров системы X , в том числе о взаимной корреляционной связи между этими параметрами и условия работоспособности по всем выходным параметрам Y . Статистические связи внутренних параметров между собой задаются в виде коэффициентов корреляций, вычисленных на основании результатов измерения этих параметров. Для получения таких данных выполняются экспериментальные измерения для большого количества комплектующих изделий, что весьма трудоемко, и разрабатываются соответствующие программы статистической обработки полученных результатов. По-

этому для реализации статистического анализа как проектной процедуры в САПР должны быть алгоритмы и программы двух групп: для статистической обработки результатов измерений; для выполнения статистического анализа с исходными данными, полученными с помощью алгоритмов и программ первой группы.

Наибольшее распространение получил численный метод статистического анализа – метод Монте-Карло – основанный на многократном моделировании (N раз) числовых значений вектора X для m случайных внутренних параметров x_j и вычислении для очередного испытания соответствующих значений всех выходных параметров Y .

При программной реализации метода Монте-Карло статистические испытания проводят с ММ объекта проектирования, где при этом моделируется задание случайных значений параметров компонентов ММ.

Алгоритм статистического анализа по методу Монте-Карло включает N -кратное выполнение анализа работы объекта, в каждом варианте анализа задаются случайные значения внутренним параметрам X в соответствии с их законами распределения и фиксируются значения выходных параметров, т. е. каждый вариант анализа работы объекта и представляет собой очередное статистическое испытание. Результаты испытаний обрабатываются с целью получения оценок числовых характеристик распределений выходных параметров и графиков статистических распределений (гистограмм).

Сложность задания случайных значений вектора X внутренних параметров модели обуславливается разнообразием законов распределения и коррелированностью элементов вектора X между собой. Выработка случайных значений внутренних параметров выполняется с помощью специальных алгоритмов. Для заданных законов распределения и коэффициентов корреляции для всех элементов вектора X возникает задача моделирования этих законов. Чтобы решить данную задачу, проводят преобразования произвольно заданных законов распределения x_j к теоретическим (обычно нормальному или равномерному). При статистическом анализе для параметров, у которых нет сведений о законе распределения, также предполагается нормальный закон распределения. Выработка случайных значений усложняется при наличии корреляционной связи между внутренними параметрами. Поэтому необходимо найти преобразование исходных статистических данных в многомерное теоретическое распределение с независимыми составляющими. Тогда обратное преобразование позволит моделировать произвольно распределенные случайные векторы X с реально существующими статистическими связями между его элементами.

Пусть L – m -мерный нормированный, нормально распределенный вектор с некоррелированными элементами; P – m -мерный нормально распределенный вектор с коррелированными элементами, отражающими реальные связи элементов в объекте.

Подготовительный этап для проведения статистических испытаний заключается в определении вектор-функций F и H прямого $P = F(X)$ и обратного $X = H(P)$ преобразований. Пусть известны гистограммы распределения для всех внутренних параметров x_j . Рассмотрим последовательность преобразования некоторой случайной величины x с произвольным распределением в величину p , имеющую нормированное нормальное распределение:

$$\varphi_p(p) = \frac{1}{\sigma_p \sqrt{2\pi}} t[h(-0.5(p - M_p) / \sigma_p),$$

где $\varphi_p(p)$ – плотность распределения p ; M_p и σ_p – математическое ожидание и среднеквадратичное отклонение случайной величины p , причем $M_p = 0$; $\sigma_p = 1$.

Плотности распределения случайных величин x и p связаны соотношением $\varphi_p(p)dp = \varphi_x(x(p)) dx$. При этом

$$\int_{x_k}^{x_{k+1}} \varphi_x(x)dx \approx \frac{\Delta N_{k+1}}{N}, \quad (10.7)$$

где x_k и x_{k+1} – границы $(k + 1)$ -го интервала гистограммы распределения величины x ; ΔN_{k+1} – число попаданий в $(k + 1)$ -й интервал гистограммы при общем количестве измерений величины x , равному N . В то же время интеграл в (10.7) равен

$$\int_{p(x_k)}^{p(x_{k+1})} \varphi_p(p)dp = \Phi(U_{k+1}) - \Phi(U_k), \quad (10.8)$$

где Φ – интеграл вероятностей, а $U_k = (p(x_k) - M_p) / \sigma_p = p(x_k)$.

Из (10.7) и (10.8) следует; что

$$\Phi(u_{k+1}) = \frac{\Delta N_{k+1}}{N} + \Phi(u_k).$$

Пусть известны U_0 , $\Phi(U_0)$ и ω, T, f , по (10.8) найдем $\Phi(U_1)$ и по таблице интеграла вероятностей $U_1 = (p(x_1))$, затем аналогично определим $\Phi(U_2)$ и U_2 и т. д. В результате получим зависимость прямого $p_k = f(x_k)$ и обратного $x_k = h(p_k)$ преобразований в табулированном виде. Совокупность функций h представляет собой преобразование H , с помощью которого можно задать случайное значение внутренним параметрам на основе вектора P , т. е. $X = H(P)$.

В качестве исходных данных для реализации m взаимно коррелированных значений внутренних параметров удобно использовать последовательность некоррелированных нормально распределенных нормированных случайных чисел l_j . Векторы L и P связаны соотношением $P = A_{LP}L$, где A_{LP} – матрица линейного преобразования. Способ получения матрицы A_{LP} давно известен и представлен в литературе, в частности, на основе корреляционной матрицы вторых моментов.

Значения внешних параметров Q должны выбираться, исходя из требований метода наихудшего случая, следовательно, статистический анализ должен начинаться с анализа чувствительности выходных параметров к изменению внешних параметров. Выбор режима статистических испытаний по внешним параметрам (номинальной, либо по наихудшему случаю) находится в компетенции проектировщика.

Алгоритм рабочего этапа метода Монте-Карло:

- задание случайных значений вектора L ;
- преобразование векторов L в $P = A_{LP}L$ и P в $X = H(P)$;
- одновариантный анализ работы объекта с расчетом Y при данном X ;
- повторение пунктов 1–3 до конечного числа испытаний N ;
- статистическая обработка результатов расчета.

Точность метода Монте-Карло во многом зависит от заданного количества испытаний N . Если задать погрешность оценки M_i и σ_i в пределах (0,01–0,001) % с доверительной вероятностью 0,95, то $N = 10^8$. Однако на практике подобная точность не требуется, так как исходные данные имеют большую погрешность. Обычно $N = 50 \div 200$ при этом погрешность оценки M_i составляет (12–24) %, σ_i – (10–23) % с доверительной вероятностью 0,9–0,95.

Достоинством метода Монте-Карло является простота реализации; универсальность, так как используется при произвольных законах распределения X , а также при нелинейных и неявных видах связи между X и Y , что является важным фактором при особенностях математического описания такого сложного и физически разнородного объекта, как современные СУ.

МОДУЛЬ 3. МОДЕЛИ СИНТЕЗА ССУ И ВЕРИФИКАЦИИ ПРОЕКТНЫХ РЕШЕНИЙ ПРИ АВТОМАТИЗАЦИИ ЭТАПА ПРОЕКТИРОВАНИЯ

ТЕМА 5. МЕТОДЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ: МЕТОДЫ СИНТЕЗА ССУ

Лекция 11. Методы и алгоритмы технической оптимизации ССУ в САПР

Параметрическая оптимизация средств и систем управления. Постановка задачи параметрической оптимизации: критерии оптимальности как функции качества СУ, назначение ограничений при решении задачи технической оптимизации, нормирование управляемых и выходных параметров ССУ. Методы и алгоритмы оптимизации ССУ в интегрированных САПР

Параметрическая оптимизация средств и систем управления

В зависимости от целей решения в САПР различают следующие задачи оптимизации [5; 6; 7; 10]:

- оптимизация параметров;
- оптимизация допусков и технических требований;
- идентификация параметров моделей и построение областей адекватности.

Последняя задача оптимизации входит в методику получения математических моделей элементов.

Проектирование – это сложный процесс, объединяющий такие важные процедуры, как синтез структуры, выбор параметров элементов, анализ и принятие решений. Начинается проектирование со структурного синтеза, при котором генерируется принципиальное решение. Таким решением может быть облик будущего летательного аппарата или физический принцип действия датчика, или одна из типовых конструкций двигателя, или функциональная схема микропроцессора. Но эти конструкции и схемы выбирают в параметрическом виде, т. е. без указания числовых значений параметров элементов. Поэтому прежде чем приступить к верификации проектного решения, нужно задать или рассчитать значения этих параметров, т. е. выполнить па-

раметрический синтез. Примерами результатов параметрического синтеза могут служить геометрические размеры деталей в механическом узле или в оптическом приборе, параметры электрорадиоэлементов в электронной схеме, параметры режимов резания в технологической операции и т. п.

Таким образом, поиск рационального технического решения при выбранном принципе действия осуществляется методами *структурного синтеза*. Определение оптимальных значений параметров технического объекта (системы) известной структуры – задача *параметрического синтеза или параметрической (технической) оптимизации*. Параметрическая оптимизация проводится в случае, если по результатам анализа проектное решение признается неокончательным, тогда начинается процесс последовательных приближений к приемлемому варианту проекта. Во многих случаях для улучшения проекта удобнее варьировать значения параметров элементов, т. е. использовать параметрический синтез на базе многовариантного анализа. При этом задача параметрического синтеза может быть сформулирована как задача определения значений параметров элементов, наилучших с позиций удовлетворения требований технического задания при неизменной структуре проектируемого объекта.

В САПР процедуры параметрического синтеза выполняются либо человеком в процессе многовариантного анализа (в интерактивном режиме), либо реализуются на базе формальных методов оптимизации (в автоматическом режиме). В последнем случае находят применение несколько постановок задач оптимизации.

Наиболее распространенной является детерминированная постановка: заданы условия работоспособности на выходные параметры Y и нужно найти номинальные значения проектных параметров X , к которым относятся параметры всех или части элементов проектируемого объекта. Эта задача оптимизации называется оптимизацией параметров. В частном случае, когда требования к выходным параметрам заданы нечетко, к числу рассчитываемых величин могут быть отнесены также нормы выходных параметров, фигурирующие в их условиях работоспособности.

Если проектируются изделия для дальнейшего серийного производства, то важное значение приобретает такой показатель, как процент выпуска годных изделий в процессе производства. Очевидно, что успешное выполнение условий работоспособности в номинальном режиме не гарантирует их выполнения при учете производственных погрешностей, задаваемых допусками параметров элементов. Поэтому целью оптимизации становится максимизация процента выхода годных, а к результатам решения задачи оптимизации относятся не только номинальные значения проектных параметров, но и их допуски. Эта задача оптимизации называется оптимизацией допусков и технических требований.

Формализация задачи оптимизации параметров технической системы управления сводится к ее формулировке в виде задачи математического программирования:

$$\begin{cases} \text{extr } F(X) \\ X \in XD \\ XD = \{X / \Phi(X) > 0, \varphi(X) = 0\} \end{cases}, \quad (11.1)$$

где $F(X)$ – целевая функция, X – вектор управляемых (проектных) параметров, $\varphi(X)$ и $\Phi(X)$ – функции-ограничения, XD – допустимая область в пространстве управляемых параметров. Запись (11.1) интерпретируется как задача поиска экстремума целевой функции путем варьирования управляемых параметров (аргументов целевой функции) в пределах допустимой области.

Постановка задачи оптимизации имеет смысл только тогда, когда есть выбор одного из конкурирующих вариантов, полученных при ограниченности ресурсов. Техническое проектирование всегда ведется в условиях жестких ограничений на материальные, энергетические, временные и другие виды ресурсов.

Критерии и ограничения при проектировании СУ многочисленны и противоречивы, существенно зависят от типа СУ и условий их эксплуатации. Отмеченное обстоятельство иллюстрируется составом целей и критериев, содержащихся в ТЗ на СУ. Критерии, принимаемые при проектировании СУ, можно представить следующими векторами:

качество $K = (k_1, k_2, \dots, k_n)$,
 мощность $M = (M_1, M_2, \dots, M_n)$,
 масса(вес) $B = (B_1, B_2, \dots, B_n)$,
 габаритные размеры $\Gamma = (\Gamma_1, \Gamma_2, \dots, \Gamma_n)$,
 надежность $H = (H_1, H_2, \dots, H_n)$,
 ремонтпригодность $P = (p_1, p_2, \dots, p_n)$,
 стоимость $C = (c_1, c_2, \dots, c_n)$.

Компоненты этих векторов отражают соответствующие составляющие, имеющие естественное физическое содержание: k_1, \dots, k_n – запасы устойчивости, время переходного процесса, динамические и статические ошибки, колебательность, оценки эффективности; M_1, \dots, M_n – мощность, потребляемая устройствами СУ, рассеиваемая энергия, энергетические ресурсы источников питания; B_1, \dots, B_n – вес отдельных устройств и элементов САУ; $\Gamma_1, \dots, \Gamma_n$ – габариты отдельных устройств и элементов САУ; H_1, \dots, H_n – среднее время между отказами, восстанавливаемость, показатели старения САУ, ее устройств и элементов; p_1, \dots, p_n – ремонтпригодность устройств и элементов, восстанавливаемость, доступность при ремонте; c_1, \dots, c_n – стоимость отдельных устройств и элементов и всей системы.

Перечисленные критерии показывают сложность задачи их объединения в единую математическую форму – функционал, причем такой функционал должен связывать структуру и параметры СУ с целями их проектирования.

САПР позволяет разработать несколько альтернативных вариантов технических решений СУ. Окончательный выбор варианта СУ проводится с учетом выработанных правил предпочтения на основании установленных критериев. В основе построения правила предпочтения лежит целевая функция, количественно выражающая качество, поэтому ее называют также функцией качества или *критерием оптимальности*. Формирование целевой функции всегда выполняется с учетом выходных параметров объекта.

В большинстве подходов к оценке СУ принято ориентироваться на эталонные образцы, на мнение ведущих специалистов отрасли (экспертные оценки) или на технико-экономические показатели, определяемые техническим заданием (ТЗ) на проектирование. При подготовке ТЗ обычно учитываются достижения, полученные в мировой практике. Традиционный процесс проектирования СУ, как правило, использует эволюционный подход, который включает анализ предыдущих проектов и действующих СУ, выбор ближайшего прототипа – «старый проект», внесение в него изменений в соответствии с требованиями на «новый проект», многочисленные проверки соответствия «исправленного старого проекта» новым требованиям (целям, критериям), возвраты к точкам изменений при несоответствии результатов проектирования этим требованиям. Движение от исходных данных и сформулированных вначале целей проектирования к проекту носит итерационный характер, он никогда не является прямым, когда непосредственно по исходным данным, целям получают проект без каких-либо возвратов к предыдущим шагам. Более того, на основе ограниченных исходных данных шаг за шагом от приближенных очертаний проекта происходит его развитие до тех пор, пока проект не будет удовлетворять поставленным целям.

Качество функционирования любого технического объекта характеризуется вектором выходных параметров Y . Среди выходных параметров одна часть может быть измерена количественно, другая часть представляет собой качественную сторону объекта.

Содержание ТЗ включает в себя конкретные числовые требования к основным выходным параметрам (*технические требования*); конкретные числовые данные, характеризующие работу объекта во внешней среде (диапазоны изменения внешних параметров: температуры, влажности, напряжения и частоты источников питания и пр.); качественное описание требований, ограничений и условий, не поддающихся количественной оценке.

Цели создания СУ во многом определяются назначением объекта управления. В каждом случае перед проектным предприятием формулируется основная цель создания СУ:

- обеспечение полета ЛА по заданной траектории и стабилизация его положения по отношению к этой траектории, более конкретно – управление полетом самолета типа ТУ по маршруту в установленном эшелоне протя-

женностью в 2000 км с точностью стабилизации по угловым параметрам до $0,1^\circ$, стабилизации высоты полета 8000 м с точностью 10 м, стабилизации скорости (числа Маха) с точностью 2 %; сохранение неизменными в пространстве заданной оси или, в более общем случае, системы координат из трех осей, материализованных в гиростабилизированной платформе (ГСП), конкретно – стабилизация платформы с установленными на ней «пассажирами» (пеленгаторы, антенны и т. д.) в инерциальном пространстве (относительно осей x, y, z , направленных на звезды) с точностью не хуже $2'$ по каждой из осей;

- управление перемещением шагающего робота по заданной поверхности с препятствиями (перемещение в произвольном направлении с преодолением препятствий высотой или глубиной до 30 см);

- удержание в требуемых пределах скорости вращения вала турбины и давления пара турбогенератора, стабилизация частоты вращения в пределах 2–3 % при изменении нагрузки от 10 до 20 %.

Требуемые соотношения между выходными параметрами и техническими требованиями (ТТ) называют *условиями работоспособности* и записывают в виде $y_j < TT_j$. Условия работоспособности имеют определяющее значение в разработке СУ, так как задачей проектирования становится выбор проектного решения, в котором наилучшим образом выполняются все условия работоспособности во всем диапазоне изменения внешних параметров и при выполнении всех качественных требований ТЗ.

Следовательно, для постановки и решения задачи параметрической оптимизации в целях наилучшего выполнения условий работоспособности объекта (системы), уменьшения затрат ресурсов на его производство необходимо формирование целевой функции $F(X)$, отражающей качество проектируемого объекта. Выбор целевой функции носит субъективный характер, поэтому объект (СУ) может быть оптимальным только в смысле данного критерия.

Постановка задач параметрической оптимизации ССУ

Для выполнения расчета номинальных значений параметров ТО необходимо сформулировать задачу в виде (11.1) и решить задачу поиска экстремума $F(X)$. Для этого нужно, во-первых, формализовать понятие «оптимальный», во-вторых, решить задачу, уже имеющую математическую формулировку. Решение задачи строгими математическими методами может быть выполнено после того как задача поставлена. Сама постановка задачи ведется с учетом назначения реального объекта и целей проектирования. Процедура постановки задачи оптимизации носит неформальный характер и включает этапы: выбор целевой функции и управляемых параметров; назначение ограничений; нормирование параметров.

Сложность постановки оптимизационных проектных задач обусловлена наличием у проектируемых объектов нескольких выходных параметров,

которые могут быть критериями оптимальности, но в задаче (11.1) целевая функция должна быть одна. Другими словами, проектные задачи являются многокритериальными, и возникает проблема сведения многокритериальной задачи к однокритериальной, что называется сверткой векторного критерия. Среди выходных параметров всегда можно найти такие пары, у которых улучшение одного параметра приводит к ухудшению другого. Такие параметры называют конфликтными, т. е. при оптимизации ТО невозможно улучшение всех выходных параметров одновременно.

В зависимости от того, каким образом выбираются и объединяются выходные параметры, в скалярной функции качества различают критерии оптимальности:

1) *Частный критерий* применяется, когда среди выходных параметров можно выделить один основной, наиболее полно отражающий эффективность проектируемого объекта. Этот параметр принимают за целевую функцию. Например, грузоподъемность для транспортных средств, мощность для энергетического объекта. Условия работоспособности остальных выходных параметров относят к ограничениям задачи (11.1). Достоинством является простота, недостатком – большой запас работоспособности можно получить только по основному параметру, который принят в качестве целевой функции, а другие выходные параметры не будут иметь запасов.

2) *Аддитивный критерий* объединяет (свертывает) все выходные параметры (частные критерии) в одну целевую функцию, представляющую собой взвешенную сумму частных критериев

$$F(X) = \sum_{j=1}^m \omega_j y_j(X), \quad (11.2)$$

где ω_j – весовой коэффициент, m – число выходных параметров. Функция (4.2) подлежит минимизации, при этом если условие работоспособности имеет вид $y_j > T_j$, то $\omega_j < 0$. Недостатки аддитивного критерия – субъективный подход к выбору весовых коэффициентов, которые выбираются либо самим инженером-проектировщиком, либо группой экспертов и в процессе оптимизации остаются неизменными, а также неучет требований ТЗ. Действительно в (11.2) не входят нормы выходных параметров.

3) *Мультипликативный критерий* используется, когда отсутствуют условия работоспособности типа равенств и выходные параметры не могут принимать нулевые значения. Целевая функция имеет вид

$$F(X) = \prod_{j=1}^m (y_j(X))^{\alpha_j}. \quad (11.3)$$

Нетрудно видеть, что если прологарифмировать (11.3), то мультипликативный критерий превращается в аддитивный. Достоинством является отсутствие нормирования выходных параметров. Недостатки те же, что у аддитив-

ного критерия. Общим недостатком двух приведенных выше критериев является улучшение одних выходных параметров за счет недопустимого ухудшения других.

4) *Максиминный (минимаксный) критерий* – наиболее предпочтительный критерий при проектировании ТО. Он позволяет достичь наилучшего удовлетворения условий работоспособности. В качестве целевой функции принимается выходной параметр, наиболее неблагоприятный с позиций выполнения условий работоспособности. Для оценки степени выполнения условия работоспособности j -го выходного параметра вводят запас работоспособности этого параметра S_j , а этот запас можно рассматривать как нормированный j -й выходной параметр. Например (здесь и далее для лаконичности изложения предполагается, что все выходные параметры приведены к виду, при котором условия работоспособности становятся неравенствами в форме $y_j < TT_j$):

$$S_j = (TT_j - y_j) / TT_j$$

или

$$S_j = a_j((TT_j - y_{j\text{ном}}) / \delta_j - 1) ,$$

где $y_{j\text{ном}}$ – номинальное значение, a_j – весовой коэффициент, δ_j – некоторая характеристика рассеяния j -го выходного параметра, например, трехсигмовый допуск. Тогда целевая функция в максиминном критерии есть

$$F(X) = \min_{j \in [1:m]} S_j(X).$$

Здесь запись $[1: m]$ означает множество целых чисел в диапазоне от 1 до m . Достоинством максиминного критерия является то, что на целевую функцию влияет только тот y_j , который в данной точке x является наихудшим с позиций выполнения требований ТЗ.

Значения внешних параметров при оптимизации выбирают либо номинальными, либо исходя из расчета на наихудший случай.

Назначение ограничений – второй этап процедуры постановки задачи оптимизации. Ограничения объективно появляются при проектировании ТО и вытекают из конкретной физической и технологической реализуемости внутренних параметров элементов, ограниченности ресурсов и т. п. Различают прямые и функциональные ограничения.

Прямые ограничения имеют вид $x_{i\text{min}} \leq x_i \leq x_{i\text{max}}$, где $x_{i\text{min}}$, $x_{i\text{max}}$ – минимально и максимально допустимые значения i -го параметра. Для многих объектов параметры элементов не могут быть отрицательными, например, геометрические размеры, масса и т. д.

Функциональные ограничения представляют условия работоспособности выходных параметров, не вошедших в целевую функцию. Функциональные ограничения могут быть типа равенств $\Phi(X) = 0$ и типа неравенств $\varphi(X) > 0$.

Прямые и функциональные ограничения формируют допустимую область поиска.

Нормирование управляемых и выходных параметров – третий этап процедуры постановки задачи оптимизации. Пространство управляемых параметров – метрическое, поэтому при выборе направлений и величин шагов поиска необходимо вводить ту или иную норму, отождествляемую с расстоянием между двумя точками. Это предполагает одинаковую размерность (или безразмерность) управляемых параметров.

Применяют следующие способы нормирования:

– логарифмическое $z_i = \ln\left(\frac{x_i}{\xi_i}\right)$, где z_i – нормированный управляемый

параметр, x_i – ненормированный управляемый параметр, ξ_i – коэффициент, численно равный единице параметра x_i ;

– линейное $z_i = \frac{x_i}{\xi_i}$;

– нелинейное $z_i = \operatorname{tg}\left(\pi \frac{x_i - (x_{i\max} + x_{i\min})/2}{x_{i\max} - x_{i\min}}\right)$, $z_i \in [z_{i\min}, z_{i\max}]$.

Методы и алгоритмы оптимизации ССУ в интегрированных САПР

Решение задач оптимизации ССУ в САПР сопряжено с рядом особенностей, среди которых в качестве основных можно назвать следующие:

- неоднозначность способа формирования целевой функции;
- сложность формулировки ограничений;
- неоднозначность выбора системы единиц физических величин x , от которой зависят масштабы по осям и, следовательно, вид $F(X)$;
- неявный, алгоритмический характер задания выходных параметров, входящих в целевую функцию, так как они вычисляются алгоритмически – через численное решение системы ОДУ – и поэтому нельзя предсказать зависимость $y_j(x)$, в частности их выпуклость или вогнутость, играющих важную роль при выборе метода оптимизации для решения задач;
- высокая «стоимость» значений $F(X)$ из-за больших затрат времени на вычисление, поскольку каждое вычисление $F(X)$ требует расчета схемы;
- невозможность вычисления производных целевой функции по аналитическим формулам и высокая стоимость их вычисления численными методами.

Указанная специфика задач технической оптимизации, особенно неизвестность рельефа и большое время вычисления целевой функции и ее производных, приводит к тому, что класс этих задач, допускающий решение строгими методами математической оптимизации, оказывается весьма невелик, и во многих случаях приходится использовать эвристические приемы. Поэтому решение задач оптимизации в САПР ведется с помощью *поисковых методов*, сводящихся к многошаговому вычислительному процессу приближения к искомому экстремуму. Каждый шаг процесса заключается в переходе из точки X_{k-1} в пространстве управляемых параметров в точку X_k . Для такого перехода нужно определить направление перемещения $g_k = \frac{P_k}{\|P_k\|}$ и величину шага h_k в этом направлении, такие, что

$$X_k = X_{k-1} + h_k g_k.$$

Получающаяся последовательность точек $(X_0, X_1, X_2, \dots, X_N)$, называемая траекторией поиска, должна сходиться к экстремальной точке X^* . Окончивается поиск с попаданием текущей точки поиска X_k в заданную окрестность экстремальной точки. Однако точка X^* неизвестна заранее и возможны различные варианты условия окончания поиска.

По ряду признаков задачи поиска экстремума могут быть отнесены к тому или иному классу. Большинство постановок задач параметрической оптимизации технических объектов сводятся к задачам нелинейного программирования, так как целевая функция и ограничения описываются нелинейными зависимостями от вектора управляемых параметров. Иногда удается так сформулировать задачу, что целевая функция и ограничения являются линейными функциями своих аргументов. Тогда решается задача линейного программирования.

Теоретические методы оптимизации подразделяются:

1. По порядку используемых в них производных целевой функции $F(X)$ на методы:

- нулевого порядка (без производных);
- первого порядка;
- второго порядка.

2. По отношению к ограничениям на методы:

- безусловной оптимизации (без учета ограничений);
- условной оптимизации (с учетом ограничений).

3. По отношению к рельефу целевой функции на методы:

- поиска локального экстремума;
- поиска глобального экстремума.

4. По количеству варьируемых переменных на методы:

- одномерного поиска;
- многомерного поиска.

5. По способу изменения варьируемых переменных на методы:
- детерминированные;
 - случайные.

Практически используемые методы, как правило, являются методами локального поиска. Это основные методы в САПР, так как методов глобальной оптимизации, обеспечивающих нахождение глобального экстремума с приемлемыми потерями на поиск, для задачи математического программирования общего вида (11.1) не существует. В САПР поиск глобального экстремума осуществляется путем локальной оптимизации из нескольких исходных точек, выбираемых случайным образом в пределах области, задаваемой прямыми ограничениями.

Наиболее многочисленную группу составляют методы безусловной оптимизации. Задача условной оптимизации может быть сформулирована как задача безусловной оптимизации с помощью методов Лагранжа или штрафных функций.

Простейшими являются методы одномерного поиска. Самостоятельного значения они обычно не имеют, а играют вспомогательную роль, обеспечивая поиск оптимума по направлениям, вычисляемым и используемым в более сложных многомерных методах. К методам одномерной оптимизации относятся методы дихотомического деления, золотого сечения, чисел Фибоначчи, полиномиальной аппроксимации и ряд их модификаций.

Содержанием любого метода поисковой оптимизации является способ выбора:

1) направления поиска g_k :

– координатное – циклы по n шагов, направление поиска на i -ом шаге цикла вдоль i -й координатной оси;

– градиентное $P_{k+1} = -gradF(Z_k)$;

– ньютоновское $\Delta Z_{k+1} = -Y_k^{-1} gradF(Z_k)$, где $Y_k = \frac{\partial^2 F(Z)}{\partial Z^2}$ – матрица вторых частных производных $F(Z)$ по Z (матрица Гессе);

– случайное $G = \frac{\xi}{\|\xi\|}$, где ξ – вектор со случайными составляющими.

2) величины шага h_k :

– способ оптимального шага, в котором h_k есть результат решения задачи одномерной минимизации целевой функции вдоль прямой, проходящей через точку Z_{k-1} параллельно вектору g_k $\min_{h_k} F(Z_{k-1} + h_k g_k)$;

– способ задаваемого шага, при котором значение h_k уменьшается в m раз между сериями шагов, оставаясь постоянным внутри серий. Исходное значение шага и m задаются. Переход к новой серии осуществляется при выполнении условий попадания в заданную окрестность экстремальной точки.

3) формул для нормирования управляемых параметров;

4) критерия окончания поиска – окончание поиска обычно осуществляют по правилу: если на протяжении r подряд идущих шагов траектория поиска остается в малой ε -окрестности текущей точки поиска Z_k , то поиск

следует прекратить, условия окончания поиска $\|Z_k - Z_{k-r}\| < \delta$ и $h_k \leq h_{\min}$, где δ и h_{\min} – заданные константы.

Эффективность поиска зависит от того, как сделан этот выбор. Составляющими эффективности являются надежность, точность, экономичность. Надежность определяется как вероятность достижения заданной ε -окрестности экстремальной точки при применении данного метода; точность характеризуется гарантированным значением ε ; экономичность отождествляется с потерями на поиск. Потери на поиск выражают трудоемкость процедуры оптимизации, которую в большинстве случаев оценивают количеством обращений к ММ технического объекта.

Используемые в САПР методы оптимизации обычно группируются в зависимости от постановки решаемого класса задач.

Методы нулевого порядка для многомерного поиска требуют значительно меньше времени на выполнение шага оптимизации, чем методы первого и второго порядка, связанные с вычислением первых и вторых производных $F(X)$. Теоретически методы высоких порядков сходятся к оптимуму быстрее методов нулевого порядка, однако это справедливо только для функций определенного вида, например, выпуклых. Однако целевые функции электронных схем, как правило, таким свойством не обладают вообще. Поэтому на практике часто методы нулевого порядка оказываются предпочтительнее методов первого и второго порядков, особенно с учетом высоких вычислительных затрат для определения производных $F(X)$.

Поскольку рельеф целевой функции в задачах автоматизации схемотехнического проектирования заранее неизвестен, то особо важную роль в оптимизации схем играют методы поиска глобального экстремума. Из известных методов высокой надежностью и простотой реализации зарекомендовал себя метод искусственного отбора.

Выбор методов оптимизации – это прежде всего выбор между методами нулевого порядка (без производных) и методами, требующими вычисления производных. Решение этой проблемы тесно связано с общей концепцией оптимизации проектируемого объекта. Поскольку целевая функция вычисляется алгоритмически и не имеет формульной записи, то ее рельеф заранее неизвестен, что не позволяет проектировщику выбрать наиболее эффективный алгоритм оптимизации. Этим объясняется сложившаяся парадигма в области методологии оптимизации:

1. Исследование рельефа целевой функции;
2. Глобальную оптимизацию с целью выхода в область оптимума глобальными методами;
3. Уточнение оптимума методами локальной оптимизации.

Первый этап необходим для определения дальнейшей стратегии оптимизации. Если окажется, что рельеф функции весьма сложен, носит многоэкстремальный или многоовражный характер, то эффективный поиск экстремума на основе какого-либо одного алгоритма невозможен, так как свойства функции вблизи и вдали от точки оптимума могут сильно различаться.

Например, в целом по области оптимизации функция может иметь седловые точки, гребни, овраги, несколько экстремумов. Оптимум таких функций трудно или невозможно отыскивать методами типа градиентных. В то же время вблизи от точки оптимума функция может быть достаточно гладкой, выпуклой и поиск оптимума градиентными методами может оказаться наиболее быстрым. Отсюда вытекает необходимость введения в парадигму оптимизации второго и третьего этапов.

Лекция 12. Формализация сведений о ССУ как объектах структурного синтеза

Структурный синтез технических систем в САПР. Классификация процедур структурного синтеза СУ: по целям синтеза и содержанию результатов; по трудностям формализации процедур синтеза; по типу синтезируемых структур. Подходы к алгоритмизации задач структурного синтеза ССУ: перебор законченных структур; наращивание структуры; выделение варианта из обобщенной структуры; трансформация описаний.

Структурный синтез технических систем в САПР

Понятие «синтез» технической системы близко по содержанию к понятию «проектирование». Задача синтеза системы заключается в получении проектного решения по закону функционирования СУ. Синтез технических систем нацелен на создание новых вариантов [6; 7; 8; 10].

Под структурой системы понимают набор составляющих ее элементов и связей между ними. Задачей структурного синтеза является поиск оптимальной структуры СУ для реализации заданных функций в рамках выбранного принципа действия. Результаты структурного синтеза в САПР могут быть представлены в виде перечня элементов вместе с таблицей соединений, схемы расположения элементов с указанием их типов и т. п.

Структурный синтез – наиболее трудная для формализации проектная процедура. Существует ряд общих подходов к постановке задач структурного синтеза, однако практическая реализация большинства из них неочевидна. Поэтому имеются лишь «островки» автоматического выполнения процедур синтеза среди «моря» проблем, ждущих автоматизации. В существующих САПР синтез выполняет человек, а ПК используется для верификации предлагаемых вариантов. Дальнейшее увеличение степени автоматизации проектных работ определяется успехами в постановке и алгоритмизации задач структурного синтеза. Однако в ряде приложений имеются и примеры успешной автоматизации структурного синтеза; среди них в первую очередь заслуживают упоминания задачи конструкторского проектирования печатных плат и кристаллов БИС, логического синтеза комбинационных схем цифровой автоматики и вычислительной техники, синтеза технологических

процессов и управляющих программ для механообработки в машиностроении и некоторые другие. При этом перечисленные примеры, как правило, относятся к решению задач структурного синтеза на этапе конструкторского и технологического проектирования, тогда как этап функционального проектирования является наиболее трудным с точки зрения формализации задач структурного синтеза, что обусловлено прежде всего сложностью математического описания СУ и сложностью организации вычислительных алгоритмов.

Кроме того, сложность формализованного синтеза структур заключается в большом количестве факторов, влияющих на свойства и параметры структур, а также в трудностях решения задач оптимизации повышенной размерности. Поэтому решение задач структурного синтеза необходимо организовывать по блочно-иерархическому принципу. Достигнутая степень обобщения выражается в установлении типичной последовательности действий и используемых видов описаний при их преобразованиях в САПР. Исходное описание, как правило, представляет собой ТЗ на проектирование СУ, по нему составляют описание на некотором формальном языке, являющемся входным языком используемых подсистем САПР. Затем выполняют преобразования описаний, и получаемое итоговое для данного этапа описание документируют – представляют в виде твердой копии или файла в соответствующем формате для передачи на следующий этап.

Важное значение для развития подсистем синтеза в САПР имеют разработка и унификация языков представления описаний (спецификаций). Каждый язык, поддерживая выбранную методику принятия решений, формирует у пользователей САПР – разработчиков технических систем определенный стиль мышления; особенности языков непосредственно влияют на особенности правил преобразования спецификаций. Примерами унифицированных языков описания проектных решений являются язык VHDL для радиоэлектроники, он сочетает в себе средства для функциональных, поведенческих и структурных описаний, язык Express – универсальный язык спецификаций для представления и обмена информацией в компьютерных средах.

Классификация процедур структурного синтеза СУ

Процедуры структурного синтеза классифицируются:

1) *По целям синтеза и содержанию результатов:*

- выбор принципов построения и функционирования технических систем – производится на стадии научно-исследовательских работ (НИР) с целью установления физических, информационных, организационных принципов;
- выбор технического решения – на стадии опытно-конструкторских работ (ОКР) с целью получения функциональных, принципиальных схем, технологических маршрутов и т. п.;

- синтез технической документации – на стадии технического и рабочего проектирования для автоматического преобразования данных о схемах и конструкциях, выраженных на внутреннем языке САПР, в текстовую и чертежную документацию оформленную по ЕСКД.

2) По трудностям формализации процедур синтеза выделяют:

- задачи 1-го уровня сложности – структура объекта predetermined результатами предварительной НИР, либо выбор структуры однозначен. Синтез сводится к выбору числовых значений параметров для заданной структуры;

- задачи 2-го уровня сложности – выбор структуры из конечного множества вариантов при условиях, что все варианты известны и мощность множества вариантов мала;

- задачи 3-го уровня сложности – выбор варианта в конечном множестве, мощность которого велика, чтобы реализовать полный перебор используются алгоритмы направленного перебора;

- задачи 4-го уровня сложности – выбор варианта структуры в множествах, мощность которых априорно неизвестна (не исключена возможность, что она неограниченна), подходом к решению задач 4-го уровня является теория эвристических решений;

- задачи 5-го уровня сложности – поиск решений, основанных на новых, ранее неизвестных идеях и принципах. В задачах предыдущих уровней существование решений не подвергалось сомнению и требовалось найти лучшее или приемлемое решение. В задачах 5-го уровня достижение решения – получение принципиально нового типа технических объектов. Алгоритмизация задач невозможна; в рамках САПР решается не сама задача, а задачи, возникшие после того как человек сформулировал идеи нового решения.

Таким образом, одним из подходов к решению задач синтеза практически на всех уровнях сложности являются понижение уровня сложности задачи за счет уменьшения числа проб при переборе вариантов ограничением множества рассматриваемых вариантов, а также уменьшение затрат на каждую пробу за счет упрощения ММ технических систем.

3) По типу синтезируемых структур:

- задачи одномерного синтеза – построение одномерных последовательностей из элементов некоторой природы, например, описание вычислительных процессов в виде алгоритмов и программ;

- схемный синтез – разработка различных схем (функциональных, структурных, принципиальных и т.п.), отражающих результаты проектирования объектов до конкретизации их геометрических форм;

- геометрический синтез – выполняется при конструировании изделий и связан с определением их геометрических форм и расположением объекта или его частей в пространстве относительно заданных ориентиров.

Средства САПР, ориентированные на автоматизацию процедур структур синтеза, в той или иной мере опираются на идеи и методы искусственно-

го интеллекта (ИИ). ИИ – это наука о знаниях, способах их получения, представления, переработки и использования в искусственных системах.

Системы ИИ, используемые в САПР:

1) информационно-поисковые системы с интерфейсом на основе естественного языка – банк данных для хранения слабоструктурированной информации. Например, тексты с описанием физических явлений;

2) интеллектуальные пакеты прикладных программ для инженерных расчетов – сведения о предметной области представлены в виде отношений между параметрами и переменными;

3) интеллектуальные программно-методические комплексы (ПМК) для моделирования и анализа систем – ММ технического объекта формируется ПМК на основе знаний о построении разнообразных объектов в данной предметной области по заданным пользователем сведениям о структуре конкретного устройства

4) экспертные системы – выполняют функцию эксперта в определенной предметной области.

Формализация процедур структурного синтеза в САПР осуществляется путем формализации знаний о предметной области, к которой относятся синтезируемые объекты.

Сведения об объектах синтеза включают в себя данные о базовых элементах, макроэлементах, законченных и обобщенных структурах. Объект синтеза – представления о реальных объектах, выражаемых в виде описаний проектируемых изделий и процессов на одном из языков проектирования, т. е. чертежи, схемы, списки и т. п.

Базовый элемент – элементарная часть синтезируемого объекта, которую невозможно или нецелесообразно разделять на составные части (R , C , и т. д.). Основную часть сведений об элементе составляет перечень его свойств, называемых атрибутами. В перечне указывают имена атрибутов, их значения, ссылки на другие элементы и процедуры, с помощью которых конкретизируется и выявляются допустимые типы связей и т. п.

Макроэлемент – типовая совокупность взаимосвязанных базовых элементов, используемая при синтезе.

Законченная структура – совокупность взаимосвязанных примитивов и макроэлементов, представляющая собой возможный вариант структуры проектируемого объекта. Например, законченная структура при проектировании локальных сетей – структура сетей Ethernet.

Обобщенная структура строится на основе многих частных законченных структур. В ней отражаются все элементы, макроэлементы и связи между ними, которые встречались хотя бы в одной из частных структур. Одинаковые части входят в обобщенную структуру без дублирования, что делает ее лаконичной.

В указанных формах сведений об объектах выделяют описания собственно элементов и отношения между ними.

В системах ИИ для формализации декларативных знаний используются способы, основанные на понятиях фрейма и семантической сети.

Фрейм – структура данных, в которой в определенном порядке представлены сведения о свойствах объекта. Типичный вид фрейма:

$$\langle \text{имя фрейма}; a_1=p_1; a_2=p_2; \dots; a_n=p_n; q_1, q_2, \dots, q_m \rangle,$$

где a_i – имя i -го атрибута, p_i – его значение; q_m – ссылка на некоторый другой фрейм или процедуру.

С помощью фреймов можно описывать иерархические структуры. При этом в качестве a_i и p_i используется имя и значение i -го слота. Слот – вложенная во фрейм структура данных. Организация слота аналогична структуре фрейма. В общем случае во фрейме может быть более двух иерархических уровней. Фреймы – естественная форма представления сведений об элементах синтезируемых объектов в системах структурного синтеза. Различают:

1) символический фреймы – атрибуты являются переменными, для них указывается только область определения, например, $\langle \text{резистор}; \text{номинал} = x_1; \text{мощность} = x_2; \text{класс точности} = x_3; \text{тип конструкции} = x_4 \rangle$, где x_1, x_2, x_3, x_4 – переменные, которые могут принимать значения из соответствующих множеств.

2) конкретные фреймы – атрибуты принимают фиксированные значения.

Совокупность конкретных фреймов в некоторой подсистеме САПР представляет собой базу данных. На основе символических фреймов выполняются функции систем ИИ с помощью операций поиска по образцу и т. п.

Семантическая сеть – форма представления знаний в виде совокупности понятий и отношений между ними в некоторой предметной области. Семантическая сеть удобно изображать графом, в котором вершины отображают понятия, а ребра – отношения между ними. В качестве вершин могут фигурировать фреймы, представляющие сведения об элементах. Структура проектируемого объекта может быть представлена в виде семантической сети, если вершинам поставить в соответствие элементы, а ребрам – соединения элементов или другие взаимодействия.

В системах структурного синтеза семантические сети используются для представления обобщенных структур проектируемых объектов. Обобщенные структуры имеют иерархическую организацию, отражают множество вариантов построения объектов и называются И-ИЛИ-деревьями. В И-ИЛИ-дереве вершины делятся на ярусы (уровни). Вершины могут быть двух типов И и ИЛИ. Вершины И отображают варианты технической реализации составных частей объектов, вершины ИЛИ – составные части объектов, выделенные по функциональному признаку. Если на верхнем ярусе вершины ИЛИ соответствуют функциональному назначению систем из некоторого класса, то вершины И соседнего яруса отображают способы реализации систем. Вершины И и ИЛИ последующих ярусов отображают функции и способы построения подсистем, блоков, узлов и базовых элементов. Ветви в И-ИЛИ-дереве представляют классификационные и признаковые отношения между понятиями.

На [рис. 12.1](#) приведен пример И-ИЛИ-дерева для вторичных источников питания. Вершины ИЛИ показаны прямоугольниками, вершины И – овалами. Сплошной линией изображены ветви, соответствующие классификационным отношениям, пунктирной – признаковым отношениям.

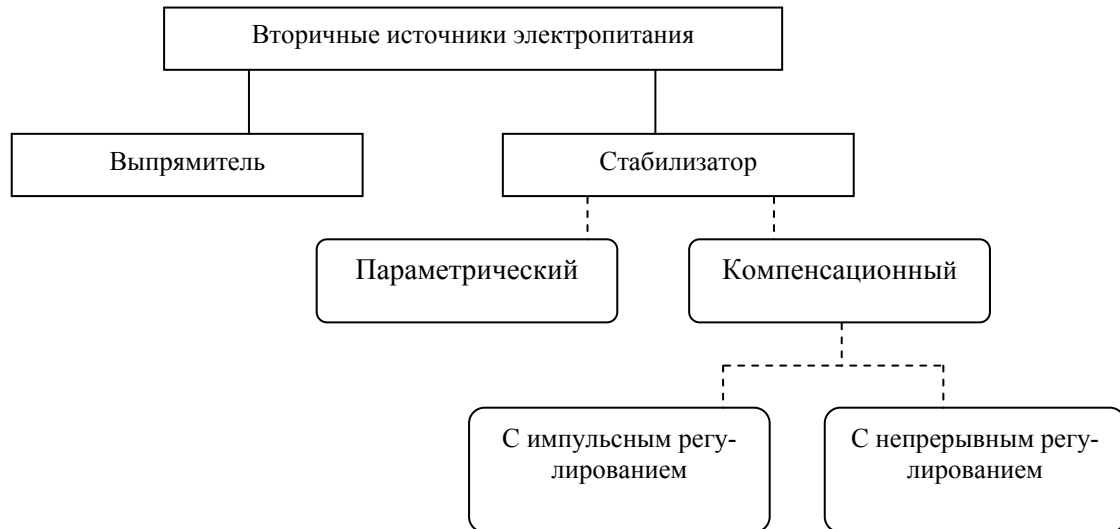


Рис. 12.1

Аналогично разделению фреймов на символические и конкретные семантические сети могут быть: интенциональные семантические сети – выражают общие закономерности, присущие данной предметной области и составляют базу знаний; экстенциональные семантические сети – выражают сведения о конкретных фактах или структурах и составляют базу данных САПР.

Подходы к алгоритмизации задач структурного синтеза ССУ

На практике используется несколько подходов к алгоритмизации задач структурного синтеза, каждый из которых может быть применен при разработке СУ. Каждый из подходов требует определенных сведений о предметной области и о требованиях к создаваемой системе. В каждом из подходов разработаны алгоритмы синтеза, применяемые в САПР, в которых можно выделить блоки формирования очередного варианта, его оценки и принятия решений.

Перебор законченных структур применяется, когда оценка промежуточных вариантов затруднительна. Формирование вариантов возможно выбором варианта из имеющейся библиотеки типовых решений, либо генерацией структуры из элементов по заданным правилам. Полный перебор осуществляется для задач второго уровня сложности. Многие задачи синтеза в своей исходной постановке относятся к третьему уровню сложности, полный

перебор невозможен из-за больших вычислительных затрат, поэтому перебор может быть только частичным, основанным на приближенных эвристических алгоритмах. Приближенные алгоритмы известны лишь для отдельных задач, поэтому практическая реализация формирования очередного варианта в большинстве задач перебора возлагается на пользователя, работающего с САПР в интерактивном режиме. Оценка вариантов структуры, предлагаемых пользователем, производится системой проектирования.

Среди задач, решаемых путем частичного перебора, выделяют задачи дискретного программирования:

$$\begin{cases} \text{Extr } F(X) \\ X \in XD \\ XD = \{X \in D / \varphi(X) \geq 0, \Psi(X) = 0, \end{cases}$$

где $F(x)$ – целевая функция; X – вектор управляемых параметров, определенный на дискретном множестве векторов D ; $\varphi(x)$ – функциональные ограничения.

Кроме того, основу большой группы математических методов, выражающих стремление к сокращению перебора, составляют операции разделения множества вариантов на подмножества и отсеечения неперспективных подмножеств. Эти методы объединяются под названием метода ветвей и границ. Основная разновидность метода ветвей и границ относится к точным методам решения комбинаторных задач. Рассмотрим эту разновидность.

Пусть имеется множество решений M , в котором нужно выбрать оптимальный по критерию $F(X_j)$ вариант, где X_j – вектор параметров варианта $m_j \in M$; пусть также имеется алгоритм для вычисления нижней границы $L(M_k)$ критерия $F(X_j)$ в любом подмножестве M_k множества M , т. е. такого значения $L(M_k)$, что $F(X_j) \geq L(M_k)$ при любом j (подразумевается минимизация $F(X)$). Тогда основная схема решения задач в соответствии с методом ветвей и границ содержит следующие процедуры:

- 1) в качестве M_k принимаем все множество M ;
- 2) ветвление: разбиение M_k на несколько подмножеств M_q ;
- 3) вычисление нижних границ $L(M_q)$ в подмножествах M_q ;
- 4) выбор в качестве M_k подмножества M_g с минимальным значением нижней границы критерия (среди всех подмножеств, имеющих на данном этапе вычислений), сведения об остальных подмножествах M_q и их нижних границах сохраняются в отдельном списке;
- 5) если $|M_k| > 1$, то переход к процедуре два, иначе одноэлементное множество M_k и есть решение.

Метод ветвей и границ в случае точного вычисления нижних границ относится к точным методам решения задач выбора и потому в неблагоприятных ситуациях может приводить к экспоненциальной временной сложности.

сти. Однако метод часто используют как приближенный, поскольку можно применять приближенные алгоритмы вычисления нижних границ.

Среди других приближенных методов решения задачи дискретного программирования отметим метод локальной оптимизации. Так как пространство D метризовано, то можно использовать понятие α -окрестности $Sa(Xk)$ текущей точки поиска Xk . Вместо перебора точек во всем пространстве D осуществляется перебор точек только в $Sa(Xk)$. Если $F(Xj) \geq F(Xk)$ для всех $Nj \in Sa(Xk)$, то считается, что найден локальный минимум целевой функции в точке Xk . В противном случае точку Xq , в которой достигается минимум $\alpha F(X)$ в $Sa(Xk)$, принимают в качестве новой текущей точки поиска.

К задачам дискретного программирования можно свести многие процедуры выбора состава оборудования в СУ.

Наращивание структуры характеризуется последовательным добавлением элементов к некоторой первоначальной структуре постепенным включением новых элементов и связей с возможностью оценки промежуточных структур. При наращивании происходит целенаправленная генерация вариантов, поэтому алгоритмы наращивания оказываются более быстродействующими, хотя и менее точными по сравнению с переборными. Например, алгоритмы решения задач размещения, трассировки.

Выделение варианта из обобщенной структуры – последовательное исключение элементов из избыточной обобщенной структуры (отражает сведения о широком классе проектируемых систем и представлена в виде семантических сетей). Для получения нужного решения используется поиск по образцу, выделение сценариев (подграфов). Например, синтез технологических маршрутов изготовления деталей в подсистемах автоматизированной подготовки производства.

Трансформация описаний используется, когда известно исходное описание структуры СУ, соответствующее некоторому иерархическому уровню и аспекту проектирования, и нужно получить описание, соответствующее другому уровню или аспекту. Алгоритмы трансформации основаны на введении правил преобразования элементов и макроэлементов исходных структур в новые формы. Трансформация описаний широко используются в САПР. Например, синтез комбинационных логических схем (трансформация логических схем формул в схемное описание); оформление конструкторской документации; трансляции программ.

Итак, задача структурного синтеза заключается в выборе принципа действий технического объекта и в определении оптимальной структуры объекта для реализации заданных функций. Структура объекта определяется природой входящих в его состав элементов и физической реализацией связей между ними. Основной особенностью решения задач структурного синтеза с целью получения оптимального варианта структуры объекта является наличие ММ объекта, представляющей собой формальное описание множества структур на принятом уровне детализации.

Процесс проектирования структуры технической системы представляет собой многоэтапную процедуру:

Этап 1: построение допустимого множества (каталога) возможных типов элементов (информационное обеспечение САПР);

Этап 2: выбор множества допустимых систем элементов, используемых при структурном синтезе (не все элементы могут стыковаться друг с другом);

Этап 3: построение вариантов структур системы с учетом их технической и технологической реализации. Для систем небольшой сложности возможно построение полного множества допустимых структур СУ.

Этап 4: оценка очередного варианта требует формирования и анализа ММ синтезированной структуры системы и выполнения параметрической оптимизации. Для уменьшения сложности этого этапа используются ММ, соответствующие определенному уровню проектирования; косвенные критерии предпочтения вариантов. Например, сопоставление длины проводников для определения уровня помех, наводимых при возбуждении одного из проводников в соседних проводниках при конструировании электронных устройств СУ.

Этап 5: коррекция технического задания, если нет вариантов, обеспечивающих выполнения ТЗ.

Для разработки алгоритмов этапов 3 и 4 в задачах различной сложности используются следующие подходы. В задачах 1-го уровня структура технического объекта задана и решение задачи синтеза выливается в параметрическую оптимизацию. В задачах 2-го уровня возможен полный перебор вариантов. В задачах 3-го уровня используются алгоритмы направленного перебора; последовательные методы синтеза. В задачах 4-го уровня используется теория эвристических решений. Основным подходом является подход, основанный на активном участии человека в решении задач структурного синтеза технического объекта. Широко используются методы решения изобретательских задач.

Имеется ряд подходов для обобщенного описания задач принятия проектных решений в процессе структурного синтеза.

Задачу принятия решений (ЗПР) формулируют следующим образом:

$$\text{ЗПР} = \langle A, K, \text{Мод}, \Pi \rangle,$$

где A – множество альтернатив проектного решения, $K = (K_1, K_2, \dots, K_m)$ – множество критериев (выходных параметров), по которым оценивается соответствие альтернативы поставленным целям; $\text{Мод} : A \rightarrow K$ – модель, позволяющая для каждой альтернативы рассчитать вектор критериев, Π – решающее правило для выбора наиболее подходящей альтернативы в многокритериальной ситуации.

В свою очередь, каждой альтернативе конкретного приложения можно поставить в соответствие значения упорядоченного множества (набора) ат-

рибутов $X = (X_1, X_2, \dots, X_n)$, характеризующих свойства альтернативы. При этом x_i может быть величиной типа `real`, `integer`, `Boolean`, `string` (в последнем случае величину называют предметной или лингвистической). Множество X называют записью (в теории баз данных), фреймом (в искусственном интеллекте) или хромосомой (в генетических алгоритмах). Модель *Мод* называют структурно-критериальной, если среди x_i имеются параметры, характеризующие структуру моделируемого объекта.

Основными проблемами в ЗПР являются:

- компактное представление множества вариантов (альтернатив);
- построение модели синтезируемого устройства, в том числе выбор степени абстрагирования для оценки значений критериев;
- формулировка предпочтений в многокритериальных ситуациях (т. е. преобразование векторного критерия K в скалярную целевую функцию);
- установление порядка (предпочтений) между альтернативами в отсутствие количественной оценки целевой функции (что обычно является следствием неколичественного характера всех или части критериев);
- выбор метода поиска оптимального варианта (сокращение перебора вариантов).

Лекция 13. Алгоритмы и методы структурного синтеза ССУ в САПР

Методы математического программирования. Методы решения изобретательских задач. Методы с неопределенными исходными данными. Методы распространения ограничений. Переборные методы. Генетические алгоритмы.

Задачи синтеза являются важнейшими и в то же время наиболее сложными при проектировании различных систем и процессов. Как уже отмечалось в предыдущей лекции, некоторые задачи синтеза могут быть решены методами анализа. Более того, опытный инженер-проектировщик часто находит вполне пригодные структуру и параметры САУ на основе приближенных расчетов и эвристических методов. Однако трудоемкость такого синтеза оказывается весьма значительной. Так, надо осуществить до нескольких сотен вариантов анализа при выборе корректирующего звена СУ турбогенератором или параметров СУ ЛА и т. д. Полученные проектные решения требуется «доводить» на последующих этапах проектирования, возможны и принципиальные ошибки в проектных решениях [4; 13].

При синтезе структуры и параметров СУ, как правило, не учитываются конструкторские и технологические особенности реализации отдельных устройств и элементов СУ и особенно стыковки их между собой. Например,

корректирующее звено с передаточной функцией $\frac{T_1^2 P^2 + T_2 P + 1}{(T_3 P + 1)(T_4 P + 1)}$, полу-

ченное методами традиционного синтеза, может оказаться нереализуемым (в частности, на пассивных или активных фильтрах), если не удастся обеспечить требуемого соотношения между постоянными времени T_1, T_2, T_3, T_4 в условиях конструктивных и технологических ограничений.

Прямая реализация теоретических и инженерных методов синтеза, не ориентированных на ЭВМ, часто оказывается невозможной из-за неустойчивости вычислительного процесса, накопления методических и трансформированных ошибок, сложности логических операторов и т. д. Поэтому для применения традиционных методов синтеза в САПР требуется их машинная ориентация. Классическими методами структурного синтеза СУ являются частотные и корневые методы.

Частотные методы синтеза САУ строятся на основе критериев качества, связанных с ЧХ: запасы устойчивости по фазе (φ) и по амплитуде (L), показатель колебательности (M), частота среза разомкнутой системы ($\omega_{ср}$) при учете ограничений на величины динамической ($Y_{дин}$) и статической ошибок ($Y_{ст}$), обусловленных типовыми воздействиями.

Системы управления считаются приемлемыми, если обеспечены запасы по фазе и амплитуде, оговоренные для СУ заданного класса.

В простейшем случае такая задача решается путем машинной ориентации метода АФЧХ. В качестве исходных данных задается исходная структурная схема СУ, свойства которой не удовлетворяют критериям качества. На структурной схеме выделяются точки возможного съема и ввода сигналов коррекции, т. е. задается множество возможных мест включения коррекции. По заданным требованиям к динамике системы в соответствии с какой-либо из известных методик частотного синтеза формируются асимптотические логарифмические ЧХ (ЛЧХ) нескольких разомкнутых желаемых СУ, которые должны быть согласованы с ЛЧХ исходной разомкнутой системы и могут отличаться некоторыми параметрами [5; 6; 7; 10].

Автоматизация процедуры синтеза состоит в получении передаточной функции корректирующего звена, обеспечивающего совпадение ЛЧХ разомкнутых желаемой и скорректированной САУ. После получения такой передаточной функции могут быть осуществлены ее аппроксимация и подбор соответствующих параметров.

Корневые методы синтеза применительно к СУ основаны на построении траекторий корней характеристического полинома передаточной функции в зависимости от изменений синтезируемых параметров $W(p)$ в предположении, что структура $W(p)$ – известна. В качестве критерия синтеза используется, в частности, степень устойчивости СУ η .

Ориентация на классические методы при применении современных САПР делает этот инструмент проектирования при решении задач структурного синтеза СУ неэффективным, поскольку только алгоритмизация задач структурного синтеза СУ позволяет рассмотреть множество вариантов

структур, оценить их и принять проектное решение за приемлемое время. Поэтому при решении задач структурного синтеза ССУ с целью повышения эффективности их алгоритмов и методов следует ориентироваться на современные подходы, базирующиеся на развивающемся разном математическом аппарате, который довольно легко реализуется в рамках интегрированных САПР на основе современных технологий программирования. Учитывая, что основным назначением САПР является создание среды проектирования, удобной для проектировщика, значительное сокращение времени разработки, именно методы, основанные на подходах, рассмотренных ниже, должны быть ведущими при проектировании ССУ.

Методы математического программирования

Выбор метода поиска решения – вторая проблема после формализации задачи структурного синтеза ССУ. Если при формализации все управляемые параметры удалось представить в числовом виде, то можно попытаться применить известные методы математического программирования.

Методы линейного программирования. (ЛП), в частности, симплекс-метод Данцига. Основные условия, позволяющие построить модели ЛП, – это пропорциональность (затраты ресурсов прямо пропорциональны объему), аддитивность (общая величина равна сумме затрат ресурсов), неотрицательность (ни одному из видов деятельности не может быть приписан отрицательный уровень). Основным недостатком применения метода ЛП – размерность ММ не должна превышать трех.

Приближенные *методы дискретного программирования* для решения задач структурного синтеза, в частности, метод локальной оптимизации. Так как пространство D метризовано, то можно использовать понятие α -окрестности $S_\alpha(X_k)$ текущей точки поиска X_k . Вместо перебора точек во всем пространстве D осуществляется перебор только в $S_\alpha(X_k)$. Если $F(X_k) < F(X_j)$ для всех $X_j \in S_\alpha(X_k)$, то считается, что найден локальный минимум целевой функции в точке X_k . В противном случае достигается минимум $F(X)$ в $S_\alpha(X_k)$, а точку X_q принимают в качестве новой текущей точки поиска.

Недостатком метода является его явно выраженная локальность – «застревание» в окрестностях локальных экстремумов. Повысить эффективность поиска можно с помощью *метода поиска с запретами* (Tabu Search). Для этого в $S_\alpha(X_k)$ вводят запреты на попадание в некоторые точки. Обычно это запреты на повторное исследование точек, пройденных на нескольких последних шагах оптимизации. Запрет распространяется и на лучшую точку X_k предыдущего шага, которая может оказаться точкой локального минимума. Тогда на данном шаге перемещение происходит в лучшую незапрещенную точку X_{k+1} , несмотря на то, что $F(X_{k+1}) > F(X_k)$. Тем самым появляется тенденция к выходу из области притяжения локального экстремума.

Методы решения изобретательских задач

Эти методы не являются формальными, но позволяют проводить дальнейший поиск методов алгоритмизации задач структурного синтеза [10].

1) Метод морфологических таблиц. Морфологическая таблица (M) представляет собой обобщенную структуру в виде множества функций, выполняемых компонентами синтезируемых объектов рассматриваемого класса, и подмножеств способов их реализации. Каждой функции можно поставить в соответствие одну строку таблицы, каждому способу ее реализации – одну клетку в этой строке. Следовательно, в морфологических таблицах элемент M_{ij} означает j -й вариант реализации i -й функции в классе технических объектов, описываемом матрицей M .

Другими словами, множество альтернатив можно представить в виде отношения M , называемого морфологической таблицей $M = \langle X, R \rangle$, где X – множество свойств (характеристик или функций), присущих объектам рассматриваемого типа, n – число этих свойств, $R = \langle R_1, R_2, \dots, R_n \rangle$, R_i – множество значений (способов реализации) i -го свойства, мощность этого множества далее обозначена N_i . При этом собственно множество альтернатив A представлено композицией множеств R_i , т. е. каждая альтернатива включает по одному элементу (значению) из каждой строки морфологической таблицы. Очевидно, что общее число альтернатив k , представляемых морфологической таблицей, равно $k = \prod_{i=1}^n N_i$.

Морфологические таблицы обычно считают средством неавтоматизированного синтеза, помогающим человеку просматривать компактно представленные альтернативы, преодолевать психологическую инерцию. Последнее связано с тем, что внимание проектировщика обращается на варианты, которые без морфологической таблицы оставались бы вне его поля зрения. Собственно таблица M не содержит сведений о способе синтеза. Однако на базе M возможно построение методов синтеза с элементами алгоритмизации. В таких методах вводится метризация морфологического пространства. Морфологическое пространство составляют возможные законченные структуры, принимается, что расстояние между структурами C_1 и C_2 есть число несовпадающих элементов (каждая клетка M есть один элемент). Поэтому можно говорить об окрестностях решений. Далее исходят из предположения о компактности «хороших» решений, которое позволяет вместо полного перебора ограничиваться перебором в малой окрестности текущей точки поиска. Таким образом, гипотеза о «компактности» и метризация пространства решений фактически приводят к построению математической модели, к которой можно применить методы дискретной оптимизации, например локальные методы.

К недостаткам M относят неучет запрещенных сочетаний элементов в законченных структурах и отражение состава элементов в структурах без конкретизации их связей. Кроме того, морфологические таблицы строят в

предположении, что множества R_i взаимно независимы, т. е. состав способов реализации i -й функции не меняется при изменении значений других функций. Очевидно, что предположение о взаимной независимости множеств R_i оправдано лишь в сравнительно простых структурах. Последний недостаток устраняется путем обобщения метода морфологических таблиц – при использовании метода альтернативных (И-ИЛИ) графов.

2) Альтернативные графы (И-ИЛИ-графы). Любую морфологическую таблицу можно представить в виде дерева (рис. 13.1). На рисунке функции представлены вершинами И (темные кружки), значения функций – вершинами ИЛИ (светлые кружки). Очевидно, что таблица представляет множество однотипных объектов, поскольку все они характеризуются одним и тем же множеством функций.

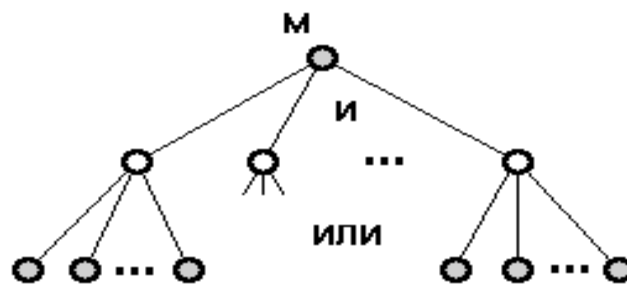


Рис. 13.1

Деревья с вершинами И и ИЛИ называют *И-ИЛИ-деревьями*. В общем случае разные функции могут быть реализованы одними и теми же способами, тогда вместо дерева имеем граф, называемый *И-ИЛИ-графом* (или альтернативным графом).

Для разнотипных объектов применяют многоярусные альтернативные графы. Например, на рис. 13.2 показан двухярусный граф, в котором для разных типов объектов предусмотрены разные подмножества функций.

Если допустить некоторую избыточность при изображении И-ИЛИ-графа, то его можно превратить в И-ИЛИ-дерево, что ведет к определенным удобствам.

Очевидно, что И-ИЛИ-дерево можно представить как совокупность морфологических таблиц. Каждая И вершина дерева соответствует частной морфологической таблице, т. е. множеству функций так, что i -я выходящая ветвь отображает i -ю функцию. Каждая ИЛИ вершина, инцидентная i -й ветви, соответствует множеству вариантов реализации i -й функции, при этом j -я исходящая из ИЛИ вершины ветвь отображает j -й вариант реализации.

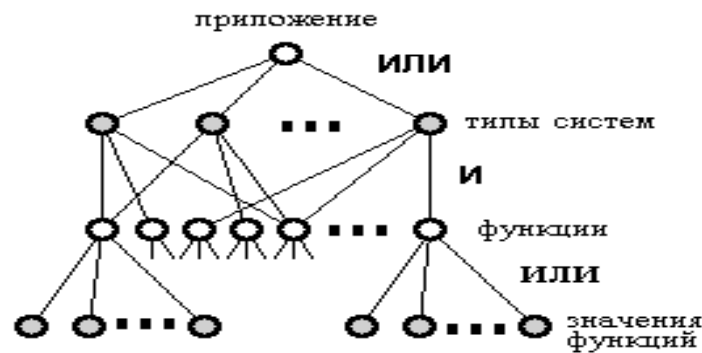


Рис. 13.2

Алгоритмизация структурного синтеза на базе И-ИЛИ-деревьев требует введения правил выбора альтернатив в каждой вершине ИЛИ. Эти правила чаще всего имеют эвристический характер, связаны с требованиями ТЗ, могут отражать запреты на сочетания определенных компонентов структур.

Трудности эффективного решения задачи существенно возрастают при наличии ограничений, типичными среди которых являются ограничения на совместимость способов реализации разных функций, т. е. ограничения вида

$$C_{ij} \wedge C_{pq} = false, \quad (13.1)$$

где $C_{ij} = true$, если в оцениваемый вариант вошел элемент \mathcal{E}_{ij} , иначе $C_{ij} = false$. Условие (13.1) означает, что в допустимую структуру не могут входить одновременно элементы \mathcal{E}_{ij} и \mathcal{E}_{pq} . Совокупность ограничений типа (13.1) можно представить как систему логических уравнений с неизвестными C_{ij} . Тогда задачу синтеза можно решать эволюционными методами, если предварительно или одновременно с ней решать систему логических уравнений (задачу о выполнимости).

2) Метод «мозгового штурма» осуществляет поиск нового с помощью двух групп специалистов. В первую группу предпочтительно включить людей с богатой фантазией. Назначение этой группы – генерирование идей для решения поставленной задачи. Предложения могут быть любыми, вплоть до самых фантастических и, на первый взгляд, нереальных. Именно среди таких неожиданных предложений наиболее вероятно встретить свежую принципиально новую идею и с ее помощью добиться цели. Поэтому при работе первой группы запрещается какая-либо критика высказанных идей. В составе группы не обязательно должны быть специалисты по данному узкому вопросу, так как преследуется цель преодоления психологической инерции мышления, которая выражается в склонности людей действовать в соответствии с уже освоенными ими правилами и выбирать построения, аналогичные известным для них прообразам. Высказанные идеи фиксируются и в дальнейшем анализируются представителями второй группы специалистов, среди которых предпочтительно иметь людей с аналитическим складом ума. Их за-

дача – поиск перспективных идей среди предложений, полученных от представителей первой группы.

3) Метод инверсии основан на преодолении психологической инерции. Он рекомендует изобретателю сознательно искать нестандартные пути подхода к проблеме, менять местами компоненты проектируемого объекта, например, части – подвижную и неподвижную, верхнюю и нижнюю и т. д.

4) Метод аналогии основан на обращении к смежным техническим дисциплинам. Известны успехи новой науки – бионики, которая исследует возможности применения принципов деятельности живых организмов для построения технических устройств.

Методы с неопределенными исходными данными

Присущая проектным задачам неопределенность и нечеткость исходных данных, а иногда и моделей, диктуют использование специальных методов количественной формулировки исходных неколичественных данных и отношений. Эти специальные методы либо относятся к области построения измерительных шкал, либо являются предметом теории нечетких множеств.

Измерительные шкалы могут быть [10]:

1) абсолютными;

2) номинальными (классификационными), значения шкалы представляют классы эквивалентности, примером может служить шкала цветов; такие шкалы соответствуют величинам неколичественного характера;

3) порядковыми, если между объектами A и B установлено одно из следующих отношений: простого порядка, гласящее, что если A лучше B , то B хуже A , и соблюдается транзитивность; или слабого порядка, т. е. либо A не хуже B , либо A не лучше B ; или частичного порядка. Для формирования целевой функции $F(X)$ производится оцифровка порядковой шкалы, т. е. при минимизации, если A предпочтительнее B , то $F(X_a) < F(X_b)$, где X_a и X_b – множества атрибутов объектов A и B соответственно;

4) интервальными, отражающими количественные отношения интервалов: шкала единственная с точностью до линейных преобразований, т. е. $y=ax+b$, $a > 0$, $-\infty < b < +\infty$, или $y = ax$ при $a \neq 0$, или $y = x + b$.

В большинстве случаев структурного синтеза математическая модель в виде алгоритма, позволяющего по заданному множеству X и заданной структуре системы рассчитать вектор критериев K , оказывается известной. Например, такие модели получаются автоматически в программах анализа типа Spice, Adams или ПА-9 для объектов, исследуемых на макроуровне. Однако в ряде других случаев такие модели неизвестны в силу недостаточной изученности процессов и их взаимосвязей в исследуемой среде, но известна совокупность результатов наблюдений или экспериментальных исследований. Тогда для получения моделей используют специальные методы иденти-

фикации и аппроксимации (модели, полученные подобным путем иногда называют феноменологическими).

Среди методов формирования модели по экспериментальным данным наиболее известны методы планирования экспериментов. Не менее популярным становится подход, основанный на использовании искусственных нейронных сетей.

Если же математическая модель остается неизвестной, то стараются использовать подход на базе систем искусственного интеллекта (экспертных систем).

Возможности практического решения задач дискретного математического программирования (ДМП) изучаются в теории сложности задач выбора, где показано, что задачи даже умеренного размера, относящиеся к классу NP -полных задач, в общем случае удается решать только приближенно.

Поэтому большинство практических задач структурного синтеза решают с помощью приближенных (эвристических) методов. Это методы, использующие специфические особенности того или иного класса задач и не гарантирующие получения оптимального решения. Часто они приводят к результатам, близким к оптимальным при приемлемых затратах вычислительных ресурсов.

Одним из подходов к решению задач принятия проектных решений является использование методов искусственного интеллекта и баз знаний.

Если все управляемые параметры альтернатив, обозначаемые в виде множества X , являются количественными оценками, то используют приближенные методы оптимизации.

Если в X входят также параметры неколичественного характера и пространство X неметризуемо, то перспективными являются эволюционные методы вычислений, среди которых наиболее развиты генетические методы.

Наконец, в отсутствие обоснованных моделей их создают, основываясь на экспертных знаниях в виде некоторой системы искусственного интеллекта.

Примерами применения генетических методов для решения задач, относящихся к проблематике ИПИ-технологий и САПР, могут служить синтез расписаний и распределение ресурсов в интегрированных логистических системах, оптимизация распределения нагрузки в корпоративных информационных сетях, распределение приложений по подсхемам виртуальной локальной вычислительной сети, решение задач конструкторского проектирования (компоновки, размещения и трассировки) и многие другие.

Методы распространения ограничений

Во многих задачах структурного синтеза множество D допустимых вариантов, задаваемое ограничениями $W(X) > 0$ и (или) $Z(X) = 0$, включает сравнительно малое число элементов, и в качестве результатов синтеза принимается любой из вариантов возможных сочетаний элементов. Такое реше-

ние задачи часто выполняют с помощью *метода распространения ограничений* (Constraints Propagation) [10].

Сущность этого метода заключается в сужении допустимых интервалов управляемых переменных X с помощью учета (распространения) исходных ограничений на выходные параметры W и Z .

Для пояснения метода рассмотрим простой пример для задачи поиска допустимых вариантов в множестве синтезируемых структур при ограничениях на совместимость элементов структуры.

Рассмотрим фрагмент структуры, состоящий из трех компонентов – A , B и C , причем $A \in \{a_1, a_2, a_3, a_4, a_5\}$, $B \in \{b_1, b_2, b_3, b_4\}$, $C \in \{c_1, c_2, c_3\}$. Заданы списки допустимых сочетаний компонентов в синтезируемой структуре:

$$\begin{aligned} L1 &: a_1, b_1; a_2, b_1; a_4, b_2; a_5, b_3; a_5, b_4; \\ L2 &: b_1, c_1; b_3, c_3; b_4, c_1; b_4, c_2; \\ L3 &: a_2, c_3; a_3, c_2; a_4, c_3; a_5, c_2. \end{aligned}$$

Сокращение первого списка выполняется путем поочередного выбора в нем a_i , фиксации в $L3$ соответствующих значений c_k , а затем в $L2$ сопряженных с c_k значений b_j . Если в $L1$ имеется элемент a_i, b_j , то он сохраняется в сокращенном списке, остальные сочетания с a_i из $L1$ удаляются. В нашем примере, поскольку значения a_1 в $L3$ нет, то сочетание a_1, b_1 недопустимо и из $L1$ удаляется. Далее для символа a_2 фиксируем в $L3$ значение c_3 , ему в $L2$ соответствует только значение b_3 . Поэтому a_2, b_1 – также недопустимое сочетание. Обработав подобным образом все списки, получаем результат распространения ограничений в виде $L1: a_5, b_4; L2: b_4, c_2; L3: a_5, c_2$.

Следовательно, решение состоит из единственной допустимой структуры, включающей компоненты a_5, b_4, c_2 .

В общем случае сокращение списков выполняется в итерационном процессе до совпадения их содержимого на двух последних итерациях.

На базе метода распространения ограничений компанией ILOG создан программный комплекс оптимизации и синтеза проектных решений, состоящий из подсистем Solver, Configurator, Scheduler и др.

Переборные методы

К точным методам решения задач структурного синтеза относятся *методы полного перебора*, которые на современном этапе могут быть решены для задач разной размерности в зависимости от выбора аппаратной платформы [15]. При увеличении размера задачи (прежде всего размерности математического описания СУ) целесообразнее для повышения экономичности решения задачи структурного синтеза СУ перейти к использованию методов направленного или сокращенного перебора.

Методы сокращенного перебора – приближенные методы.

Например, метод ветвей и границ в случае точного вычисления нижних границ относится к точным методам решения задач выбора и потому в неблагоприятных ситуациях может приводить к экспоненциальной временной сложности. Однако метод часто используют как приближенный, поскольку можно применять приближенные алгоритмы вычисления нижних границ.

Эволюционные методы (ЭМ) предназначены для поиска предпочтительных решений и основаны на статистическом подходе к исследованию ситуаций и итерационном приближении к искомому состоянию систем. В отличие от точных методов математического программирования ЭМ позволяют находить решения, близкие к оптимальным, за приемлемое время. В отличие от известных эвристических методов оптимизации они характеризуются существенно меньшей зависимостью от особенностей приложения (т. е. более универсальны), и в большинстве случаев обеспечивают лучшую степень приближения к оптимальному решению. Универсальность ЭМ определяется также применимостью к задачам с неметризуемым пространством управляемых переменных (т. е. среди управляемых переменных могут быть и лингвистические).

Эволюционные вычисления составляют один из разделов искусственного интеллекта. При построении систем ИИ по данному подходу основное внимание уделяется построению начальной модели и правилам, по которым она может изменяться (эволюционировать). Причем модель может быть составлена по самым различным методам, например, это может быть и нейронная сеть, и набор логических правил. К основным эволюционным методам относятся методы отжига, генетические, поведения «толпы» (PSO), колонии муравьев (ACO), генетического программирования.

В *методе отжига* (Simulated Annealing) имитируется процесс минимизации потенциальной энергии тепла во время отжига деталей. В текущей точке поиска происходит изменение некоторых управляемых параметров. Новая точка принимается всегда при улучшении целевой функции и лишь с некоторой вероятностью при ее ухудшении.

В *методе PSO* (Particles Swarm Optimization) имитируется поведение множества агентов, стремящихся согласовать свое состояние с состоянием наилучшего агента.

Метод колонии муравьев (ACO) основан на имитации поведения муравьев, минимизирующих длину своих маршрутов на пути от муравьиной кучи до источника пищи.

Важнейшим частным случаем ЭМ являются *генетические методы* и алгоритмы [14; 15; 16]. Генетические алгоритмы (ГА) основаны на поиске лучших решений с помощью наследования и усиления полезных свойств множества объектов определенного приложения в процессе имитации их эволюции.

Свойства объектов представлены значениями параметров, объединяемых в запись, называемую в ЭМ *хромосомой*. В ГА оперируют подмножеством хромосом, называемом популяцией. Имитация генетических принципов – вероятностный выбор родителей среди членов популяции, скрещива-

ние их хромосом, отбор потомков для включения в новые поколения объектов на основе оценки целевой функции – ведет к эволюционному улучшению значений целевой функции (функции полезности) от поколения к поколению.

Среди ЭМ находят применение также методы, которые в отличие от ГА оперируют не множеством хромосом, а единственной хромосомой. Так, *метод дискретного локального поиска* (его англоязычное название Hillclimbing) основан на случайном изменении отдельных параметров (т. е. значений полей в записи или, другими словами, значений генов в хромосоме). Такие изменения называют мутациями. После очередной мутации оценивают значение критерия оптимальности F (Fitness Function) и результат мутации сохраняется в хромосоме, только если F улучшился. При «моделировании отжига» результат мутации сохраняется с некоторой вероятностью, зависящей от полученного значения F .

Генетические алгоритмы

Простой генетический алгоритм. Для применения ГА необходимо:

1) выделить совокупность свойств объекта, характеризующихся внутренними параметрами и влияющих на его полезность, т. е. выделить множество управляемых параметров $X=(x_1, x_2, \dots, x_n)$, среди x_i могут быть величины различных типов (real, integer, Boolean, enumeration). Наличие нечисловых величин (enumeration) обуславливает возможность решения задач не только параметрической, но и структурной оптимизации;

2) сформулировать количественную оценку полезности вариантов объекта – функцию полезности F . Если в исходном виде задача многокритериальна, то такая формулировка означает выбор скалярного (обобщенного) критерия;

3) разработать математическую модель объекта, представляющую собой алгоритм вычисления F для заданного вектора X ;

4) представить вектор X в форме хромосомы – записи следующего вида (рис. 13.3).

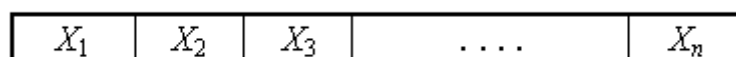


Рис. 13.3

В ГА используется следующая терминология:

ген – управляемый параметр;

аллель – значение гена;

локус (позиция) – позиция, занимаемая геном в хромосоме;

генотип – экземпляр хромосомы, генотип представляет совокупность внутренних параметров проектируемого с помощью ГА объекта;

генофонд – множество всех возможных генотипов;
 функция полезности (приспособленности) F – целевая функция;
 фенотип – совокупность значений критериев, получаемых после декодирования хромосомы, под фенотипом часто понимают совокупность выходных параметров синтезируемого с помощью ГА объекта.

Вычислительный процесс начинается с генерации исходного поколения – множества, включающего N хромосом, N – размер популяции. Генерация выполняется случайным выбором аллелей каждого гена.

Далее организуется циклический процесс смены поколений:

```
for (k=0; k<G; k++)
{ for (j=0; j<N; j++)
  { Выбор родительской пары хромосом;
    Кроссовер;
    Мутации;
    Оценка функции полезности F потомков;
    Селекция;
  }
  Замена текущего поколения новым;
}
```

Для каждого витка внешнего цикла генетического алгоритма выполняется внутренний цикл, на котором формируются экземпляры нового (следующего за текущим) поколения. Во внутреннем цикле повторяются операторы выбора родителей, кроссовера родительских хромосом, мутации, оценки приспособленности потомков, селекции хромосом для включения в очередное поколение.

Рассмотрим алгоритмы выполнения операторов в простом генетическом алгоритме.

1. *Выбор родителей.* Этот оператор имитирует естественный отбор, если отбор в родительскую пару хромосом с лучшими значениями критерия оптимальности (функции полезности) F более вероятен. Например, пусть F требуется минимизировать. Тогда вероятность P_i выбора родителя с хромосомой C_i можно рассчитать по формуле

$$P_i = \frac{F_{\max} - F_i}{\sum_{j=1}^N F_{\max} - F_j}, \quad (13.2)$$

где F_{\max} – наихудшее значение целевой функции F среди экземпляров (членов) текущего поколения, F_i – значение целевой функции i -го экземпляра.

Правило (13.2) называют правилом колеса рулетки. Если в колесе рулетки выделить секторы, пропорциональные значениям $F_{\max} - F_i$, то вероятности попадания в них суть P_i , определяемые в соответствии с (13.2).

2. *Кроссовер* (скрещивание). Кроссовер, иногда называемый кроссинговером, заключается в передаче участков генов от родителей к потомкам. При простом (одноточечном) кроссовере хромосомы родителей разрываются в некоторой позиции, одинаковой для обоих родителей, выбор места разрыва равновероятен, далее происходит рекомбинация образующихся частей родительских хромосом, где разрыв подразумевается между пятым и шестым локусами.

3. *Мутации*. Оператор мутации выполняется с некоторой вероятностью P_m , т. е. с вероятностью P_m происходит замена аллеля случайным значением, выбираемым с равной вероятностью в области определения гена. Именно благодаря мутациям расширяется область генетического поиска.

4. *Селекция*. После каждого акта генерации пары потомков в новое поколение включается лучший экземпляр пары.

Внутренний цикл заканчивается, когда число экземпляров нового поколения станет равным N . Количество повторений G внешнего цикла чаще всего определяется автоматически по появлению признаков вырождения (стагнации) популяции, но с условием не превышения заданного лимита машинного времени.

Адаптивные генетические алгоритмы. Очевидно, что эффективность генетического поиска зависит от типов используемых в ГА операторов и других параметров алгоритма. К числу таких параметров можно отнести вероятности кроссовера и мутаций, вероятности выбора тех или иных эвристик в методе комбинирования эвристик, число точек разрыва хромосом при кроссовере и др. Как правило, генетические операторы и параметры определяются пользователем априорно, однако при этом велика опасность использования неудачных вариантов и, как следствие, значительное снижение эффективности генетического поиска. Кроме того, на разных этапах генетического поиска те или иные операторы могут быть эффективными в различной степени. Выход из ситуации – применение адаптивных генетических алгоритмов.

Известен ряд работ, посвященных адаптивным ГА. Идея адаптации применительно к ГА высказывалась в работе [1]. Наибольшее внимание уделялось алгоритмам с использованием адаптации к такому параметру, как вероятность мутации. Так, в работе [2] рассмотрена стратегия, при которой вероятность мутации изменяется с ходом эволюции и для каждой хромосомы вероятность мутации настраивается индивидуально.

Метагенетический алгоритм с использованием генетического поиска для решения как основной задачи, так и вспомогательной задачи настройки параметров адаптации предложен в [3]. Ниже приведено его описание в виде псевдокода.

```
GENETIC ALGORITHM STEP()  
begin  
for i = 1 to n {  
select parent1 and parent2 from population P1;  
offspring = crossover(parent1, parent2)  
mutation(offspring)  
if suited(offspring) then  
replace(P1, offspring);  
}  
end  
META-GENETIC ALGORITHM  
begin  
create initial population P1;  
create initial population P2;  
do{  
for i = 1 to n {  
select parent1 and parent2 from P2;  
offspring = crossover(parent1, parent2)  
mutation(offspring)  
evaluate(offspring)  
add offspring tmpPop  
}  
meta-population = tmpPop;  
}  
GENETIC ALGORITHM STEP(select answer from P2)  
} until stopping condition  
//report the best answer;  
end.
```

Рассматриваются два множества хромосом. В генах хромосом первого множества ($P1$) хранятся номера эвристик для выполнения очередного шага синтеза решения. Вторая популяция ($P2$) состоит из хромосом, в генах которых хранятся конфигурационные параметры для основного генетического алгоритма.

Каждая хромосома из популяции $P2$ оценивается следующим образом:

SIMPLE_STEP ($P2(ti)$)

begin

$P2(ti) \rightarrow \text{select parent1 and parent2 from population } P1;$

$\text{offspring} = P2(ti) \rightarrow \text{crossover}(\text{parent1}, \text{parent2})$

$P2(ti) \rightarrow \text{mutation}(\text{offspring})$

$\text{return } P2(ti) \rightarrow \text{evaluate}(\text{offspring})$

end

Для каждой конфигурации (для каждой хромосомы популяции $P2(ti)$) выполняется процедура *SIMPLE_STEP*, в которой из популяции $P1$ выбираются ($P2(ti) \rightarrow \text{select}$) родители. В результате их скрещивания ($P2(ti) \rightarrow \text{crossover}$) появляется потомок (*offspring*) – новая хромосома, которая впоследствии мутируется ($P2(ti) \rightarrow \text{mutation}$). Оценка потомка также является оценкой для данной конфигурации $P2(ti)$.

После очередной генерации метагенетического алгоритма с помощью метода турнирной селекции из популяции $P2$ выбирается хромосома, и с данной конфигурацией формируется очередное поколение с помощью основного генетического алгоритма.

В Simple GA начальная популяция генерируется случайным образом. Впоследствии сгенерированная популяция оценивается, и из нее n раз выбираются два решения, от скрещивания которых генерируются два потомка. Созданные индивидуумы подвергаются мутации с заданной вероятностью.

Этот цикл повторяется до тех пор, пока новая популяция не заполнится. Когда новая популяция заполняется, старая может быть удалена с сохранением наилучших решений.

Метод комбинирования эвристик. Возможны два подхода к формированию хромосом (кодировке решений) при использовании генетических методов для структурного синтеза.

Первый из них основан на использовании в качестве генов проектных параметров. Например, в задаче размещения элементов на плате локусы соответствуют посадочным местам на плате, а генами являются номера (имена) микросхем. Другими словами, значением k -го гена будет номер микросхемы в k -й позиции. Область значений вещественных параметров аппроксимируется множеством целых чисел. Часто управляемые параметры кодируют двоичными числами, тогда хромосомы представляются битовыми строками, а параметры – частями этих строк.

Во втором подходе генами являются не сами проектные параметры, а номера эвристик, используемых для определения проектных параметров. Так, для задачи размещения можно применять несколько эвристик. По одной

из них в очередное посадочное место нужно помещать микросхему, имеющую наибольшее число связей с уже размещенными микросхемами, по другой – микросхему с минимальным числом связей с еще не размещенными микросхемами и т. д. Генетический поиск в этом случае есть поиск последовательности эвристик, обеспечивающей оптимальный вариант размещения.

Второй подход относится к генетическому программированию и реализован в методе, названном *методом комбинирования эвристик* (или мультиметодным генетическим алгоритмом). Этот метод оказывается предпочтительным во многих случаях. Например, в задачах синтеза расписаний распределяется заданное множество работ во времени и между обслуживающими устройствами – серверами, т. е. проектными параметрами для каждой работы будут номер сервера и порядковый номер в очереди на обслуживание. Пусть N – число работ, M – число серверов. Если гены соответствуют номерам работ, то в первом подходе в хромосоме нужно иметь $2N$ генов и общее число отличающихся друг от друга хромосом W заметно превышает наибольшее из чисел $N!$ и M^N .

Согласно методу комбинирования эвристик число генов в хромосоме в два раза меньше, чем в первом подходе, и равно N . Поэтому если число используемых эвристик равно K , то мощность множества возможных хромосом уже несравнимо меньше, а именно $W = K^N$.

Очевидно, что меньший размер хромосомы ведет к лучшей вычислительной эффективности, а меньшее значение W позволяет быстрее найти окрестности искомого экстремума. Кроме того, в методе комбинирования эвристик все хромосомы, генерируемые при кроссовере, будут допустимыми. В то же время при применении обычных генетических методов необходимо использовать процедуры типа РМХ для корректировки генов, относящихся к номерам в очереди на обслуживание, что также снижает эффективность поиска.

ТЕМА 6. АВТОМАТИЗАЦИЯ КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ ССУ

Лекция 14. Математическое моделирование ССУ при конструировании

Автоматизация конструкторского проектирования в рамках комплексной автоматизации этапа проектирования ССУ: основные понятия. Модели коммутационных схем. Математические модели для решения задач конструкторского проектирования. Модель квадратичного назначения. Метод ветвей и границ при конструировании. Непрерывные модели конструирования.

Автоматизация конструкторского проектирования в рамках комплексной автоматизации этапа проектирования ССУ: основные понятия

Конструирование СУ представляет собой часть процесса проектирования, цель которого состоит в разработке конструкторской документации для изготовления изделий на производственном предприятии. Этот этап начинается с передачи принципиальных схем СУ, их устройств и элементов, разработанных на этапе функционального проектирования с помощью САЕ – части интегрированных САПР. Эта передача осуществляется довольно просто в рамках единой информационной модели СУ при использовании сквозного проектирования на основе интегрированных САПР [6].

Математическими моделями описания конструкций СУ, устройств и элементов СУ являются структурные топологические (2D) и геометрические (3D) модели, которые образуют вместе с методами конструирования математическое обеспечение САД-систем в рамках интегрированных САПР.

Учитывая разнородность устройств и элементов СУ (механических, электромеханических, гидравлических, электронных и т. д.) рассмотреть все вопросы конструирования достаточно сложно из-за большого разнообразия математических описаний конструкций. Можно лишь выделить основные направления при конструировании СУ, считая основным содержанием этапа конструкторского проектирования определение конфигурации и размеров элементов, устройств и СУ в целом при сохранении условий работоспособности согласно ТЗ на проектирование.

Ограничимся рассмотрением математического описания моделей и методов конструирования устройств СУ на примере электронных устройств. Разработка конструкций элементов устройств СУ и систем в целом связана, прежде всего, с разработкой математического описания в виде геометриче-

ских 3D моделей на основе методов геометрического моделирования, что выходит за рамки рассмотрения данного курса. Например, конструирование корпуса ЛА, при распространении потока, обтекающего элементы корпуса ЛА при проектировании СУ по ЛА по курсу. Методы формирования конструктивов элементов аналогичны рассмотренным выше методам формирования функциональных моделей элементов СУ, среди которых ведущим является метод конечных элементов.

Алгоритмы и методы конструирования являются едиными при рассмотрении описания СУ как для уровня устройств СУ, так и для уровня системы в целом, независимо от вида структурной модели (2D или 3D).

Этап конструкторского проектирования устройств СУ определяется комплексом задач, связанных с преобразованием функциональных логических или принципиальных электрических схем в совокупность конструктивных узлов, между которыми устанавливаются необходимые пространственные, механические и электрические связи. Принципом конструирования устройств СУ является применение функционально-узлового (модульного) метода проектирования, предусматривающего выделение конструктивных узлов (модулей) различной степени сложности, находящихся в отношении соподчинённости [7].

Таким образом, конструкция электронного устройства (ЭУ) СУ представляет собой иерархическую структуру, в которой узлы низшего уровня объединяются в узлы высшего уровня. Широкому практическому распространению модульного принципа конструирования содействовали такие факторы, как развитие методов микроминиатюризации электронных компонентов, создание и использование интегральных микросхем, печатных и многослойных печатных плат.

Конструкции ЭУ, как правило, имеют иерархическую структуру, в которой конструктивы низшего уровня объединяются в конструктивы высшего уровня.

Используются следующие названия конструктивов ЭУ.

Элемент – выделяемый (проверяемый по своим техническим условиям) конструктив, поставляемый как отдельное изделие, например интегральная микросхема (ИС), гибридная интегральная схема (ГИС), транзистор, диод, резистор и другие электроэлементы.

Ячейка или *типовой элемент замены* (ТЭЗ) – это объединение элементов в едином конструктивном исполнении на основе печатной платы, в виде микроэлектронного узла или микросборки. Обычно ячейка реализует сложную функциональную схему, в большинстве случаев функционально-законченную, проверяемую по своим техническим условиям, и является съёмной при ремонте и устранении неисправностей. Соединения между элементами в ячейке, как правило, печатные.

Следующие иерархические уровни занимают конструктивы, называемые *устройствами* (панелями, субблоками), *блоками* (стойками). Каждый из конструктивов последующего уровня состоит из нескольких конструктивов предыдущего уровня, размещаемых в некоторой несущей конструкции (кар-

касе) и соединяемых между собой проводным монтажом, гибкими шлейфами или многослойными печатными объединительными платами.

Печатная плата представляет собой составную пластину, на одной или обеих поверхностях которой размещены элементы. Электрическая связь между выводами элементов обеспечивается печатными проводниками, нанесенными на поверхности отдельных диэлектрических пластин платы. Печатные платы бывают одно-, двух- и многослойными в зависимости от числа слоев формируемых систем печатных проводников [17].

Элементы, устанавливаемые на печатных платах, различаются габаритами, числом и типом выводов, их расположением. Элементы могут иметь корпуса различных конструкций или бескорпусное исполнение. Корпуса бывают с плоскими (планарными) или штыревыми выводами. Минимальное расстояние между выводами, как правило, постоянно. К элементам относятся также соединители для подключения внешних цепей к электрической схеме печатной платы. Соединители размещаются у краев печатных плат и содержат планарные или сквозные контактные площадки, к которым припаиваются металлические лепестки, соединенные с контактами вилки разъема, устанавливаемого на печатной плате.

В конструкции соединительного монтажа печатных плат выделяют следующие основные элементы: токопроводящий рисунок наружных поверхностей печатной платы (внешних слоев) и внутренних слоев проводимости, межслойные переходы, сквозные отверстия и иногда навесные токопроводящие шины. Рисунок токопроводящих слоев печатных плат состоит из контактных площадок и печатных проводников различной ширины. Различают планарные контактные площадки прямоугольной формы для монтажа планарных выводов элементов и контактные площадки металлизированных сквозных отверстий для монтажа жестких штыревых выводов элементов. Печатные проводники предназначены либо для электрического соединения элементов согласно электрической схеме соединений ячейки (имеют нерегулярную структуру и проектируются индивидуально), либо для разводки на плате цепей специального назначения – заземления, питания и др. (являются стандартными для плат определенного типоразмера и часто имеют регулярную структуру). В многослойных печатных платах цепи заземления и питания обычно размещаются в одном (внутреннем) слое и подключаются к контактным площадкам путем соединения с соответствующими металлизированными сквозными отверстиями. Межслойные переходы представляют собой контактные площадки металлизированных сквозных отверстий, имеют круглую форму и предназначены для обеспечения электрической связи печатных проводников, проведенных к нему на разных слоях платы. Неметаллизированные сквозные отверстия служат для крепления крупногабаритных элементов на плате и платы к ее каркасу.

В конструкторском проектировании выделяют три *группы задач*:

- синтез конструкций;
- контроль полученных решений;

- оформление документации конструкторской (КД) и технологической (ТД), включая выпуск носителей информации.

Основными задачами синтеза конструкций ЭУ являются следующие коммутационно-монтажные задачи:

- компоновка конструктивов i -го уровня в конструктив $(i - 1)$ -го уровня (компоновка простейших функциональных элементов логических схем И, ИЛИ, НЕ в корпуса микросхем, компоновка интегральных микросхем (ИМС) и других электрорадиоэлементах (ЭРЭ) в ячейки, ячеек в блоки и устройства);

- размещение конструктивов i -го в конструктивах $(i - 1)$ -го уровня (размещение цифровых и аналоговых бескорпусных ИМС по подложке гибридной интегральной схемы (ГИС), размещение ЭРЭ на печатной плате, размещение ячеек в блоке и т. д.);

- трассировка монтажных соединений между конструктивами на всех уровнях (соединения между кристаллами ИМС, печатный монтаж, межплатные соединения).

Модели коммутационных схем

Любая логическая или принципиальная электрическая схема устройства СУ состоит из некоторого набора базовых элементов, определенным образом связанных между собой. Элементами таких схем могут быть резисторы, конденсаторы, транзисторы, логические элементы, микросхемы и т. д. Связи в схеме соответствуют подаче электрических сигналов (дискретного или аналогового типа) на выходы (или полюса) определенных элементов.

Для описания схем удобно воспользоваться символикой теории множеств. Множество – это любая определённая совокупность объектов. Объекты, из которых состоит множество, называются его элементами. Множества как объекты могут быть элементами других множеств. Множество, элементами которого являются множества, обычно называются классом или семейством. Множество, не содержащее элементов, называется пустым.

Учитывая характер основных задач конструирования устройств СУ, можно рассматривать исходную схему как некоторое множество элементов $E = \{e_1, e_2, \dots, e_n\}$ соединенных между собой электрическими цепями из множества $V = \{v_1, v_2, \dots, v_m\}$. Назовем такое представление схемой соединений или коммутационной схемой.

Каждый элемент схемы имеет некоторое множество соединительных выводов $C = \{c_1, c_2, \dots, c_k\}$. Кроме выводов элементов, в схеме присутствуют внешние выводы C_0 , осуществляющие связь схемы с другими устройствами. Для дальнейшего удобства будем считать эти выводы принадлежащими фиктивному элементу e_0 .

Два вывода схемы будут связанными, если они объединяются одной электрической цепью (принадлежат одному эквипотенциалу).

Электрическая цепь в общем случае связывает больше двух выводов в схеме. Назовем комплексом совокупность эквипотенциальных выводов схемы, а число выводов в комплексе – размером комплекса или размером соответствующей цепи. Ясно, что существует взаимнооднозначное соответствие между множеством комплексов и множеством цепей схемы, далее обозначаемых одинаковыми символами.

Легко показать, что определенное выше отношение связности выводов, заданное на множестве всех выводов схемы $C = U \cdot C_i$, является отношением эквивалентности, поскольку для него справедливы свойства рефлексивности, симметричности и транзитивности. Отсюда следует, что семейство комплексов $V = \{v_{i,j} = 1, 2, \dots, M\}$ представляет собой разбиение множества выводов C на непересекающиеся классы эквивалентности:

$$C = \bigcup_{j=1}^M v_j \quad \begin{matrix} v_i \neq 0 \\ v_i \cap v_j = 0 \\ i \neq j \end{matrix} \quad (14.1)$$

Соотношение в содержательном плане означает тот факт, что выводы, принадлежащие различным цепям схемы, непосредственно электрически не связаны. Из (14.1) непосредственно следует равенство:

$$K = \sum_{i=0}^n k_i = \sum_{j=1}^M p_j, \quad (14.2)$$

где k_i – число выводов на элементе e_i ; p_j – размер j -й цепи; K – общее число выводов в схеме.

Два элемента схемы считаются связанными, если имеется, по крайней мере, одна электрическая цепь, содержащая выводы этих элементов. Некоторое множество элементов называется несвязанным, если любые два элемента этого множества являются несвязанными. Несвязанное множество элементов будет максимальным, если добавление к нему любого другого элемента схемы нарушает свойство несвязанности элементов.

Наряду с ранее введенным понятием комплекса, введем понятие элементного комплекса. Под элементным комплексом v'_j будем понимать подмножество элементов из $E = \{e_0, e, \dots, e_n\}$, соединенных цепью j ($j = 1, 2, \dots, M$). Если каждый элемент схемы связан, по крайней мере с одним другим элементом, что естественно для реальных схем, то $E = \bigcup_{j=1}^M v'_j$. Однако элементные комплексы могут содержать общие элементы, т. е. $v'_i \cap v'_j \neq 0$.

Число элементов в комплексе v'_j назовем размером элементарного комплекса: $p'_j = |v'_j|$, $p'_j \geq 1$. Случай $p'_j = 1$ означает, что цепь j соединяет выходы одного и того же элемента.

Отметим, что всегда $p'_j \leq p_j$ ($j = 1, 2, \dots, M$) (рис. 14.1, $a-p = 6$, $p' = 3$). Кроме того, соответствие между множеством цепей схемы и множеством элементарных комплексов не является взаимнооднозначным.

Характерной особенностью коммутационных схем, вызывающей в ряде случаев недоразумения при их описании, является присутствие в них цепей, соединяющих несколько элементов (выводов). Собственно схема в общем случае не задает конкретного способа реализации таких многоконцевых соединений: он должен быть определен в процессе решения основных задач конструирования.

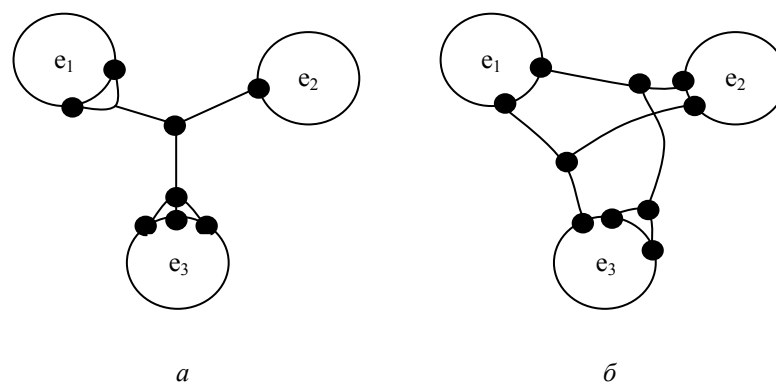


Рис. 14.1

В связи с этим будем называть конструктивную реализацию электрической цепи монтажным соединением, а непосредственное соединение двух выводов – элементарным (монтажным) соединением.

Среди различных вариантов описания коммутационных схем наибольшей общностью и наглядностью обладает описание схемы в виде графа. Такое представление широко используется при математической постановке различных оптимизационных задач конструирования и позволяет в целом ряде случаев найти адекватные задачи в теории графов и воспользоваться при разработке алгоритмов решением задач конструирования известными математическими методами.

При выборе способа описания коммутационной схемы графом необходимо учитывать особенности схемы, а также допустимую степень идеализации модели для использования в конкретной задаче конструирования: при компоновке, размещении и трассировке. Рассмотрим несколько способов описания схем графами.

Первый из них является наиболее общим и предполагает построение графа коммутационной схемы (ГКС). В отличие от обычного линейного графа, определяемого заданием ребер между названными парами вершин, в ГКС будем различать несколько типов ребер и вершин.

Введем вершины трех типов: E , C , V . Вершины E соответствуют элементам схемы, вершины C – выводам элементов, включая внешние выводы схемы, а вершины V – цепям (комплексам) схемы.

Среди ребер ГКС будем различать элементные ребра F и сигнальные ребра W .

Элементные ребра определяют принадлежность выводов из множества C элементам из множества E и задаются парами вершин $(e_i c_k)$.

Сигнальные ребра W определяют вхождение выводов из C в отдельные цепи и описываются парами вершин $(v_i c_k)$.

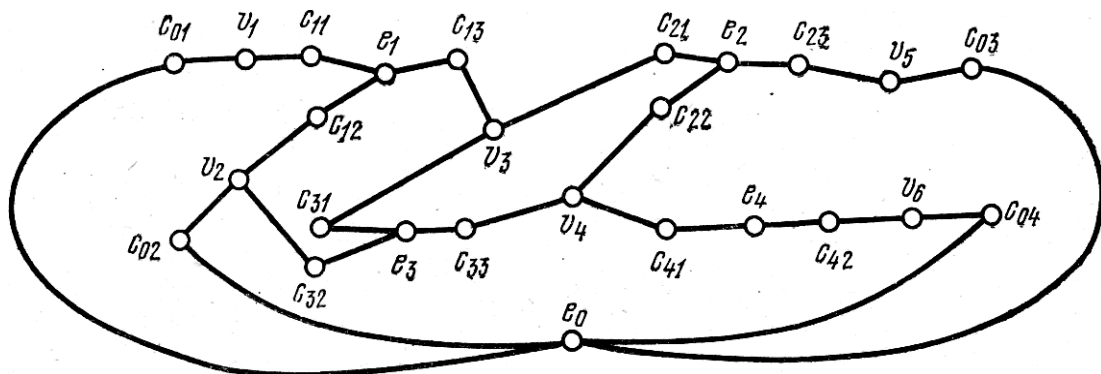


Рис. 14.2

При решении некоторых задач конструирования может оказаться необходимым дополнить введенную модель схемы. Так, иногда существенна информация об источниках и приемниках сигналов в каждой цепи, поэтому сигнальным ребрам придается соответствующая ориентация, и граф схемы становится ориентированным. В других ситуациях, например при трассировке соединений на плоскости, необходимо учитывать порядок расположения конструктивных выводов на элементах. В связи с этим в ГКС нужно уточнить порядок следования элементных ребер, что может быть достигнуто специальными средствами. В большинстве же случаев степень детализации описания коммутационной схемы (рис. 14.2), обеспечиваемая ГКС, вполне достаточна.

Известно, что произвольный неориентированный граф $G = (X, U)$ с множеством вершин X и множеством ребер U может быть задан в числовой форме матрицей инциденций $C = \|C_{ij}\|_{n \times m}$, в которой элемент $c_{ij} = 1$, если вершина X_i инцидентна ребру u_j , и $c_{ij} = 0$ в противном случае. Учитывая, что ГКС содержит вершины и ребра разных типов, его структуру удобнее описать с помощью пары матриц A и B .

Матрица A представляет цепи схемы и определяется следующим образом: $A = \|a_{ij}\|_{M \times K}$, где M – число цепей, K – число выводов в схеме; элемент $a_{ij} = 1$, если вывод C принадлежит цепи v_j , и $a_{ij} = 0$ в противном случае.

Матрица $B = ||b_{ij}||_{n \times k}$ выделяет подмножества выводов, принадлежащих отдельным элементам. Строки матрицы соответствуют элементам, а столбцы – выводам. Элемент $b_{ij} = 1$, если вывод c_j принадлежит элементу, и равен нулю в противном случае.

Модель в виде ГКС используется при задании полной информации о схеме в автоматизированном процессе конструирования. Вместе с тем при алгоритмическом решении отдельных задач конструирования удобнее пользоваться упрощенными моделями схем. Так, например, при компоновке узлов можно отождествить наборы выводов c_i с самими элементами e_i . В результате этого преобразования комплексы V_j переходят в элементные комплексы v'_j , что соответствует в ГКС «стягиванию» определенных подмножеств вершин из C в вершины из E и устранению элементных ребер F . Таким образом, получаем граф $G' = (E, V, W)$, подмножества вершин которого E и V' соответствуют элементам и элементным комплексам схемы, а множество ребер W определяет вхождение элементов в комплексы. Граф G' является двудольным графом, поскольку и E , и V' являются несвязанными множествами вершин. Полученную модель схемы будем называть графом элементных комплексов (ГЭК) (рис. 14.3).

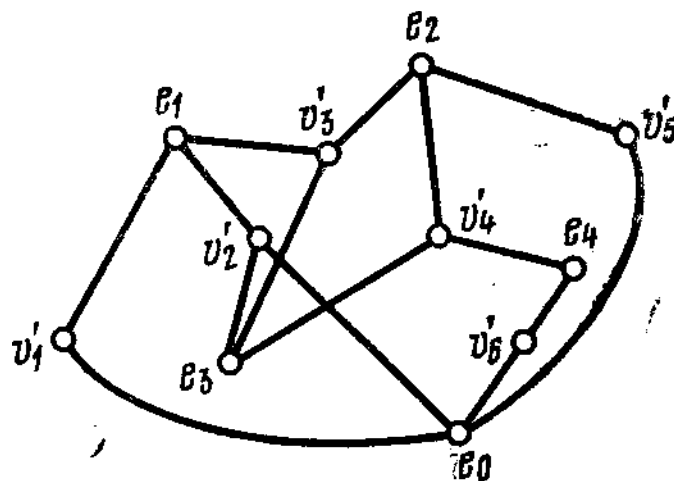


Рис. 14.3

Для описания ГЭК удобно воспользоваться матрицей комплексов Q , строки которой соответствуют элементам e_i , а столбцы – элементным комплексам v'_j . Значение $q_{ij} = 1$, если элемент e_i входит в комплекс v'_j (связан с j -й цепью), и $q_{ij} = 0$ в противном случае. Число единиц в любой строке матрицы равно числу цепей, связанных с соответствующим элементом. Число единиц в столбце равно размеру данного комплекса. Отметим, что одноэлементные комплексы ($p'_j = 1$) приводят к образованию петель на некоторых вершинах e_i в ГЭК. Такие комплексы соответствуют целям, связывающим выводы одного и того же элемента, и не оказывают влияния на решение задач компоновки и размещения. В связи с этим при задании схемы в виде ГЭК они могут быть опущены.

Заметим, что между матрицей Q и введенными ранее для описания ГКС матрицами A и B существует простая связь:

$$Q = B \cdot A', \quad (14.3)$$

где A' – транспонированная матрица A , символ означает умножение матриц с булевыми переменными 0 и 1.

Модели схемы в виде ГКС или ГЭК задают электрически связанные подмножества выводов (элементов), но оставляют свободу в определении конкретных монтажных соединений группы выводов. Напомним, что в понятии графа существен только факт присутствия ребер между отдельными вершинами, а не способ их изображения. Так, одному и тому же графу могут соответствовать различные чертежи, отличающиеся расположением вершин и геометрической формой соединяющих эти вершины ребер.

Расчет оптимальных конфигураций соединений составляет основу для разработки схем проводного и печатного монтажа. Здесь же целесообразно остановиться на некоторых общих свойствах графов монтажных соединений. Одним из таких свойств, как уже отмечалось, является связность. С другой стороны, графы соединений, как правило, не содержат циклов. Действительно, удаление любого ребра, входящего в цикл, не нарушает электрического соединения рассматриваемой группы выводов. Поэтому можно считать, что граф, соответствующий соединению группы выводов, есть связный граф без циклов, т. е. дерево.

Кратко остановимся на структуре деревьев соединений.

При проводном монтаже возможна реализация лишь элементарных соединений (c_k, c_i) типа «вывод – вывод». Таким образом, дерево соединений $T = (X, U)$ для цепи размером p состоит из набора вершин $x \in X$, являющихся выводами данной цепи ($|X| = p$), и множества ребер $u \in U$, соответствующих элементарным соединениям (c_k, c_i) , причем $|U| = p-1$.

Согласно теореме Кэли, для p вершин существует p^{p-2} различных деревьев.

При печатном (пленочном) способе выполнения соединений для осуществления необходимых электрических связей, помимо выводов, можно использовать внутренние точки проводников. В этом случае возможно образование соединений типа «вывод – проводник» и «проводник – проводник», что значительно увеличивает общее количество возможных деревьев соединений. Теперь дерево соединений $T_p = (X_p, U)$, помимо исходного набора вершин X , имеет некоторое множество дополнительных вершин: $X_p = X \cup P$, а множество U состоит из $p + |P| - 1$ ребер. Такое дерево называется деревом Штейнера.

Учет особенностей реализации соединений позволяет использовать при решении отдельных задач компоновки и размещения упрощенные модели описания схем, основанные на задании «степени связанности» элементов друг с другом.

Один из способов состоит в следующем. Подсчитаем для каждой пары элементов число связывающих их цепей. Далее построим граф $G = (E, U)$, в котором вершины e_i соответствуют элементам, а ребра u_{ij} с приписанными к ним весами $r_{ij} > 0$ – количеству цепей между элементами e_i и e_j . Полученный граф называется взвешенным графом схемы (ВГС) (рис. 14.4).

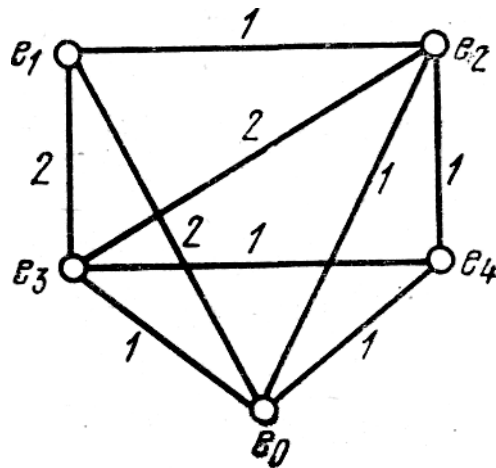


Рис. 14.4

Данный способ построения ВГС равносильно выполнению следующего преобразования в ГЭК схемы рис. 14.4. Построим для каждого элементного комплекса полный граф элементарных соединений. Очевидно, что для комплекса размером p получим $p(p - 1)/2$ соединений. Далее, для каждой пары вершин e_i и e_j введем ребро, если между ними имеется хотя бы одно элементарное соединение. Припишем ребру u_{ij} вес r_{ij} , равный числу элементарных соединений между вершинами e_i и e_j . Поскольку в данном случае веса r_{ij} целочисленные, построенный ВГС можно считать мультиграфом, в котором веса задают кратности ребер. Результат применения указанного преобразования к графу рис. 14.3 показан на рис. 14.4.

В общем случае ВГС может быть описан матрицей соединений $R = ||r_{ij}||_{n \times n}$ строки и столбцы которой соответствуют элементам схемы, а r_{ij} равен весу, приписанному соединению элементов e_i и e_j . Матрица R – симметрическая с нулевой главной диагональю ($r_{ij} = 0, i = 1, 2, \dots, n$).

Иногда удобно считать $r_{ij} = \sum_j^n r_{ij}$ ($i = 1, 2, \dots, n$) равным суммарному числу цепей, связанных с элементом e_i . При этом условии и рассмотренном способе расчета весов r_{ij} между матрицей соединений R и матрицей комплексов Q существует простая связь:

$$R = Q \times Q', \tag{14.4}$$

где Q' – транспонированная матрица Q .

$$r_{ij} = \sum_{s=1}^M q_{is} \cdot q'_{sj} = \sum_{s=1}^M q_{is} \cdot q_{js} \quad (14.5)$$

Отсюда следует, что r_{ij} равно числу цепей, связывающих элементы e_i и e_j .

При $i = j$ из (14.5) получим

$$r_{ij} = \sum_{s=1}^M q_{is} \cdot q'_{sj} = \sum_{s=1}^M q_{is} \cdot q_{js} \quad (14.6)$$

т. е. r_{ij} равно количеству цепей, связывающих элемент e_i с другими элементами.

Поскольку для соединения выводов одной цепи в действительности используются деревья соединений, описанный способ построения ВГС является относительно грубым. Поэтому в ряде алгоритмов размещения и компоновки используют весовые оценки, учитывающие приоритеты цепей и вероятности реализации отдельных соединений.

Широкий класс весовых оценок, используемых при образовании матрицы соединений R , описывается выражением

$$r_{ij} = \sum_{s=1}^M q_{is} q_{js} \cdot w_s \cdot f_s \quad (14.7)$$

где w_s – коэффициент ($0 < w_s \leq 1$), отражающий особенности s -й цепи; f_s – коэффициент учета размера цепи. Задание соединений цепи полным графом, как это было сделано выше, соответствует выбору $w_s = f_s = 1$.

Перед размещением элементов каждая цепь заменяется полным графом, однако ребро графа получает вес $2/p_s$, где p_s – размер цепи. Данная оценка определяет вероятность построения элементарного соединения между парой выводов цепи при условии равновероятного выбора любого из возможных деревьев.

Иногда оценка связности элементов по отдельной цепи выполняется с использованием величины $f_s = l/(p_s - 1)$. Эта оценка принимает значение 1 при $p_s = 2$. С ростом размера цепи «степень связи» элементов, принадлежащих одной цепи, монотонно уменьшается.

В алгоритме размещения Куртцберга $f_s = (p_s + \lambda)/p_s$, что позволяет различать вклад цепей различного размера в весовую оценку r_{ij} .

Коэффициент w_s может быть сделан относительно большим у цепей, для которых существенно время передачи сигнала.

При использовании в программах решения задач конструирования матрицы соединений R можно воспользоваться ее симметричностью.

Под связностью схемы при ее описании ВГС (матрицей R) понимается величина

$$S = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n r_{ij} = \sum_{i=1}^n \sum_{j>i}^n r_{ij} \quad (14.8)$$

При описании схемы ГЭК (матрицей Q) под связностью будем понимать суммарное число элементарных соединений:

$$S = \sum_{j=1}^M (\rho'_j - 1) = \sum_{j=1}^M \rho'_j - M, \quad (14.9)$$

где M – число цепей.

Очевидно, что когда все цепи имеют размер 2, значения связности схемы при ее описании ВГС и при описании схемы ГЭК совпадают.

В дальнейшем при рассмотрении алгоритмических методов решения основных задач конструирования используются различные модели для описания схем. Как правило, большинство из рассматриваемых алгоритмов могут при соответствующей модификации применяться для разных способов описания схем. Вместе с тем излишнее усложнение описания может привести к слишком сложным вычислительным процедурам и к большим затратам машинного времени, не оправданным с точки зрения конечного результата. В связи с этим сделаем несколько замечаний.

Описание коммутационной схемы с помощью ГКС является наиболее полным и точным и в одной из эквивалентных ему форм входит составной частью в исходную информацию для систем автоматизированного конструирования. Данная модель непосредственно используется при алгоритмическом решении задач трассировки соединений на ЭВМ. На основании этой же модели, как было показано выше, могут быть получены другие, более простые модели описания схемы. Модель схемы в виде ГКС должна иметь преимущества при решении ряда задач размещения, в которых игнорирование размеров элементов и положения их выводов является необоснованным. Сюда, прежде всего, относятся задачи размещения разногабаритных элементов, задачи размещения, в которых среднее расстояние между элементами на коммутационном поле сравнимо с размерами самих элементов, задачи, в которых внешние выводы существенно распределены по периферии схемы, и др. В этих случаях отождествление элементов с геометрическими точками (ГЭК или ВГС) приводит к значительным погрешностям уже на этапе постановки задачи оптимизации по тому или иному критерию (суммарной длине соединений, числу пересечений и т. п.). В связи с этим использование более сложных и точных алгоритмов в таких ситуациях бессмысленно. Для решения задачи размещения целесообразно использовать упрощенные модели описания схем (ГЭК или ВГС) для выбора начального размещения и более точные (ГЭК и ГКС) при получении окончательного размещения.

Для задач компоновки элементов в узлы, учитывая специфику используемых при ее решении критериев и ограничений (числа межузловых связей, числа узлов и т. д.), информация о точном расположении выводов на элементах и узлах практически несущественна. Поэтому модели

описания схем в виде ГЭК или ВГС в данном случае наиболее естественны, причем с точки зрения адекватности модели физическому содержанию задачи преимущество имеет представление схемы с помощью ГЭК.

С точки зрения реализации вычислительных процедур, наиболее просто описание схемы с помощью ВГС (матрицы соединений R).

Математические модели для решения задач конструкторского проектирования

Пусть конструктивному узлу сопоставимо некоторое непустое подмножество элементов схемы $T = \{e_1, e_2, \dots, e_j, \dots, e_k\}$.

Произвольный вариант компоновки представляет собой разбиение множества элементов схемы E на непересекающиеся подмножества $T_1, T_2, \dots, T_s, \dots, T_\gamma$:

$$T_s = \{e_{s_1}, \dots, e_{s_i}, \dots, e_{s_k}\},$$

$$T_s \cap T_l = \emptyset; s \neq l; s, l = 1, 2, \dots, \gamma, \quad (14.10)$$

$$\bigcup_{s=1}^{\gamma} T_s = E.$$

Разбиение (14.10) определяет $\gamma + 1$ новых схем соединений: γ схем внутриузловых соединений и одну схему междуузловых соединений.

Схема внутриузловых соединений для некоторого узла T_s ($s = 1, 2, \dots, \gamma$) задаёт соединения между элементами этого узла и на нее распространяются все основные понятия.

Дадим формальное определение схемы междуузловых соединений. Будем считать, что схема задана графом элементных комплексов $G = (E, V, W)$ или взвешенным графом схемы $G = (E, U)$. Разбиению (14.10) соответствует разбиение множества вершин E графа схемы на непересекающиеся подмножества.

Определим граф междуузловых соединений G с помощью следующего преобразования «склеивания»:

1) Каждое подмножество $T_s \subset E$ заменим одной вершиной t_s ($s = 1, 2, \dots, \gamma$).

2) Все ребра, инцидентные вершинам из T_s , считаем инцидентными вершине t_s ($s = 1, 2, \dots, \gamma$).

При представлении схемы графом $G = (E, U)$ получим граф $G' = (T, U')$, в котором множество вершин $T = \{t_s\}$ соответствует узлам схемы, а множество ребер $U' = \{u'_{sl}\}$ с приписанным к ним весом r'_{sl} определяет связи между соответствующими узлами.

Очевидно, что

$$r'_{sl} = \sum_{e_i \in T_s} \sum_{e_j \in T_l} r_{ij}, \quad (14.11)$$

где r_{sl} – элемент матрицы соединений R исходной схемы.

По аналогии с (14.8) под связностью схемы между узловых соединений в этом случае будем понимать величину

$$S' = \frac{1}{2} \sum_{s=1}^{\gamma} \sum_{i=1}^{\gamma} r'_{si}, \quad (14.12)$$

где r'_{si} определяется выражением (14.11).

Таким образом, при задании ВГС схема между узловых соединений описывается графом $G' = (T, U')$, которому соответствует матрица соединений $R' = \|r'_{sl}\|_{\gamma \times \gamma}$. Матрица R' – симметрическая, с нулевой главной диагональю. Элемент матрицы r'_{sl} , вычисляется из (14.11).

Пусть теперь исходная схема задана графом $G = (E, V, W)$. После применения к нему указанного выше преобразования получим граф между узловых соединений $G' = (T, V', W')$, в котором множество вершин T соответствует узлам схемы, а множество вершин V' и рёбер W' задают междуузловые соединения (комплексы) (рис. 14.5, а).

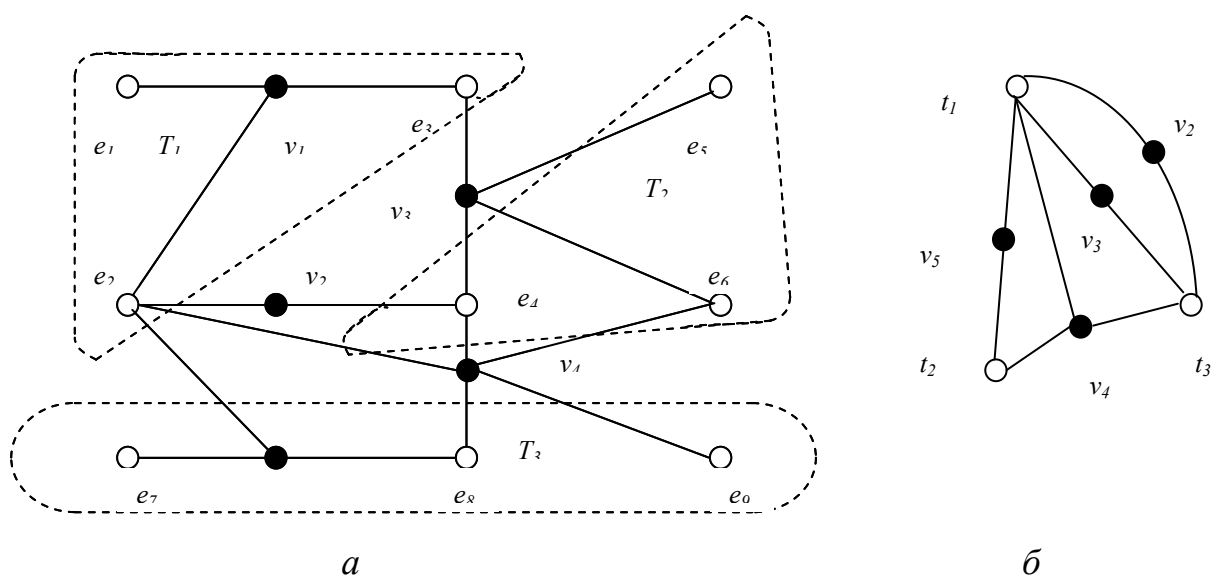


Рис. 14.5

Определение междуузловых комплексов аналогично определению элементного комплекса: узлы играют роль элементов. На рис.14.5 показано преобразование схемы при разбиении на узлы $T_1 = \{e_1, e_2, e_3\}$, $T_2 = \{e_4, e_5, e_6\}$, $T_3 = \{e_7, e_8, e_9\}$.

Следует отметить, что некоторые цепи становятся внутренними, другие переходят в схему межузловых соединений. В общем случае размеры межузловых цепей меньше размеров соответствующих цепей исходной схемы: $p(v'_j) \leq p(v_j)$. Например, $p(v_4) = 5$, а $p(v'_4) = 3$.

По аналогии с (14.9) связность схемы межузловых соединений определится выражением

$$S' = \sum_{j=1}^F (p'_j - 1) = \sum_{j=1}^F p'_j - F, \quad (14.13)$$

в котором F – количество межузловых цепей; p'_j – размер цепи.

Граф межузловых соединений может быть описан матрицей комплексов $Q' = \|q'_{ij}\|_{\gamma \times F}$, строки которой соответствуют узлам, а столбцы – межузловым цепям. Элемент матрицы $q'_{ij} = 1$, если узел T_i связан цепью v_j с другим узлом, и $q'_{ij} = 0$ в противном случае.

Решение задач компоновки связано с расчетом таких характеристик, как число внутриузловых и межузловых соединений, число внешних соединений отдельных узлов и т. д. При использовании ВГС для представления схемы расчет указанных характеристик не вызывает затруднений и может быть легко выполнен по матрице соединений R . В дополнение к (14.11) и (14.12) приведем выражение для расчета числа внешних соединений узла T_s

$$v_s = \sum_{\substack{l=1 \\ l \neq s}}^{\gamma} r'_{sl}, \quad (14.14)$$

где r'_{sl} определяется выражением (14.11).

Использование ГЭК для описания исходной схемы в наибольшей степени отвечает специфике задачи компоновки. В этом случае для постановки задачи и реализации соответствующих алгоритмов удобно использовать аппарат алгебры множеств.

Сопоставим элементу схемы e_i множество связанных с ним цепей: $J_i = \{j_1, \dots, j_k, \dots, j_{m_i}\}$.

Пусть необходимо определить число общих цепей для двух элементов – e_i и e_s . Очевидно, оно равно числу цепей, входящих одновременно в множества J_i и J_s , т. е. числу цепей в пересечении этих множеств: $J_{is} = J_i \cap J_s$. Например, если $J_{is} = \{1, 2, 3, 5, 7\}$, а $J_s = \{2, 4, 7\}$, то общие цепи определяются множеством $J_{is} = \{2, 7\}$.

Операция объединения множеств J_i и J_s определяет множество цепей $J_{i \cup s} = J_i \cup J_s$, связанных по крайней мере с одним из элементов e_i , или e_s .

Операция дополнения множества J_i определяет множество цепей J'_i не связанных с элементом e_i .

Рассмотрим еще одну операцию – разность множеств J_i и J_s . Она выделяет те цепи элемента e_i , которые не связаны с элементом e_s : $J_{i-s} = J_i \setminus J_s$. Известно, что разность множеств может быть выражена следующим образом: $J_{i-s} = J_i \cap \bar{J}_s$. Назовем симметрической разностью двух множеств J_i и J_s множество $J_{i \otimes s} = J_{i-s} \cup J_{s-i}$. Симметрическая разность в данном случае определяет множество цепей, связанных только с одним из элементов e_i или e_s . Например, если $J_i = \{1, 2, 3, 5, 7\}$, а $J_s = \{2, 4, 7\}$, то $J_{i \otimes s} = \{1, 3, 5, 4\}$.

Пусть m_i – число цепей в множестве J_i , тогда число цепей в множествах J_{is} , $J_{i \cup s}$, J_{i-s} и $J_{i \otimes s}$ определяется из выражений:

$$m_{is} = m_i + m_s - m_{i \cup s}, \quad (14.15)$$

$$m_{i \cup s} = m_i + m_s - m_{is}, \quad (14.16)$$

$$m_{i-s} = m_i - m_{is}, \quad (14.17)$$

$$m_{i \otimes s} = m_i + m_s - 2m_{is}. \quad (14.18)$$

Назовем значение m_{is} конъюнкцией элементов e_i и e_s , а значение $m_{i \otimes s}$ дизъюнкцией.

Если $R = \|r_{ij}\|_{n \times n}$ – матрица соединений, полученная заданием для каждого комплекса полного графа элементарных соединений, то $r_{ij} = m_{ij}$. В связи с этим матрицу R иногда называют матрицей конъюнкций.

По аналогии определим матрицу дизъюнкции $C = \|c_{ij}\|_{n \times n}$, строки и столбцы которой соответствуют элементам схемы, а $c_{ij} = m_{i \otimes j}$. Матрица C – симметрическая, с нулевой главной диагональю ($c_{ii} = 0, i=1, 2, \dots, n$). Элемент матрицы c_{ij} выражается через элементы матрицы R :

$$c_{ij} = \sum_{j=1}^n r_{ij} + \sum_{i=1}^n r_{ij} - 2r_{ij}. \quad (14.19)$$

Введенные операции могут быть использованы для расчета соединений между группами элементов.

Множество цепей, связанных с элементами некоторого узла $T_s = \{e_1, \dots, e_i, \dots, e_{k_s}\}$, определяется выражением

$$J = \bigcup_{i=1}^{k_s} J_i, \quad (14.20)$$

где J_i – множество цепей элемента $e_i \in T$.

Рассчитаем число внешних цепей для узла T при некотором разбиении схемы, т. е. число цепей, связывающих элементы из T с элементами, не принадлежащими узлу T . Для упрощения обозначим множество цепей некоторого узла T_s , через J_s . Цепи, связанные с элементами, не принадлежащими узлу, определяются выражением

$$J_{\bar{s}} = \bigcup_{\lambda \neq s} J_{\lambda}. \quad (14.21)$$

Тогда число внешних цепей узла T_s рассчитывается по формуле

$$v_s = |J \cap (\bigcup_{\lambda \neq s} J_{\lambda})| = |\bigcup_{\lambda \neq s} J_{s\lambda}|, \quad (14.22)$$

где J_{sb} – пересечение множеств цепей узлов T_s и T_{λ} .

С помощью известного в комбинаторике принципа включения и исключения, позволяющего определить число элементов в множестве, образованном объединением заданных множеств, формулу (14.22) можно привести к следующему виду:

$$v_s = \sum_{\lambda=1, \lambda \neq s}^{\gamma} (-1)^{\lambda+1} \sum_{l_1, \dots, l_k, \dots, l_{\lambda}} |J_{sl_1 l_2 \dots l_{\lambda}}|, \quad (14.23)$$

где $J_{sl_1 l_2 \dots l_{\lambda}}$ – пересечение множеств цепей узлов $T_s, T_{l_1}, \dots, T_{l_{\lambda}}$ суммирование производится по всем комбинациям размерностью $2, 3, \dots, \gamma$.

Проиллюстрируем расчет по (14.23) для узла T_2 : $|J_{21}| = 3$, $|J_{23}| = 1$, $|J_{213}| = 1$, отсюда $v_2 = 3 + 1 - 1 = 3$.

Заметим, что когда $p_j = 2$, т. е. схема задана элементарными соединениями, все пересечения множеств в (14.23) при $\lambda \geq 2$ будут пусты, а (14.23) сводится к (14.14). Кроме того, для связности схемы (учитывая (14.14)), имеет место соотношение

$$S' = \frac{1}{2} \sum_{s=1}^{\gamma} v_s. \quad (14.24)$$

При задании ГЭК аналогичная формула (14.24) принимает вид

$$S' = \sum_{s=1}^{\gamma} v_s - F. \quad (14.25)$$

Учитывая, что исходные M цепей схемы при компоновке распределяются на внутриузловые цепи и межузловые цепи, можно представить выражение для связности схемы и в таком виде:

$$S^i = \sum_{s=1}^{\gamma} |J_s| - M. \quad (14.26)$$

Модель квадратичного назначения

Пусть даны элементы e_1, e_2, \dots, e_n и для каждой пары элементов заданы весовые коэффициенты $r_{ij} (i, j = 1, 2, \dots, n)$, определяющие «степень связи» элементов друг с другом. Таким образом, считаем, что схема задана матрицей соединений $R = \|r_{ij}\|_{n \times n}$.

Пусть также имеется некоторый фиксированный набор позиций для размещения элементов $l_1, l_2, \dots, l_m (m \geq n)$. Будем полагать, что $m = n$. Если $m > n$, то можно ввести $m - n$ фиктивных элементов, не связанных с остальными. Определим расстояние d_{ij} между парами позиций. В любом случае, если на коммутационном поле фиксированы позиции для размещения элементов, то можно задать матрицу соединений $D = \|d_{ij}\|_{n \times n}$, в которой элемент d_{ij} равен расстоянию между центрами позиций l_i и l_j . Очевидно, что матрица D – симметрическая с нулевой диагональю ($d_{ii} = 0, i = 1, 2, \dots, n$).

Произвольное размещение элементов в позициях представляет собой некоторую перестановку $p = p(1), \dots, p(i), \dots, p(n)$, где $p(i)$ задает номер позиции, присвоенный i -му элементу. Таким образом, всего имеется $n!$ различных вариантов размещения элементов.

Рассмотрим задачу минимизации суммарной длины (МСВД) соединений при следующих предположениях. Соединения будем считать условно исходящими из геометрических центров элементов. Кроме того, предполагаем совпадение центров элементов и позиций. Как правило, при решении задачи размещения необходимо учитывать предварительное закрепление некоторых элементов в позициях и соединения элементов с внешними выводами. Сопоставляя внешним выводам элемент e_0 и фиксируя расположение элементов, получим упрощенное представление коммутационного поля. Очевидно, что длина соединений между элементами e_i и e_j оценивается величиной $r_{ij} d_{p(i)p(j)}$. Обозначим через L_s множество всех фиксированных элементов, включая элемент e_0 ; тогда суммарная взвешенная длина соединений элемента e_i с элементами из L_s оценивается по формуле

$$a_{ip(i)} = \sum_{s \in L_s} r_{ij} d_{p(i)s}, \quad (14.27)$$

где $d_{p(i)s}$ – расстояние между элементом e_i , находящимся в позиции $p(i)$, и элементом e_s .

Учитывая сказанное, а также симметричность матриц R и D , запишем выражение для суммарной взвешенной длины соединений при произвольном размещении:

$$F(p) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n r_{ij} d_{p(i)p(j)} + \sum_{i=1}^n a_{ip(i)}. \quad (14.28)$$

Таким образом, задача размещения по критерию МСВД соединений состоит в минимизации функционала (14.28) на множестве перестановок p . Данная задача является вариантом общей математической модели, впервые рассмотренным Купмансом и Бекманом и получившим название задачи квадратичного назначения.

Определим матрицу переменных $X = \|x_{ij}\|_{n \times n}$, строки которой соответствуют элементам, а столбцы – позициям. Переменная $x_{ij} = 1$, если элемент e_i находится в l_j позиции, и $x_{ij} = 0$ в противном случае. Легко установить взаимно-однозначное между матрицей X и некоторой перестановкой элементов в позициях. Поэтому X называют перестановочной матрицей или матрицей назначений. Очевидно, что матрица X содержит в каждом столбце j и в каждой строке i одну единицу:

$$\sum_{i=1}^n x_{ij} = 1, i = 1, 2, \dots, n, \quad (14.29)$$

$$\sum_{j=1}^n x_{ij} = 1, j = 1, 2, \dots, n. \quad (14.30)$$

С учётом введённых обозначений задачу размещения можно записать в форме обобщённой задачи квадратичного назначения: найти матрицу X_0 , для которой

$$F(X_0) = \min \left(\frac{1}{2} \sum_{i,j=1}^n \sum_{s=1}^n r_{ij} d_{ks} x_{ik} x_{js} + \sum_{i,k=1}^n a_{ik} x_{ik} \right). \quad (14.31)$$

Отметим, что линейный член в (14.31) представляет собой функционал обычной (линейной) задачи о назначении. Последняя формулируется следующим образом. Пусть задана некоторая матрица $A = \|a_{ij}\|_{n \times n}$, строки ее соответствуют некоторым объектам, а столбцы – местам их назначения. Элемент a_{ij} оценивает некоторую условную стоимость назначения объекта i на место j . Задача состоит в нахождении такого назначения объектов по местам

(без повторений), при котором суммарная стоимость всех назначений минимальна. В другой терминологии задача сводится к минимизации

$$\sum_{i=1}^n a_{ip(i)} \quad (14.32)$$

на множестве перестановок $P = \{p\}$, $A = \|a_{ij}\|_{n \times n}$ – заданная матрица коэффициентов.

Отметим, что к задаче линейного назначения сводится ряд частных задач оптимизации, возникающих при размещении элементов и при распределении их выводов. Из (14.28) следует, что задача размещения несвязанного множества элементов эквивалентна задаче линейного назначения.

Метод ветвей и границ при конструировании

Основная идея метода ветвей и границ состоит в разбиении всего множества допустимых решений задачи на некоторые подмножества, внутри которых осуществляется упорядоченный просмотр решений с целью выбора оптимального. Для всех решений, входящих в выделенные подмножества, вычисляется нижняя граница минимального значения целевой функции. Как только нижняя граница становится больше значения целевой функции для наилучшего из ранее известных решений, то подмножество решений, соответствующее этой границе, исключается из исходной области решений. Это обеспечивает сокращение перебора. Процесс поиска, сопровождаемый разбиением поля решений и вычислением нижних границ, продолжается до тех пор, пока не будут исключены все решения, кроме оптимального.

Различия модификации общего метода применительно к задаче размещения (квадратичного назначения) отличаются способами расчёта нижних границ функционала (14.28) и способами разбиения поля решений.

Пусть имеется множество элементов $e_1, \dots, e_i, \dots, e_n$ и множество позиций $l_1, \dots, l_i, \dots, l_n$. Для описания процесса поиска решений введём дерево решений. Ребра первого яруса дерева пусть соответствуют назначениям элемента e_1 , рёбра второго яруса – возможным вариантам назначения элемента e_2 в свободные позиции и т. д.

На рис.14.6 показана часть дерева полного дерева решений, в котором отмечены возможные назначения при условии, что элемент e_1 назначен в позицию l_j .

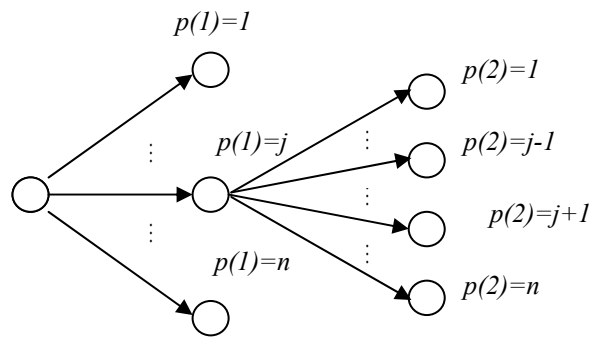


Рис. 14.6

Произвольному размещению элементов соответствует в полном дереве некоторый путь, исходящий из начальной вершины. Для каждой вершины дерева можно рассчитать нижнюю границу целевой функции для множества путей (размещений), связанных с этой вершиной. Если граница больше значения целевой функции для известного размещения, то дальнейшее продвижение по дереву в данной вершине прекращается, поскольку оно приводит к заведомо неоптимальным решениям.

Частичным размещением q будем называть назначение элементов множества $E_k = \{e_1, \dots, e_i, \dots, e_k\}$ в позиции множества $L_k = \{l_1, \dots, l_j, \dots, l_k\}$ в соответствии с правилом $j = q(i)$, $k \leq n$. Ясно, что $q \subset p$, где p – некоторое размещение n элементов, в котором k имеют размещение q . Частичному размещению q соответствует в дереве решений путь, проходящий через вершины отдельных назначений.

Если в процессе продвижения по дереву нижние границы не рассчитываются и соответственно не производится усечений дерева, то метод ветвей и границ сводится к полному перебору.

Рассмотрим способы вычисления нижних границ функционала (14.28) при частичных размещениях q . Обозначим первый член в выражении $F(p)$ через $w(p)$, а второй через $v(p)$:

$$F(p) = w(p) + v(p). \tag{14.33}$$

Нижняя граница для $F(p)$ получена суммированием нижних границ для $w(p)$ и $v(p)$. Обозначим их соответственно b_F , b_w и b_v .

Для вычисления b_w воспользуемся следующим свойством: если $r = (r_1, r_2, \dots, r_m)$ и $d = (d_1, \dots, d_2, \dots, d_m)$ – два вектора, то минимум скалярного произведения r и d , т. е. минимум $\sum_{i=1}^m r_i d_{p(i)}$ на множестве всех перестановок p , соответствует расположению составляющих вектора r в возрастающем порядке, а составляющих вектора d – в убывающем.

Нижняя граница для $F(p)$

$$b_F = b_w + b_v \quad (14.34)$$

должна быть соотнесена начальной вершине дерева решений.

Пусть теперь имеется некоторое частичное размещение q , определяющее назначение элементов из множества E_k в позиции из множества L_k . Тогда можно легко рассчитать вклад в общую длину соединений уже размещенных элементов и вычислить новую нижнюю границу соединений.

Запишем (14.28) для $F(p)$ в следующем виде:

$$F(p) = F(q) + \sum_{i \in E_k} \sum_{s \in E'_k} r_{is} d_{q(i)p(s)} + \sum_{i \in E'_k} \sum_{s \in E'_k} r_{is} d_{q(i)p(s)} + \sum_{i \in E'_k} a_{ip(i)}, \quad (14.35)$$

где E'_k – множество не размещенных элементов для частичного размещения $q \subset p$.

Структуру (14.34) можно записать в компактной форме:

$$F(p) = F(q) + w_q(p) + v_q(p) + u_q(p), \quad (14.36)$$

где $F(q)$ – длина межсоединений размещённых элементов; члены $w_q(p)$ и $v_q(p)$ аналогичны членам выражения (14.33) и соответствуют межсоединениям неразмещенных элементов; $u_q(p)$ представляет собой суммарную длину соединений размещенных элементов с неразмещенными.

Поскольку $F(q)$ полностью определяется частными размещениями q , для вычисления новой нижней границы для $F(p)$ необходимо рассчитать нижние границы для остальных членов выражения (14.36). Границы для $w_q(p)$ и $u_q(p)$ рассчитываются по матрицам R и D , а нижняя граница для $v_q(p)$ определяется решением задачи линейного назначения неразмещенных элементов E'_k в незанятые позиции L'_k .

Выбор направления поиска в дереве решений может быть организован различными способами. Как правило, очередное ветвление производится из вершины дерева с наименьшим значением нижней границы.

Далее поиск продолжается в соответствии с общей схемой метода ветвей и границ. Процесс заканчивается тогда, когда в просмотренной части дерева решений отсутствуют вершины, для которых нижняя граница меньше, чем у наилучшего из известных решений. Это решение может быть либо получено одним из рассматриваемых в данной главе приближенных алгоритмов, и в этом случае доказываемость оптимальности решения, либо определено в процессе поиска путем завершения некоторого частичного размещения q . В последнем случае в дереве решений определяется путь, соответствующий оптимальному размещению.

В заключение сделаем ряд замечаний об эффективности метода ветвей и границ для решения задачи размещения. Как уже указывалось, расчет нижних границ при частичных размещениях осуществляется с целью усече-

ния дерева решений. Естественно, что чем точнее эти границы, тем существеннее усечение. В связи с этим укажем более точный способ вычисления нижней границы для $F(p)$.

Пусть r_i – вектор-строка матрицы R без элемента $r_{ii} = 0$, а d_j – вектор-строка матрицы D без элемента $d_{jj} = 0$. При назначении элемента e_i в позицию l_j минимальная длина его связей с остальными элементами равна минимуму скалярного произведения векторов r_i и d_j . Обозначим его через $r_i \cdot d_j$. образуем матрицу $A' = \|a'_{ij}\|_{n \times n}$, в которой элемент a'_{ij} представляет оценку назначения элемента e_i в позицию l_j :

$$a'_{ij} = r_i \cdot d_j + a_{ij}, \quad (14.37)$$

где a_{ij} – элемент линейной части $v(p)$ функционала (14.28) или (14.31).

Теперь нижняя граница для $F(p)$ может быть определена решением задачи линейного назначения (14.32) для матрицы A .

Распространение этого способа на вычисление границ при частичном размещении q производится путем образования матрицы $A'_k = \|a'_{ij}\|_{(n-k) \times (n-k)}$ для неразмещенных элементов из множества E'_k .

Непрерывные модели конструкций

Снова рассмотрим выражение для суммарной взвешенной длины:

$$F = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n r_{ij} d_{ij}, \quad (14.38)$$

где d_{ij} – расстояние между точками (x_i, y_i) и (x_j, y_j) , определяющими положение элементов e_i и e_j .

Функцию F можно считать функцией $2n$ переменных: x_i ($i = 1, 2, \dots, n$) и y_i ($i = 1, 2, \dots, n$). Если $d_{ij} = d(x_i, y_i, x_j, y_j)$, рассчитывать по

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (14.39)$$

или по

$$d_{ij} = |x_i - x_j| + |y_i - y_j|, \quad (14.40)$$

то функция (14.38) является кусочно-выпуклой и потому для ее минимизации могут быть в принципе использованы аналитические методы. Для отыскания минимума приравняем частные производные по нефиксированным пе-

ременным нулю и после очевидных упрощений получим систему алгебраических уравнений:

$$\sum_{j=1}^n \frac{\partial d(x_i, y_i, x_j, y_j)}{\partial x_i} = 0, \quad i = 1, 2, \dots, n,$$

$$\sum_{j=1}^n \frac{\partial d(x_i, y_i, x_j, y_j)}{\partial y_i} = 0, \quad i = 1, 2, \dots, n.$$
(14.41)

Очевидно, что если все переменные независимы, то (14.41) имеет тривиальное решение: $x_i = \text{const}$ ($i = 1, 2, \dots, n$), $y_i = \text{const}$ ($i = 1, 2, \dots, n$). Следовательно, данная модель не может быть непосредственно использована для размещения элементов, поскольку все элементы «стягиваются» в точку. Возможно несколько путей корректировки этой модели.

Один из них состоит в предварительной фиксации положений некоторых элементов. Такими элементами могут быть внешние выводы узла или элементы, положение которых задано конструктором. Предположим, что не фиксированы положения первых l элементов, а положения $n-l$ элементов фиксированы.

Рассмотрим более подробно случай задания расстояния по формуле

$$d_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2.$$
(14.42)

Тогда (14.41) сводится к системе линейных алгебраических уравнений:

$$\sum_{j=1}^n r_{ij} (x_i - x_j) = 0 \quad i = 1, 2, \dots, l,$$

$$\sum_{j=1}^n r_{ij} (y_i - y_j) = 0 \quad i = 1, 2, \dots, l,$$
(14.43)

Запишем ее в следующем виде:

$$\sum_{j=1}^n r_{ij} x_i - \sum_{j=1}^l r_{ij} x_j = \sum_{j=l+1}^n r_{ij} x_{ij}, \quad i = 1, 2, \dots, l,$$

$$\sum_{j=1}^n r_{ij} y_i - \sum_{j=1}^l r_{ij} y_j = \sum_{j=l+1}^n r_{ij} y_{ij}, \quad i = 1, 2, \dots, l.$$
(14.44)

Определим матрицу коэффициентов $A = ||a_{ij}||_{l \times l}$, в которой $a_{ij} = \sum_{j=1}^n r_{ij}$, $a_{ij} = -r_{ij}$. Отметим, что матрица A – симметрическая ввиду симметричности матрицы соединений R . Введем векторы $b = (b_1, \dots, b_i, \dots, b_l)$ и $c = (c_1, \dots, c_i, \dots, c_l)$, составляющие которых равны:

$$b_i = \sum_{j=l+1}^n r_{ij} x_j, \quad i = 1, 2, \dots, l, \tag{14.45}$$

$$c_i = \sum_{j=l+1}^n r_{ij} y_j, \quad i = 1, 2, \dots, l.$$

Тогда (14.44) может быть представлена в стандартной матричной форме

$$\begin{aligned} AX &= b, \\ AY &= c, \end{aligned} \tag{14.46}.$$

где $X = (x_1, \dots, x_i, \dots, x_l)$ и $Y = (y_1, \dots, y_i, \dots, y_l)$ – векторы неизвестных.

Для решения (14.46) могут быть использованы известные вычислительные методы. В. М. Помазановым исследованы условия сходимости метода Гаусса – Зейделя для этой системы и даны рекомендации по предварительной фиксации некоторых элементов с целью получения достаточной равномерности расположения элементов.

Рассмотренную точечную модель размещения элементов можно расширить, задав геометрию отдельных элементов. В частности, при размещении кристаллов интегральных схем на подложке учитывается расположение выводов элементов и внешних выводов микросхем.

Аналогичные модели могут быть построены и для других способов расчета расстояний d_{ij} . Однако в этом случае соответствующие системы уравнения (14.41) оказываются значительно более сложными.

Основными недостатками рассмотренного метода являются необходимость предварительной фиксации отдельных элементов, возможность перекрытия конфигураций элементов в получаемом размещении и недостаточная равномерность их расположения. Поэтому данный метод рекомендуется применять для определения грубого начального варианта размещения, который впоследствии должен корректироваться.

Значительно большая степень равномерности может быть достигнута при введении в модель размещения ограничений на расположение элементов. В теоретическом плане представляет интерес градиентный метод Линского, в котором задача размещения сводится к задаче нелинейного программирования, и работа Холла, в которой решается задача на услов-

ный экстремум функции (14.38) при квадратичных ограничениях на расположение элементов.

Обратимся снова к (14.43). После очевидных преобразований получим следующие выражения:

$$x_i = \frac{\sum_{j=1}^n r_{ij} x_j}{\sum_{j=1}^n r_{ij}} \quad i = 1, 2, \dots, n,$$

$$y_i = \frac{\sum_{j=1}^n r_{ij} y_j}{\sum_{j=1}^n r_{ij}} \quad i = 1, 2, \dots, n. \tag{14.47}$$

Выражения (14.47) аналогичны известным формулам вычисления центра масс системы материальных точек. В ряде алгоритмов размещения используется последовательный процесс установки элемента e_i в «центр масс» связанных с ним элементов e_j , которые в этот момент считаются неподвижными. В качестве массы элемента e_j принимается величина r_{ij} – число соединений e_j с элементом e_i . Масса элемента e_i оценивается суммарным числом его соединений с остальными элементами.

Использование (14.47) для определения положения элементов представляет меньшие вычислительные трудности по сравнению с решением (14.46) и потому имеет явные преимущества при решении задач размещения с большим числом элементов n . Кроме того, при этом легче учитываются ограничения на расположение элементов и их размеры. Аналогичные модели могут быть использованы и для других способов расчета расстояний между элементами.

Отметим, что использование данного метода предполагает задание начального варианта размещения и введение средств предотвращения группировки элементов в отдельных областях коммутационного поля. Развитием описанной выше механической аналогии можно считать метод силовых функций. Он основан на сведении задачи размещения к задаче отыскания состояния равновесия системы материальных точек, на которые действуют силы притяжения и отталкивания. Непрерывные модели и механические аналогии представляют определенный интерес для решения задач размещения элементов, в которых набор позиций заранее не задан. Однако их непосредственное использование в практических задачах размещения малоэффективно, поскольку они, как правило, не обеспечивают равномерности расположения элементов на коммутационном поле.

Лекция 15. Алгоритмы автоматизации конструкторского проектирования ССУ

Алгоритмы конструкторского проектирования систем управления: конструктивные и итерационные. Алгоритмы компоновки. Последовательный алгоритм компоновки по связности. Итерационные алгоритмы компоновки. Алгоритмы размещения. Алгоритмы последовательного размещения по связности. Метод обратного размещения. Структура итерационных алгоритмов. Алгоритмы парных перестановок. Алгоритмы трассировки соединений.

Алгоритмы конструкторского проектирования систем управления: конструктивные и итерационные

Среди алгоритмов конструкторского проектирования (КП) выделяют два основных класса: конструктивные и итерационные.

Конструктивные алгоритмы формируют проектное решение за ряд последовательных шагов:

- выбирается один элемент схемы рассматриваемого уровня;
- к выбранному элементу по определенным правилам присоединяется второй;
- к полученному комплексу элементов добавляется третий и т. д.

Алгоритмы, использующие подобную методологию, называются последовательными. Алгоритмы, в которых формируются несколько групп элементов в пределах одного шага, называются параллельными.

Итерационные алгоритмы требуют задания начального приближения решения задачи КП, которое затем улучшается. Начальное решение задается инженером-проектировщиком (пользователем САПР) или является результатом работы конструктивного алгоритма.

Анализ процесса конструирования электронных устройств (ЭУ) на основе существующих методов можно проводить по-разному. На входе процесса проектирования ЭУ имеется функциональная схема, которая содержит информацию о базовых элементах (в зависимости от уровня проектирования), о связях между элементами и внешних связях проектируемого ЭУ. Кроме того, задаются технологические параметры, например, при конструировании печатного узла – размер печатной платы, разрешенные для применения в данной разработке серии микросхем, шаг сетки трассировки и т. п. В результате проектирования должен получиться рисунок трассировки.

Общей целевой функцией Φ всего процесса проектирования печатной платы (ПП) следует считать число проведенных связей. Целевая функция Φ зависит от случайных входных параметров ζ (например, числа базовых элементов, числа задействованных выводов элементов схем и т. д.), так и от переменных, статистически устойчивых для класса разработок β (размеры ПП,

серии микросхем, разрешенных для применения и т. п.). Хотя оптимальный размер ПП может быть определен с достаточной точностью как функция входных параметров, этот вопрос практического применения не нашел, поскольку изготовление и применения плат различных размеров в одной разработке связано с изменением технологической оснастки производства.

Действия над входными величинами (ζ , β) в процессе проектирования можно представить следующим образом. Обозначив совокупность используемых в САПР алгоритмов компоновки K , совокупность алгоритмов размещения P и совокупность алгоритмов трассировки T можно представить схему последовательной реализации основных алгоритмов (рис.15.1)

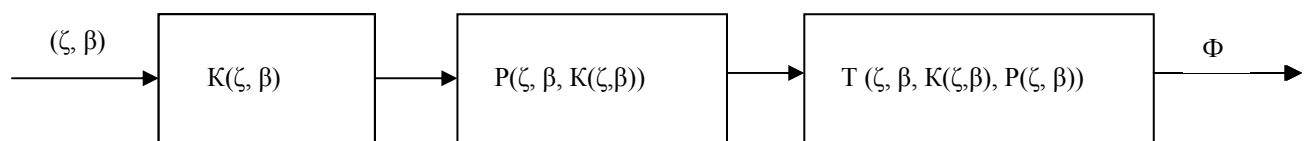


Рис.15.1

Возможно использование алгоритмов совместного решения задач компоновки и размещения, размещения и трассировки, компоновки и трассировки (при этом трассировка выполняется на условном коммутационном поле).

Общая задача проектирования формулируется следующим образом: имея заданные условия β и случайные воздействия ζ , найти такую совокупность алгоритмов и критериев $\{K, P, T\}$, которая обеспечивала бы получение максимального конечного значения целевой функции Φ . Таким образом, имеется задача о выборе и принятии решения в условиях неопределенности. Оптимизация решения на каждом шаге отдельно не всегда дает в сумме оптимальное решение, особенно, если на промежуточных этапах слабо учитывается конечный критерий. Указанный недостаток является причиной поиска связей между основными этапами КП.

Реализованные алгоритмы K, P, T в составе САПР должны симитировать действия конструктора, который решает одновременно все три задачи с оптимизацией. Это требует создания обширного математического обеспечения САПР, позволяющего варьировать различными алгоритмами. Полная алгоритмизация действий конструктора затруднительна, поэтому задача решается поэтапно, через выстраивание преемственных критериев всех этапов, поэтому при разработке САПР ставятся упрощенные задачи, т. е. на каждом шаге принятия решений приближаться к оптимальной траектории Φ . Очевидно, что само выделение трех этапов КП (компоновки, размещения и трассировки) направлено на снижение размерности общей задачи. Декомпозиция проводится следующим образом: сначала осуществляется компоновка путем оптимальной группировки функциональных узлов, затем размещение компонентов выделенных узлов в их монтажном пространстве с учетом критериев, отражающих оптимальность последующего этапа, – трассировки соединений.

Алгоритмы компоновки

Задачи компоновки можно представить в виде сложных комбинаторных моделей, для которых в большинстве случаев отсутствуют эффективные точные методы решения даже при использовании ЭВМ. Тем не менее анализ таких моделей представляет не только методический, но и практический интерес, поскольку в процессе формализации задачи часто вырабатываются эффективные приближенные алгоритмы ее решения.

Разнообразие и зачастую противоречивость требований, возникающих при решении задач компоновки, затрудняет разработку единой алгоритмической методики компоновки электронных устройств. Некоторые оценки приводимых ниже алгоритмов будут даны по мере их изложения, а также в конце главы.

Для двух основных классов задач компоновки можно выделить алгоритмы компоновки конструктивных узлов и алгоритмы компоновки типовых узлов (ячеек) [рис.15.2](#). Алгоритмы первой группы можно классифицировать по критериям оптимизации, по ограничениям на формирование узлов или по структуре вычислительной процедуры.

Основными критериями оптимизации являются минимум числа узлов, минимум числа межузловых соединений, а ограничениями – количество элементов в узле и число внешних выводов на узле. Иногда, например при разработке заказных БИС, число внешних выводов на узлах может стать основным критерием компоновки. В качестве основного ограничения при компоновке БИС принимается площадь, занимаемая схемой. Указанное ограничение может быть учтено при задании размеров каждого элемента и суммарной площади, требуемой для реализации соединений.

С точки зрения вычислительной процедуры, алгоритмы компоновки конструктивных узлов можно разделить на последовательные, параллельно-последовательные и итерационные. В алгоритмах первого типа вводится последовательный процесс компоновки узлов, на каждом шаге которого в очередной узел добавляется один из элементов схемы (например, алгоритм компоновки по связности). В параллельно-последовательных алгоритмах сначала выделяется некоторое исходное количество групп элементов, которые потом распределяют по узлам с учётом ограничений и критериев на компоновку.

Последовательные и параллельно-последовательные алгоритмы применяют для создания базового (начального) варианта компоновки при заданных ограничениях на число элементов в узле число выводов на узле.

Итерационные алгоритмы компоновки служат для улучшения некоторого начального распределения в соответствии с принятыми критериями и используются в сочетании с другими алгоритмами компоновки.

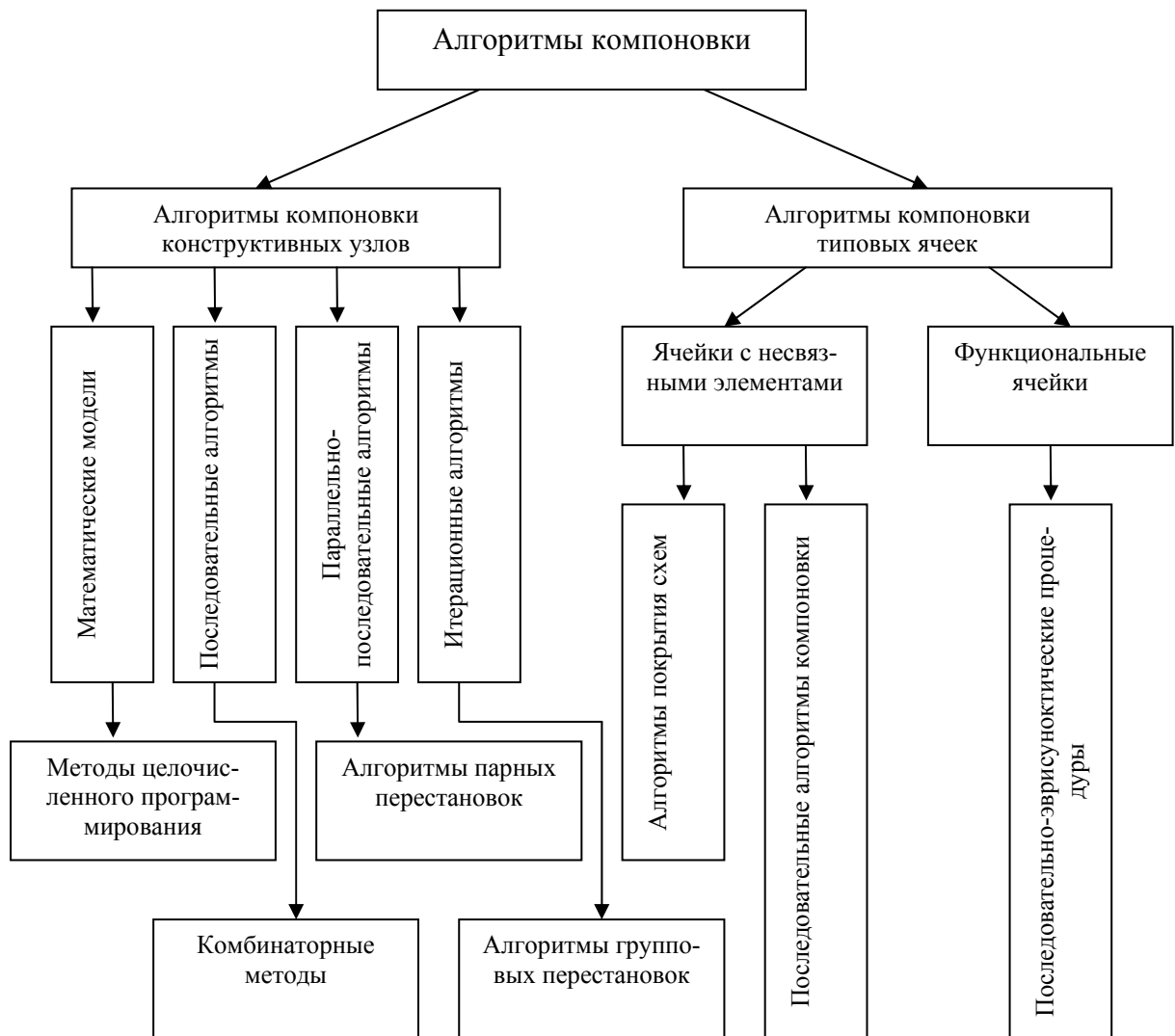


Рис. 15.2

Основной задачей алгоритмов компоновки типовых узлов является получение покрытия с минимальной суммарной стоимостью (минимум числа использованных типовых узлов). Структура алгоритмов зависит от особенностей используемого набора типовых узлов.

Последовательный алгоритм компоновки по связности

Рассматриваемый ниже метод занимает центральное место в группе последовательных алгоритмов компоновки. Его первоначальная версия была изложена в работе Гэмблина и получила название метода максимальной конъюнкции – минимальной дизъюнкции. Основу метода составляет последовательная процедура выделения из исходной схемы связанных групп элементов, осуществляемая с помощью операций конъюнкции и дизъюнкции над элементами схемы. Этот метод был использован для компоновки узлов (ячеек

и панелей) в системе автоматизации фирмы IBM. В дальнейшем появились различные модификации метода, учитывающие конкретные ограничения в задачах компоновки конструктивных узлов и модулей логических схем.

Пусть дана схема соединений элементов из множества $E = \{e_1, \dots, e_n\}$. Определим последовательный процесс назначения элементов $e_i \in E$ ($i=1, \dots, n$) в узлы T_r . Выбирается один из не распределенных элементов и приписывается очередному узлу.

Узел считается завершенным, если число элементов в узле равно заданному числу k либо назначение любого из нераспределенных элементов приводит к образованию такого числа внешних связей узла, которое превышает допустимое значение v .

После завершения очередного узла аналогичная процедура повторяется для следующего, причем кандидатами для назначения являются элементы, не включенные в предыдущие узлы. Процесс заканчивается, когда все элементы множества E распределены.

Тактика назначения заключается:

1) на очередном шаге процесса выделяются те из нераспределенных элементов, включение каждого из которых в данный узел не нарушает ограничений по числу элементов и выводов узла;

2) элементом, включенным в узел на очередном шаге, является тот из указанных в п. 1 элементов, который имеет наибольшее число связей с элементами уже включенных в узел (максимальная конъюнкция). При нескольких таких элементах включается тот из них, который имеет минимальную дизъюнкцию с элементами узла.

Описанный выше последовательный процесс компоновки отражает общую схему метода максимальной конъюнкции – минимальной дизъюнкции. Конкретные алгоритмы, реализующие данный метод, отличаются способами представления схемы соединений и вычисления оценок, управляющих процессом компоновки.

Ниже рассматривается алгоритм компоновки, в котором схема представлена графом $G = (E, V, W)$ (матрицей комплексов Q).

Формирование очередного узла $T_r = (r = 1, 2, \dots, \gamma)$ начинается с выбора базового элемента i_r^* из множества нераспределенных элементов I_r . В начале процесса все элементы считаются не распределенными, т. е. $I_1 = E$.

Для элемента $x \in I_r$ введем функционал

$$L_1(x) = |x \cap I'_r|, \tag{15.1}$$

определяющий число цепей, связывающих элемент x и элементы из множества $I'_r = I_r \setminus x$. Для упрощения записи здесь и в дальнейшем будем отождествлять элемент (множество элементов) с множеством цепей, связанных с этим элементом (множеством элементов).

Базовый элемент i_r^* есть первый по порядку элемент из I_r , для которого функционал (15.1) принимает максимальное значение. Элемент i_r^* поме-

щается в узел T_r , а оставшиеся элементы $I_r \setminus i_r^*$ являются кандидатами для включения в узел T_r на последующих шагах алгоритма. Таким образом, элемент i_r^* , помещаемый первым в узел, станет как бы «центром группирования», к которому в дальнейшем добавляются новые элементы.

Последовательность компоновки узла T_r управляется функционалами $L_2(x)$ и $L_3(x)$.

Рассмотрим λ -й шаг ($\lambda = 1, 2, \dots, k-1$) при назначении элементов в узел T_r ($r = 1, 2, \dots, \gamma$). Пусть в узле уже размещено λ элементов:

$$T^\lambda = \{i_r^*, e_1, e_2, \dots, e_{\lambda-1}\}. \tag{15.2}$$

Функционал $L_2(x)$, заданный на множестве нераспределенных к данному моменту элементов $x \in I_r^\lambda$,

$$I_r^\lambda = I_r \setminus T_r^\lambda, \tag{15.3}$$

определяет число внешних соединений для узла

$$T_r^\lambda(x) = \{i_r^*, e_1, e_2, \dots, e_{\lambda-1}, x\}, \tag{15.4}$$

полученного добавлением элемента в узел (15.3).

В том случае, когда $L_2(x) > v$, число внешних соединений превышает предельно допустимое. Так как процесс компоновки узлов является последовательным, то включение в узел T_r – элементов, для которых $L_2(x) > v$, может привести к тому, что завершённый узел будет иметь недопустимое число внешних соединений. В силу этого такие элементы из рассмотрения исключаются.

Для формального задания $L_2(x)$ обратимся к рис.15.3, где представлена компоновка узла.

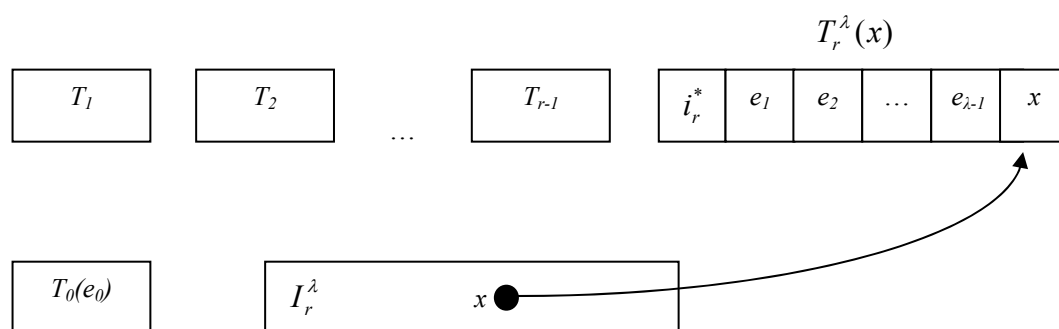


Рис. 15.3

Число выводов, требуемое для соединения элементов множества $T_r^\lambda(x)$ (15.4) с остальными элементами, равно

$$L_2(x) = |T_r^\lambda(x) \cap \{\bigcup_{s=0}^{r-1} T_s \cup I_r^\lambda\}|, \quad (15.5)$$

где I_r^λ – цепи, связанные с элементами множества I_r^λ (15.3), за исключением элемента x .

С помощью функционала $L_3(x)$ из элементов, удовлетворяющих условию $L_2(x) \leq \nu$, отбирается такой элемент, для которого число цепей, связанных с элементами из T_r^λ , максимально

$$L_3(x) = |x \cap T_r^\lambda|. \quad (15.6)$$

Элемент x имеет максимальную конъюнкцию с множеством T_r^λ . Если имеется несколько элементов с равными и максимальным значениям $L_3(x)$, то выбирается тот, для которого $L_2(x)$ минимальное значение.

В рассматриваемом алгоритме вместо вычисления дизъюнкции элемента x и множества I_r^λ выбор (при равных значениях конъюнкции) осуществляется на основании значения $L_2(x)$, что приводит к более экономной схеме вычислений.

Итерационные алгоритмы компоновки

Пусть имеется некоторый начальный вариант компоновки, полученный либо одним из последовательных алгоритмов, либо вручную. Основой рассматриваемых алгоритмов компоновки является использование итерационного процесса обменов местами элементов или групп элементов, принадлежащих различным узлам, с целью минимизации некоторого критерия. Поскольку структура большинства алгоритмов этого типа имеет много общего, рассмотрим процесс улучшения компоновки применительно к произвольному критерию F .

Обозначим через K_0 некоторый исходный вариант компоновки элементов. Ему соответствует значение функции-критерия F_0 . В некотором смысле структура итерационного процесса улучшения варианта K_0 аналогична градиентным методам оптимизации, в которых на каждой итерации осуществляется движение в области определения функции в направлении уменьшения ее значений. Изменениям переменных в данном случае соответствует перестановка элементов между различными узлами.

Пусть задано некоторое разбиение K_0 множества элементов E на узлы $T_1, \dots, T_i, \dots, T_j, \dots, T_\gamma$. Предположим, что произведен выбор пары элемен-

тов $e_x \in T_i$ и $e_y \in T_j$, и они взаимно меняются местами. В этом случае получаем новый вариант разбиения K_1 с узлами

$$T'_i = (T_i \setminus e_x) \cup e_y, \quad T'_j = (T_j \setminus e_y) \cup e_x, \quad (15.7)$$

которому соответствует значение критерия F_1 .

Если $\Delta F = F_0 - F_1 > 0$, то происходит уменьшение функции-критерия F и обмен, соответствующий данному случаю, считается успешным.

Рассматривая вариант компоновки K_i снова как исходный, можно осуществить поиск другой пары элементов, обмен которых приведет к уменьшению значения F_1 , и т. д. Таким образом, в ходе указанного итерационного процесса образуется последовательность вариантов компоновки $K_0, K_1, K_2, \dots, K_s$, которым отвечает монотонно убывающая последовательность значений критерия: $F_0 > F_1 > F_2 > \dots > F_s$ (рис.15.4). Поскольку значения функции F на множестве допустимых вариантов компоновки $\{K\}$ ограничены снизу величиной абсолютного минимума функции F^* , естественно, что на некотором шаге процесс минимизации закончится, причем $F_s \geq F^*$, т. е. мы получим вариант компоновки K_s , отвечающий локальному минимуму функции F .

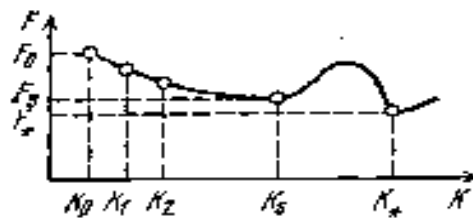


Рис. 15.4

В этот момент обмен любой пары элементов уже не приводит к уменьшению значения F_s . Для дальнейшего уменьшения F надо рассматривать, например, перестановки групп элементов из различных узлов.

Нужно указать на возможность использования различных тактик при выборе пары элементов, обеспечивающих успешный обмен.

Одна из них состоит в следующем. Выбирается некоторый узел T_i и в нем произвольный элемент e_x . Осуществляются попытки обмена этого элемента последовательно со всеми элементами e_y , не принадлежащими рассматриваемому узлу, и рассчитывается приращение $\Delta F(e_x, e_y)$. Для обмена выбирается первый элемент e_y , для которого $\Delta F(e_x, e_y) > 0$.

Другой способ состоит в подборе такой пары элементов, для которой $\Delta F(e_x, e_y)$ принимает максимальное значение. Очевидно, что в этом случае время поиска успешного обмена возрастает, однако в результате обмена число межузловых соединений будет меньше.

Из сказанного следует, что обобщенная характеристика эффективности любого итерационного алгоритма компоновки должна учитывать как время получения решения, так и степень его приближения к оптимальному. Вместе с тем в настоящее время не представляется возможным дать теоретическую оценку эффективности различных алгоритмов компоновки. Наиболее реальным способом оценки является их экспериментальное сопоставление на задачах одного типа. Имеющиеся данные показывают, что итерационные алгоритмы наиболее целесообразно использовать для улучшения начального варианта компоновки, полученного одним из быстрых последовательных алгоритмов.

Для фиксации момента окончания итерационного процесса при реализации алгоритма на ЭВМ применяются различные правила. Например, может задаваться число итераций m либо параметр δ , определяющий число итераций неявным образом:

$$(F_m - F_{m+1}) \setminus F_m \leq \delta. \tag{15.8}$$

Пусть задано некоторое разбиение множества $E = \{e_1, e_2, \dots, e_n\}$ на узлы $T_1, T_2, \dots, T_\gamma$.

Рассмотрим сначала случай, когда схема описана матрицей соединений $R = ||r_{ij}||_{n \times n}$. Оценим изменение в количестве межузловых соединений при обмене местами элементов $e_x \in T_i$ и $e_y \in T_j$. Поскольку при обмене перераспределяются лишь соединения элементов e_x и e_y , рассмотрим более детально структуру их соединений (рис. 15.5).

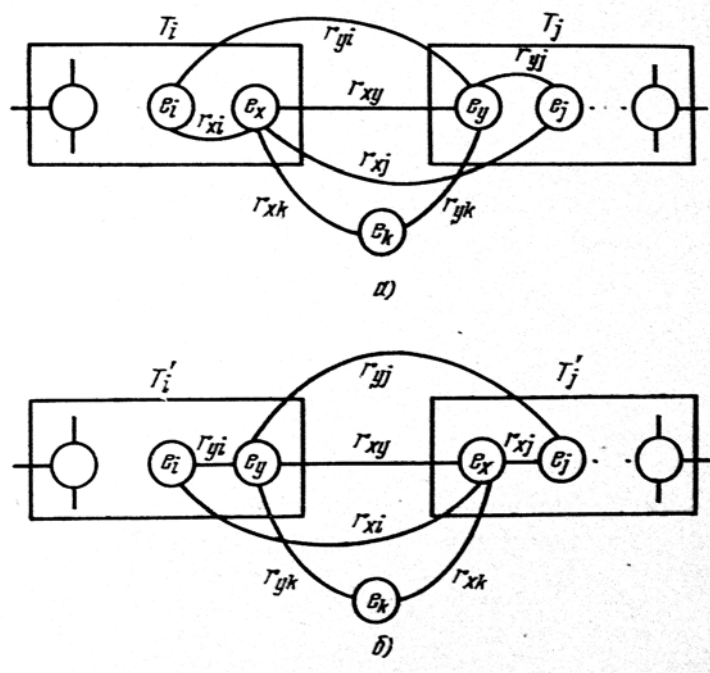


Рис. 15.5

Обозначим через U , множество элементов $\{e_k: e_k \notin T_i; \text{ и } e_j \in T_j\}$. Тогда в соответствии с [рис.15.5, а](#) количество межузловых соединений до обмена равно

$$L = \sum_{e_j \in (T_j \setminus e_y)} r_{xj} + \sum_{e_i \in (T_i \setminus e_x)} r_{yj} + \sum_{e_k \in U_{ij}} r_{xk} + \sum_{e_k \in U_{ij}} r_{yk} + r_{xy} + C, \quad (15.9)$$

где C – число межузловых соединений, не участвующих в обмене элементов.

В соответствии с [рис.15.5, б](#) после обмена количество межузловых соединений станет равным

$$L_1 = \sum_{e_i \in (T_i \setminus e_y)} r_{xi} + \sum_{e_j \in (T_j \setminus e_x)} r_{yj} + \sum_{e_k \in U_{ij}} r_{yj} + \sum_{e_k \in U_{ij}} r_{xk} + r_{xy} + C. \quad (15.10)$$

Вычитая [\(15.10\)](#) из [\(15.9\)](#), получим

$$\Delta L = L - L_1 = \left(\sum_{e_j \in (T_j \setminus e_y)} r_{xj} - \sum_{e_i \in (T_i \setminus e_x)} r_{xi} \right) + \left(\sum_{e_i \in (T_i \setminus e_x)} r_{yi} - \sum_{e_j \in (T_j \setminus e_y)} r_{yj} \right). \quad (15.11)$$

Пусть L_{xj} – число соединений e_x с элементами узла T_j , L_{yi} – число соединений e_y с элементами узла T_i :

$$L_{xi} = \sum_{e_i \in T_i} r_{xi}, \quad L_{yj} = \sum_{e_j \in T_j} r_{yj}, \quad (15.12)$$

а F_{xi} – число соединений e_x с элементами узла T_i и F_{yj} – число соединений e_y с элементами узла T_j

$$F_{xi} = \sum_{e_i \in T_i} r_{xi}, \quad F_{yj} = \sum_{e_j \in T_j} r_{yj}. \quad (15.13)$$

Тогда [\(15.11\)](#) принимает вид:

$$\Delta L(x, y) = [L_{xj} - F_{xi}] + [L_{yi} - F_{yj}] - 2r_{xy}, \quad (15.14)$$

L_{xi} (L_{yj}) и F_{xi} (F_{yj}) соответственно внешние и внутренние соединения элементов e_x и e_y .

Для произвольного элемента $e_x \in T_i$ удобно ввести характеристику $D_x = L_{xj} - F_{xi}$, представляющую собой разность числа внешних и внутренних соединений. Аналогично вводится характеристика для $e_y \in T_j$.

С учётом принятых обозначений формула [\(15.14\)](#) приобретает вид

$$\Delta L(x, y) = D_x + D_y - 2r_{xy}. \quad (15.15)$$

При обмене местами элементов $e_x \in T_i$ и $e_y \in T_j$ наряду с изменением количества межузловых соединений L происходит перераспределение внешних соединений для узлов T_i и T_j .

При задании схемы матрицей R суммарное число выводов на узлах $V = 2L$, поэтому

$$\Delta V(x, y) = \Delta v_i(x, y) + \Delta v_j(x, y) = 2\Delta L, \quad (15.16)$$

где $\Delta V(x, y)$ – изменение суммарного числа выводов на узлах, а $\Delta v_i(x, y)$ и $\Delta v_j(x, y)$ – изменение числа выводов на узлах T_i и T_j .

Получим теперь выражения для $\Delta v_i(x, y)$ и $\Delta v_j(x, y)$. Обратимся к [рис.15.4](#). Число внешних выводов на узле равно числу соединений между элементами узла и элементами, не входящими в узел. В связи с этим для расчета $\Delta v_i(x, y)$ можно использовать [\(15.15\)](#), если придать входящим в нее характеристикам D_x и D_y другой содержательный смысл.

Можно считать, что имеются два узла T_i и $T_j^* = U_{ij} \cup T_j = E \setminus T_i$. Введем характеристики D_x^* и D_y :

$$D_x^* = L_{xj^*} - F_{xi}, \quad D_y^* = L_{yi} - F_{yj^*}, \quad (15.17)$$

тогда на основании [\(15.15\)](#)

$$\Delta v_i(x, y) = D_x^* + D_y^* - 2r_{xy}. \quad (15.18)$$

Заметим, что

$$D_x^* = D_x + \sum_{e_k \in U_{ij}} r_{xk}, \quad (15.19)$$

а

$$D_y^* = D_y - \sum_{e_k \in U_{ij}} r_{yk}, \quad (15.20)$$

поэтому формулу [\(15.18\)](#) можно записать, учитывая [\(15.15\)](#), в таком виде:

$$\Delta v_i(x, y) = \Delta L(x, y) + \sum_{e_k \in U_{ij}} (r_{xk} - r_{yk}). \quad (15.21)$$

На основании [\(15.16\)](#) получим

$$\Delta v_j(x, y) = \Delta L(x, y) + \sum_{e_k \in U_{ij}} (r_{yk} - r_{xk}). \quad (15.22)$$

Пусть теперь схема задана одним из способов, при котором непосредственно учитываются соединения с размером $p > 2$. Для формул прираще-

ний, аналогичных (15.15), (15.21) и (15.22), воспользуемся аппаратом алгебры соединений.

Число межузловых соединений L в этом случае будет равно

$$L = \sum_{r=1}^{\gamma} |J_r| - M, \quad (15.23)$$

где J_r – множество цепей, связанных с r -м узлом; M – общее количество цепей в схеме.

Найдем приращение количества межузловых соединений $\Delta L(x,y)$ при обмене элемента $e_x \in T_i$ с элементом $e_y \in T_j$.

Для упрощения будем обозначать множество цепей, связанных с некоторым подмножеством элементов $A \subset E$, тем же символом. Тогда в результате обмена новым узлам T'_i и T'_j будут отвечать множества цепей:

$$T'_i = (T_i \setminus e_x) \cup e_y, \quad T'_j = (T_j \setminus e_y) \cup e_x. \quad (15.24)$$

В соответствии (15.23) после обмена количество межузловых соединений станет равно

$$L' = |T'_i| + |T'_j| + \sum_{r \neq i,j} |T_r| - M, \quad (15.25)$$

поскольку $T'_r = T_r$ при $r \neq i,j; r = 1, 2, \dots, \gamma$.

На основании (15.23) и (15.25) получим

$$\Delta L = L - L' = |T_i| - |T'_i| + |T_j| - |T'_j|. \quad (15.26)$$

Представим множества цепей узлов T_i и T_j в виде

$$T_i = (T_i \setminus e_x) \cup e_x, \quad T_j = (T_j \setminus e_y) \cup e_y. \quad (15.27)$$

Подставляя в (15.26) выражения (15.24) и (15.27) и применяя тождество

$$|a \cup b| = |a| + |b| - |a \cap b|, \quad (15.28)$$

получим после ряда упрощений формулу, аналогичную (15.11):

$$\Delta L(x,y) = (|e_x \cap (T_j \setminus e_y)| - |(T_i \setminus e_x) \cap e_x|) + (|e_y \cap (T_i \setminus e_x)| - |e_y \cap (T_j \setminus e_y)|). \quad (15.29)$$

Используем ранее принятые обозначения. Пусть L_{xy} – число цепей, связывающих элемент e_x с элементами узла T_j , и F_{xi} – число цепей, связывающих e_x с элементами узла T_i . Очевидно, что

$$L_{xj} = |e_x \cap T_j|, \quad F_{xi} = |e_x \cap (T_i \setminus e_x)|.$$

Аналогичные выражения имеют место для характеристик L_{yi} и F_{yi} элемента e_y . Преобразуем теперь выражение для L_{xj} , используя тождество (15.28):

$$\begin{aligned} L_{xj} &= |e_x \cap ((T_j \setminus e_y) \cup e_y)| = |(e_x \cap (T_j \setminus e_y)) \cup (e_x \cap e_y)| = \\ &= |e_x \cap (T_j \setminus e_y)| + |e_x \cap e_y| - |e_x \cap e_y \cap (T_j \setminus e_y)|. \end{aligned} \quad (15.30)$$

Осуществив подобное же преобразование выражения для L_{yi} и подставляя вместо первого и третьего членов в формуле (15.29) соответствующие выражения, получим

$$\begin{aligned} \Delta L(x, y) &= [L_{xj} - F_{xi}] + [L_{yi} - F_{yi}] - 2|e_x \cap e_y| + \\ &+ |e_x \cap e_y \cap (T_j \setminus e_y)| + |e_x \cap e_y \cap (T_i \setminus e_x)|. \end{aligned} \quad (15.31)$$

Обозначая, как и ранее в (15.14), члены в скобках (15.31) через D_x и D_y , окончательно придем к следующему результату:

$$\begin{aligned} \Delta L(x, y) &= D_x + D_y - 2|e_x \cap e_y| + |e_x \cap e_y \cap (T_j \setminus e_y)| + \\ &+ |e_x \cap e_y \cap (T_i \setminus e_x)|. \end{aligned} \quad (15.32)$$

Сравнение (15.32) с (15.15), полученной для задания схемы матрицей соединений R , показывает, что при представлении схемы цепями (комплексами) необходимо при расчете ΔL учитывать поправочные члены, связанные с наличием многоконцевых цепей ($p > 2$). Заметим, что член $|e_x \cap e_y|$ равен числу цепей, соединяющих элементы e_x и e_y , и в (15.15) ему соответствует член r_{xy} .

Перейдем теперь к расчету изменений выводов на узлах. Согласно (15.19) число выводов на узле T_i равно

$$v_i = |J_i \cap J_i|, \quad (15.33)$$

где e_x и e_y – множества цепей, связанных соответственно с элементами узла T_i , и с элементами, не входящими в узел T_i .

При обмене местами элементов e_x и e_y изменяется число выводов лишь на узлах T_i и T_j .

Вычислим приращение в числе выводов на узле T_i :

$$\Delta v_i(x, y) = v_i - v'_i, \quad (15.34)$$

где v_i , и v'_i – соответственно число внешних выводов узла до и после обмена. Заметим, что приращение числа выводов $\Delta v_i(x, y)$ равно приращению числа соединений между элементами узла T_i и элементами, не входящими в узел T_i . Поэтому формула для расчета $\Delta v_i(x, y)$ может быть непосредственно получена из (15.34) при замене узла T_j на множество элементов $I_i = \bigcup_{r \neq i} T_r = E \setminus T_i$

$$\Delta v_i(x, y) = D_x^{I_i} + D_y^{I_i} - 2|e_x \cap e_y| + |e_x \cap e_y \cap (I_i \setminus e_i)| + |e_x \cap e_y(T_i \setminus e_x)|, \quad (15.35)$$

где

$$D_x^{I_i} = L_{xI_i} - F_{xi}, \quad D_y^{I_i} = L_{yI_i} - F_{yi} \quad (15.36)$$

$$L_{xI_i} = |e_x \cap I_i|, \quad F_{yi} = |e_y \cap I_i|. \quad (15.37)$$

Аналогичная формула имеет место для приращений числа выводов на узле T_j . Для ее получения необходимо лишь заменить везде в (15.35), (15.36), (15.37) символ I_i на I_j , обозначающий множество элементов, не входящих в узел T_j : $I_j = E \setminus T_j$, символ x на y , а символ y на x .

Заметим, что для многоконцевых соединений равенство, аналогичное (15.16), уже не выполняется.

Пусть известен некоторый вариант разбиения схемы на γ узлов. Можно считать, что в каждом узле содержится ровно k элементов, причем $n = k\gamma$. В случае, когда в каком-либо узле число элементов k_r меньше k , всегда можно дополнительно ввести $n_\delta = k - k_r$ фиктивных элементов, не связанных с другими элементами схемы: $r_{sj} = 0, s = n+1, \dots, n + n_\delta, j = 1, 2, \dots, n + n_\delta$.

Рассмотрим два узла – T_i , и T_j , ($i, j = 1, 2, \dots, \gamma$). В соответствии с (15.15) приращение числа межузловых соединений при обмене местами элементов $e_x \in T_i$, и $e_y \in T_j$ будет равно $\Delta L(x, y) = D_x + D_y - 2r_{xy}$, где характеристики D_x и D_y рассчитываются по матрице соединений R .

Опишем процесс выбора пары элементов e_x и e_y для обмена. Сначала определяются характеристики D_z , для всех элементов e_z , принадлежащих T_i , или T_j . Очевидно, что число этих характеристик равно $2k$.

Далее находится пара элементов $e_x \in T_i$ и $e_y \in T_j$, для которой (15.15) принимает максимальное значение.

Если $\Delta L(x, y) > 0$, то производится обмен. В результате образуются узлы T'_i и T'_j .

Зафиксировав положение элементов e_x и e_y соответственно в узлах T'_i и T'_j , выполняем аналогичный расчет для элементов из множеств $T_i \setminus e_x$ и $T_j \setminus e_y$ и определяем новую пару элементов для обмена e_{x_2} и e_{y_2} . Процесс об-

менов продолжается до тех пор, пока на очередном шаге $S + 1$ среди элементов $T_i^s \setminus e_{x_s}, T_j^s \setminus e_{y_s}$ не найдется ни одной пары, для которой $\Delta L(x, y) > 0$. Узлы T_i^s и T_j^s отвечают результату минимизации числа межузловых соединений между исходными узлами T_i и T_j .

Заметим, что при обмене элементов не все из рассчитанных на предыдущем шаге характеристик D изменяются. Имеют место следующие соотношения:

$$\begin{aligned} D'_x &= D_x + 2r_{xx_1} - 2r_{xy_1}, e_x \in (T_i \setminus e_{x_1}), \\ D'_y &= D_y + 2r_{yy_1} - 2r_{yx_1}, e_y \in (T_j \setminus e_{y_1}), \end{aligned} \tag{15.38}$$

$$D'_{x_1} = -D_{x_1} + 2r_{x_1y_1}, D'_{y_1} = -D_{y_1} + 2r_{x_1y_1}. \tag{15.39}$$

Справедливость (15.38) может быть легко проверена. Действительно, r_{xx_1} внутренних соединений элемента e_x в результате обмена становятся внешними по отношению к узлу T_i , а r_{xy_1} внешних соединений – внутренними.

Отсюда непосредственно следует первое из соотношений (15.33). Остальные соотношения могут быть проверены аналогично.

Использование (15.38) и (15.39) существенно сокращает трудоемкость вычислений характеристик D при расчете приращений $\Delta L(x, y)$ на каждой итерации. Для времени поиска пары элементов с $\Delta L(x, y) > 0$ можно предварительно упорядочить характеристики D (сортировка).

Расположим характеристики D элементов из узлов T_i и T_j следующим образом:

$$\begin{aligned} x : D'_{x_1} &\geq D'_{x_2} \geq \dots \geq D'_{x_k} \\ y : D'_{y_1} &\geq D'_{y_2} \geq \dots \geq D'_{y_k} \end{aligned} \tag{15.40}$$

Организуем процесс вычисления $\Delta L(x, y)$ так, чтобы в первую очередь кандидатами для обмена были элементы, находящиеся в начале последовательностей x и y в (15.40), тогда, если $D_{x_{s^*}} + D_{y_{s^*}}$ меньше, чем максимальное из ранее найденных значений $\Delta L(x, y)$, то по (15.15) среди оставшихся элементов $\{x_s: s > s^*\}$ и $\{y_l: l > l^*\}$ нет пары с большим значением $\Delta L(x, y)$, поэтому поиск может быть прекращен.

Таким образом, предварительная сортировка D позволяет сократить количество просмотров для вычисления $\Delta L(x, y)_{\max}$ по сравнению с k^2 просмотрами при обычном переборе.

После окончания процесса минимизации числа соединений между узлами T_i и T_j (малая итерация) аналогичная процедура может быть применена к другой паре узлов. Так как имеется γ узлов, то в принципе возможно осуществить $p = \gamma(\gamma - 1)/2$ малых итераций. Совокупность малых итераций со-

ставляет одну большую. После окончания большой итерации минимизацию числа межузловых соединений можно продолжить, начиная снова с первой пары узлов. Процесс заканчивается, когда либо обмены элементов между двумя любыми узлами не приводят к уменьшению целевой функции, либо было проведено p^* больших итераций.

В рассмотренном выше алгоритме парных перестановок производится последовательная минимизация связей между парами узлов в начальном варианте компоновки. Однако возможны и другие способы организации итерационного процесса. Среди них отметим метод последовательного разделения и метод последовательного выделения.

Идея первого метода: для упрощения предположим, что число компонентов узлов $\gamma = 2^l$. В качестве начального варианта используем какой-либо вариант разбиения n элементов на две группы A и B по $n/2$ элементов. Далее применяем основной алгоритм парных перестановок для узлов A и B . Полученные узлы снова разделяем на два и т. д. до получения узлов требуемого размера. На [рис. 15.6](#) представлен процесс последовательного разделения.

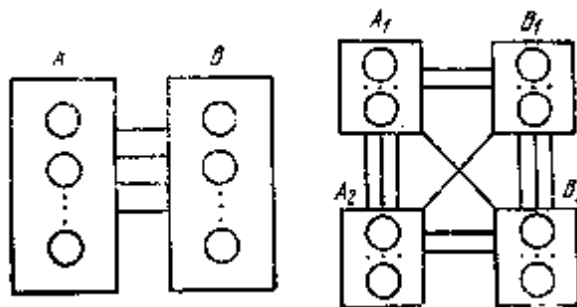


Рис. 15.6

Вначале минимизируются связи между группами A и B , затем связи между группами A_1 и A_2 , B_1 , и B_2 . Результат, полученный для одного уровня разделения, не ухудшается при последующих разделениях. Данный метод имеет наибольшую эффективность для схем, содержащих сильно связанные группы элементов, причем размеры этих групп приблизительно одинаковы.

Другой метод состоит в последовательном выделении ([рис.15.7](#)) из исходного множества n элементов групп по k элементов с использованием итерационного процесса обменов элементов, входящих в выделенную группу, T_i элементами из множества

$$T' = E \setminus \bigcup_{p=1}^i T_p. \tag{15.41}$$

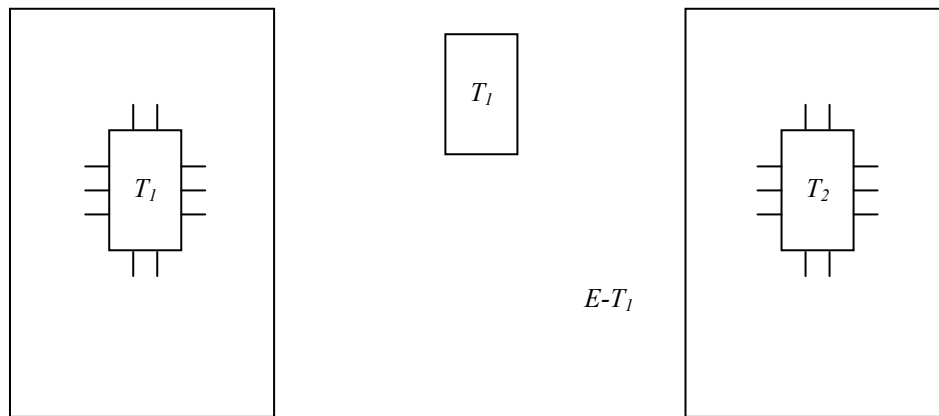


Рис. 15.7

Рассмотрим [рис. 15.7](#). Пусть получено разбиение исходного множества элементов на две группы – T_1 и $T'_1 = E \setminus T_1$. Указанное разбиение может быть, например, произведено с помощью алгоритма последовательной компоновки. Обычным образом производится минимизация числа соединений между узлом T_i и множеством элементов T'_1 . Далее из множества T'_1 выделяется следующая группа T_2 из k элементов и осуществляется аналогичная процедура минимизации для множеств T'_2 и $T'_2 = T'_1 \setminus T_2$ т. д. Процесс продолжается до тех пор, пока не будут выделены γ узлов.

Метод минимизации соединений между всеми парами узлов разбиения имеет наибольшую точность, однако требует значительных затрат машинного времени. Эффективность метода последовательного разделения в большей степени зависит от структуры связей между группами элементов схемы.

В методе последовательного выделения существен «удачный» выбор первых элементов, поскольку в ходе процесса элементы, вошедшие в группы $T_p (p < i)$, исключаются из рассмотрения.

Алгоритмы размещения

Выделяют основные группы алгоритмов размещения ([рис. 15.8](#)).

1. Алгоритмы решения математических задач, являющиеся моделями задачи размещения.
2. Конструктивные алгоритмы начального размещения.
3. Итерационные алгоритмы улучшения начального варианта размещения.
4. Непрерывно-дискретные методы размещения.

К первой группе относятся, прежде всего, метод ветвей и границ для задачи квадратичного назначения, к которой при определённых упрощениях сводится задача размещения: набор позиций считается фиксированным, элементы рассматриваются как геометрические точки, схема соединений представляется взвешенным графом. Другой класс моделей связан с оптимизацией размещения, когда набор позиций для установки заранее не фиксирован.

Вторая и третья группы включают приближённые алгоритмы, в основном предназначенные для оптимизации размещения элементов в фиксированном наборе позиций.

Характерной чертой конструктивных алгоритмов является то, что они создают размещение. Тогда как итерационные алгоритмы предполагают задание начального размещения. Конструктивные алгоритмы используют последовательный или параллельно-последовательный процесс установки элементов в позиции при локальной оптимизации функции-критерия размещения.

В итерационных алгоритмах производится перерасположение элементов или их групп с целью минимизации выбранного критерия. Эти алгоритмы требуют существенных затрат машинного времени и используются для получения конечного размещения.

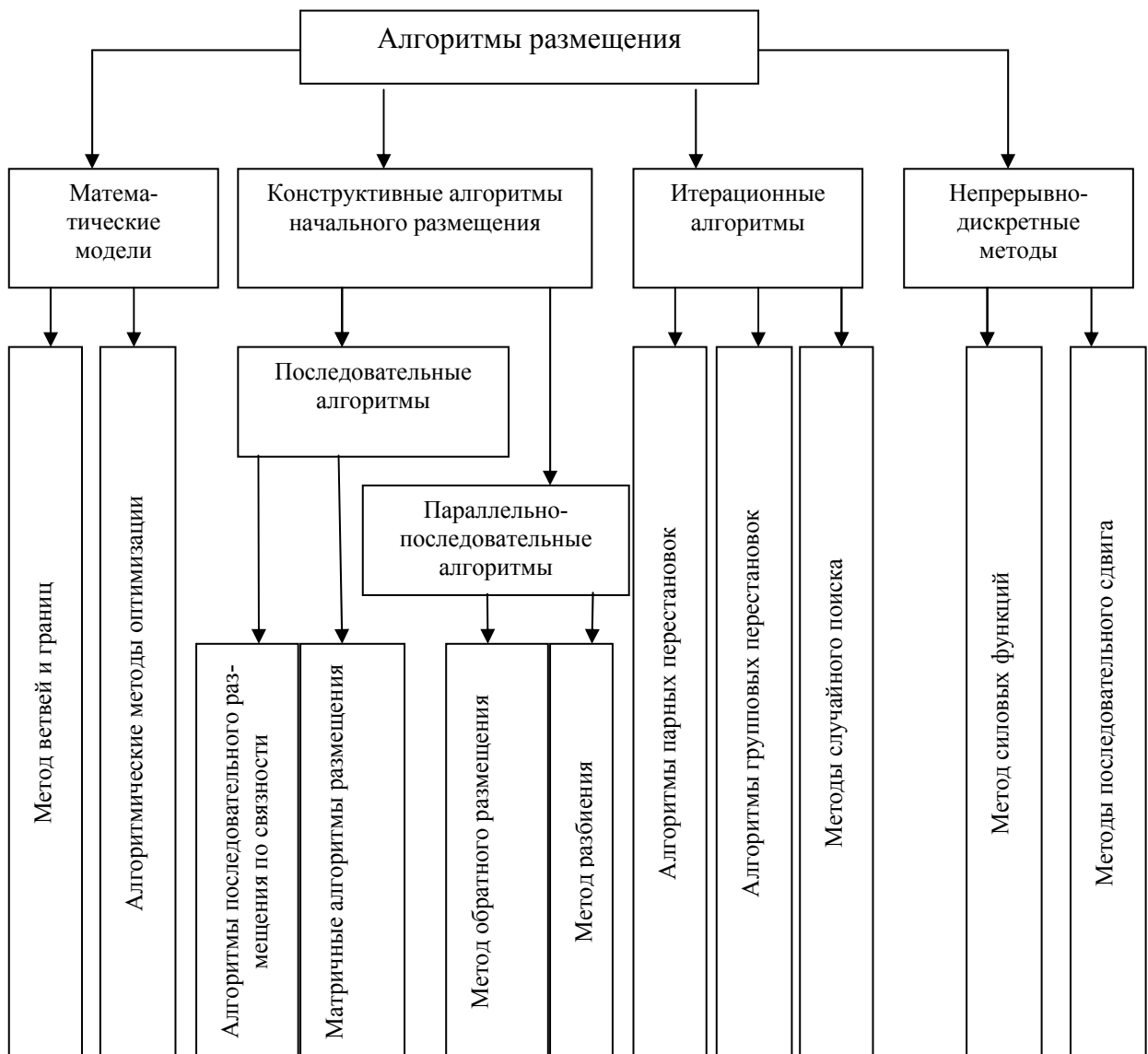


Рис. 15.8

Основной областью применения непрерывно-дискретных методов размещения являются конструкции, в которых позиции для установки элементов заранее не фиксированы. Исходной базой для построения этих алгоритмов служат непрерывные модели.

Исходной информацией для размещения является схема соединений, параметры конструкции элементов и коммутационного поля. Прежде всего рассмотрим наиболее представительную группу алгоритмов, использующих последовательный процесс установки элементов в позиции.

Алгоритмы последовательного размещения по связности

Пусть $E = \{e_1, e_2, \dots, e_n\}$ – множество элементов, подлежащих размещению, а $L = \{l_1, l_2, \dots, l_n\}$ – множество позиций для их установки. Вводится n -шаговый процесс принятия решений, на каждом шаге которого выбирается один из неразмещенных элементов и помещается в одну из незанятых позиций. Структура любого последовательного алгоритма размещения определяется правилами выбора очередного элемента и позиции для его установки.

Пусть E_k – элементы, размещенные до k -го шага, а L_k – позиции, занятые этими элементами; E'_k и L'_k – соответственно неразмещенные элементы и незанятые позиции. Отметим, что перед началом размещения могут быть две ситуации: нет размещенных ранее элементов, внешние выводы узла (контакты, разъемы и т. п.) не закреплены (в этом случае в алгоритме должен быть особо определен способ установки первого элемента); имеется группа заранее размещенных элементов или закрепленных внешних выводов.

В основу большинства последовательных алгоритмов размещения положен эвристический принцип оптимизации целевой функции, сводящийся к выбору на данном шаге локально оптимальной позиции для одного из элементов при неизменности положения ранее размещенных элементов. Поскольку критерий минимума суммарной длины соединений в силу отмеченных ранее причин наиболее распространен, он и будет рассмотрен при описании алгоритмов данной группы. Вместе с тем используемые в этих алгоритмах тактики размещения могут быть (с некоторыми модификациями) применены и для других критериев. В алгоритмах размещения по связности элемент и позиция выбираются независимо.

Выбор элемента. Любое правило выбора элемента для размещения основано на вычислении «меры связности» еще неразмещенных элементов с уже размещенными. Естественной мерой связности двух элементов e_i и e_j является количество соединений между ними, заданное в матрице соединений $R = \|r_{ij}\|_{n \times n}$.

Так, в алгоритме «попарных связей» Куртзберга для каждого неразмещенного элемента e_i подсчитывается характеристика

$$c_i = \max_{e_j \in E_k} r_{ij}, \quad (15.42)$$

причем r_{ij} засчитывается по формуле

$$r_{ij} = \sum_{s \in J_{ij}} \frac{p_s + \lambda}{p_s} w_s, \quad (15.43)$$

в которой J_{ij} – множество цепей, связывающих элементы e_i и e_j , p_s – размер цепи; w_s – весовой коэффициент; λ – целочисленный параметр.

Параметр λ позволяет дифференцировать вклад цепей различного размера. Чем больше значение λ , тем больше влияние цепей с малым значением p_s . Очевидно, что при $\lambda = 0$ значение r_{ij} (15.43) равно суммарному весу цепей между элементами e_i и e_j . Поскольку всегда $P_2 \geq 2$, то $\lambda \geq -1$.

Очередным размещаемым элементом является элемент, имеющий максимальную характеристику (15.42), т. е. выбор элемента осуществляется по наибольшему числу связей с уже размещенными элементами.

Другое правило выбора основано на расчете для каждого неразмещенного элемента $e_i \in E'_k$ суммарной связности с уже размещенными элементами. Если задаётся матрица соединений R , то соответствующая характеристика будет такой:

$$c_i = \sum_{e_j \in E_k} r_{ij}. \quad (15.44)$$

Если для схемы задан список цепей (комплексов), то аналогичная характеристика имеет вид

$$c_i = |J_i(\bigcup_{e_j \in E_k} J_j)|, \quad (15.45)$$

где J_i – множество цепей, связанных с элементом e_i . C_i равна конъюнкции множества цепей элемента e_i и множества цепей E_k . Выбор в данном случае осуществляется на основании максимального значения характеристики (15.44) и (15.45).

Можно использовать другое правило выбора элемента, основанное на оценке числа связей не размещённого элемента $e_i \in E'_k$ как с размещенными, так и с неразмещенными элементами:

$$c_i = \sum_{e_j \in E_k} r_{ij} - \sum_{e_j \in E'_k} r_{ij}. \quad (15.46)$$

В этом случае выбирается элемент с максимальным значением c_i (15.46). Заметим, что данный выбор аналогичен принципу максимальной конъюнкции – минимальной дизъюнкции, применяемому в алгоритмах компоновки.

К этому же типу относится характеристика относительной связности

$$c_i = \sum_{e_j \in E_k} r_{ij} / \sum_{e_j \in E_k} r_{ij}.$$

На очередном шаге алгоритма размещается элемент, имеющий максимальный коэффициент относительной связности.

Так как при размещении элементов число соединений между отдельными группами элементов не может быть строго определено (оно зависит от способа последующей трассировки соединений), то в некоторых алгоритмах используются вероятностные оценки для подсчета меры связности.

Выборанный для размещения элемент e_{i_0} должен быть установлен в одну из незанятых позиций из множества L'_k . Эта позиция выбирается с учетом минимизации критерия размещения, в частности, МСД соединений. Естественно, что при последовательном процессе размещения может быть оценена лишь суммарная длина частичных монтажных соединений данного элемента e_{i_0} с уже размещенными элементами E_k .

В принципе возможен подход, когда при установке элемента в позицию рассчитываются трассы соответствующих соединений. Длина этих соединений является критерием для выбора позиции. Однако большие затраты машинного времени делают этот подход нереальным, по крайней мере, при конструировании узлов с печатными соединениями, и ограниченно применимым при конструировании монтажных схем проводных соединений.

Учитывая сказанное, используют приближенные оценки длин монтажных соединений. Наиболее простая из них и наименее точная сводится к расчету расстояний между позициями.

Метод обратного размещения

В методе обратного размещения осуществляется предварительная оценка каждого из размещаемых элементов e_1, e_2, \dots, e_n и каждой свободной позиции l_1, l_2, \dots, l_n , после чего все элементы размещаются одновременно.

Пусть дана матрица соединений $R = |r_{ij}|_{n \times n}$ и матрица расстояний между позициями $D = |d_{ij}|_{n \times n}$.

Этапы метода обратного размещения:

1) Для каждого элемента e_i по матрице R найдем суммарное число соединений этого элемента с остальными элементами:

$$r_i = \sum_{j=1}^n r_{ij}, \quad i = 1, 2, \dots, n. \quad (15.47)$$

2) Для каждой позиции l_i по матрице D найдем характеристику

$$d_i = \sum_{j=1}^n d_{ij}, \quad i = 1, 2, \dots, n, \quad (15.48)$$

определяющую суммарное расстояние этой позиции до остальных позиций.

Позиции в центральной части КП имеют меньшую характеристику d_i , чем позиции на периферии. Естественно, что центральные позиции наиболее благоприятны для размещения сильно связанных элементов, т. е. элементов, имеющих большее значение r_i . Рассматривая с этой точки зрения выражение (15.47) и учитывая условия минимальности скалярного произведения $r \cdot d$, получаем следующее:

3) Упорядочить элементы по возрастанию характеристики

$$r_i \quad (r_{i_1} \leq r_{i_2} \leq \dots \leq r_{i_n}).$$

4) Упорядочить позиции по убыванию характеристики

$$d_{ij} \quad (d_{i_1} \geq d_{i_2} \geq \dots \geq d_{i_n}).$$

5) Определить размещение $p(i_k) = j_k, k = 1, 2, \dots, n$.

Описанный метод отличается от большинства алгоритмов размещения чрезвычайной простотой. Он может быть использован даже при ручной методике разработки схем для грубого начального варианта размещения элементов. Опыт использования данного алгоритма показывает, что он, как правило, дает решение, несколько уступающее по качеству размещению, выполненному опытным конструктором, когда как последовательные алгоритмы дают сокращение длины соединений на 10–15 %.

Структура итерационных алгоритмов

Алгоритмы данной группы используют общие идеи методов последовательных приближений и являются комбинаторными аналогами градиентных методов оптимизации. Для этих алгоритмов необходимо задать начальный вариант размещения. Итерационные алгоритмы применяются для решения задачи размещения с различными критериями оптимизации $F(p)$: суммарная длина соединений, суммарное число пересечений соединений и т. д. В любом итерационном алгоритме исследуется некоторое подмножество размещений, в определенном смысле близких к начальному, для выделения в нем размещения с меньшим значением функции-критерия. Найденное размещение вновь принимается за начальное, и процесс повторяется. Алгоритм завершается при отыскании такого размещения, в окрестности которого отсутствуют варианты с меньшим значением функции-критерия. В большинстве случаев этот процесс приводит к получению локального минимума функции $F(p)$.

Пусть $F(p)$ – некоторая функция-критерий размещения, а $p_{\text{нач}}$ – начальное размещение. Тогда в результате применения итерационного алгоритма размещения получается последовательность размещений $p_{\text{нач}}, p_1, p_2, \dots, p_*$, которой соответствует монотонно убывающая последовательность значений $F(p_{\text{нач}}) > F(p_1) > F(p_2) > \dots > F(p_*)$. Значение $F(p_*)$, вообще говоря, соответствует локальному минимуму функции. Однако практика применения подобных алгоритмов показывает, что получаемые размещения близки к оптимальным. Это, по-видимому, связано с тем, что используемые критерии оптимизации являются относительно пологими функциями, не имеющими «острых» экстремумов. Характерной чертой итерационного алгоритма является возможность получения варианта размещения в любой момент итерационного процесса. Поэтому при реализации алгоритма на ЭВМ, как правило, итерационный процесс заканчивается, как только разность значений функции-критерия для двух соседних итераций становится относительно малой:

$$[F(p_k) - F(p_{k-1})] / F(p_k) < \delta, \quad (15.49)$$

где δ – заранее заданное число.

Различные итерационные алгоритмы размещения имеют сходную структуру, содержащую такие элементы:

- 1) преобразование очередного размещения;
- 2) вычисление функции размещения;
- 3) выбор лучшего размещения на основе п. 2);
- 4) переход к следующей итерации и правило остановки.

В качестве начального может быть взято размещение, полученное одним из конструктивных алгоритмов размещения или заданное конструктором.

Назовем окрестностью очередного размещения множество вариантов, рассматриваемых при выборе лучшего размещения. Иногда это размещения, получаемые из данного попарными или групповыми перестановками элементов. В других случаях – это размещения, получаемые при оптимизации мест расположения отдельных элементов, и т. д. В любом варианте должно быть определено правило выбора размещения с меньшим значением $F(p)$. При этом могут быть использованы различные тактики, например выбор на основе максимального уменьшения $F(p)$ или выбор первого размещения с меньшим значением $F(p)$ и т. д.

Отметим, что из-за отсутствия теоретических оценок эффективности различных итерационных алгоритмов невозможно отдать явное предпочтение одному из них.

Алгоритмы парных перестановок

Алгоритмы парных перестановок являются простейшими в рассматриваемом классе алгоритмов размещения.

Пусть имеется некоторое размещение (начальное или результат предыдущей итерации). Выбираются два элемента e_i и e_j , и меняются местами. Рас-

считывается новое значение $F(p)$; если оно меньше прежнего, то производится обмен. Выбирается другая пара элементов и осуществляется аналогичная процедура. Процесс итеративно продолжается до тех пор, пока не будет применено используемое в алгоритме правило остановки.

Рассмотрим минимизацию суммарной длины соединений

$$F(p) = \sum_{i=1}^n \sum_{j=i+1}^n r_{ij} d_{p(i)p(j)}, \quad (15.50)$$

где $R = ||r_{ij}||_{n \times n}$ и $D = |d_{ij}|_{n \times n}$ – соответственно матрицы соединений и расстояний. Найдем приращение значения функции (15.50) при перестановке местами элементов e_i и e_j , находящихся первоначально в позициях h и k соответственно.

На рис.15.9 показано положение этих элементов до и после перестановки, а также их связи с некоторым элементом $e_s (s \neq i \neq j)$, не участвующим в перестановке и находящимся в позиции $p(s)$.

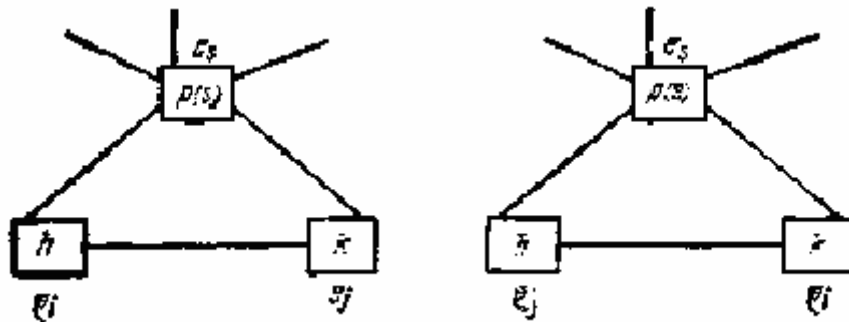


Рис. 15.9

Если длину соединений между всеми парами элементов, не затрагиваемых данной перестановкой, обозначить через C , то получим следующее выражение суммарной длины соединений:

$$F = \sum_s r_{is} d_{hp(s)} + r_{ij} d_{hk} + \sum_s r_{js} d_{kp(s)} + C. \quad (15.51)$$

После перестановки местами элементов e_i и e_j значение суммарной длины изменится и станет равным

$$F' = \sum_s r_{js} d_{hp(s)} + r_{ij} d_{hk} + \sum_s r_{is} d_{kp(s)} + C. \quad (15.52)$$

Сравнивая (15.51) и (15.52), легко рассчитать приращение функции $\Delta F_{ij} = F - F'$. После некоторых преобразований получим

$$\Delta F = \sum_s (r_{is} - r_{js})(d_{hp(s)} - d_{kp(s)}), \quad s \neq i, j. \quad (15.53)$$

Алгоритмы парных перестановок для минимизации суммарной длины соединений отличаются как способом определения исследуемой окрестности очередного варианта размещения, так и стратегией выбора следующего варианта с меньшим значением длины соединений. Очередная пара элементов для перестановки выбирается либо случайно, либо, что более эффективно, некоторым систематическим образом.

Можно, например, вначале все элементы упорядочить в соответствии с убыванием характеристики

$$r_i = \sum_{j=1}^n r_{ij}, \quad i = 1, 2, \dots, n, \quad (15.54)$$

которая рассчитывается по матрице соединений R и определяет количество соединений элемента e_i с остальными. В результате можно найти последовательность номеров элементов:

$$I = i_1, i_2, \dots, i_n \quad (r_{i_1} \geq r_{i_2} \geq \dots \geq r_{i_n}). \quad (15.55)$$

На очередной итерации для каждого элемента i_k в последовательности I рассчитывается приращение ΔF (15.53) при условии перестановки этого элемента с элементами, стоящими правее в последовательности I . Далее выбирается максимальное положительное значение ΔF ; и элемент i_k меняется местами с соответствующим элементом последовательности. Если для данного элемента i_k в наборе $\{\Delta F_{i_k j}\}$ отсутствуют положительные приращения, он остается на своем месте и осуществляется переход к отысканию «наилучшего» места для следующего элемента i_{k+1} . Таким образом, для определения места элемента i_1 рассчитывается $n-1$ приращений, элемента i_2 рассчитывается $n-2$ приращений и т. д. Всего на одной итерации рассчитывается $n(n-1)/2$ приращений и уточняется местоположение всех n элементов.

По окончании очередной итерации получается размещение с меньшей длиной соединений и осуществляется переход к следующей итерации.

Итерационный процесс заканчивается, когда изменение общей длины соединений в соответствии с (15.50) становится относительно малым.

Можно модифицировать описанный алгоритм, учитывая изменение длин соединений отдельных элементов при перестановках. Последовательность выбора элементов для перестановки определяется не количеством соединений r_i , а их суммарной длиной:

$$F_i = \sum_{j=1}^n r_{ij} d_{p(i)p(j)}, \quad i = 1, 2, \dots, n.$$

При этом в первую очередь уточняется расположение элементов с большими значениями F_i . После каждого успешного обмена значения F_i пересчитываются, и процесс повторяется.

Алгоритмы трассировки соединений

Трассировка соединений является, как правило, заключительным этапом конструкторского проектирования электронных устройств СУ и состоит в определении линий, соединяющих эквипотенциальные контакты элементов, и компонентов, составляющих проектируемое устройство.

Задача трассировки – одна из наиболее трудоемких в общей проблеме автоматизации проектирования ЭУ СУ. Это связано с несколькими факторами, в частности с многообразием способов конструктивно-технологической реализации соединений, для каждого из которых при алгоритмическом решении задачи применяются специфические критерии оптимизации и ограничения. С математических позиций трассировка – наисложнейшая задача выбора из огромного числа вариантов оптимального решения.

Одновременная оптимизация всех соединений при трассировке за счет перебора всех вариантов в настоящее время невозможна. Поэтому разрабатываются в основном локально оптимальные методы трассировки, когда трасса оптимальна лишь на данном шаге при наличии ранее проведенных соединений.

Основная задача трассировки формулируется следующим образом: по заданной схеме соединений проложить необходимые проводники на плоскости (плате, кристалле и т. д.), чтобы реализовать заданные технические соединения с учетом заранее заданных ограничений. Основными являются ограничения на ширину проводников и минимальные расстояния между ними.

Исходной информацией для решения задачи трассировки соединений обычно являются список цепей, параметры конструкции элементов и коммутационного поля, а также данные по размещению элементов. Критериями трассировки могут быть процент реализованных соединений, суммарная длина проводников, число пересечений проводников, число монтажных слоев, число межслойных переходов, равномерность распределения проводников, минимальная область трассировки и т. д. Часто эти критерии являются взаимоисключающими, поэтому оценка качества трассировки ведется по доминирующему критерию при выполнении ограничений по другим критериям либо применяют аддитивную или мультипликативную формы оценочной функции, например следующего вида:

$$F = \sum_p \lambda_i f_i,$$

где F – аддитивный критерий; λ_i – весовой коэффициент; f_i – частный критерий; p – число частных критериев.

Известные алгоритмы трассировки печатных плат можно условно разбить на три большие группы.

1) Волновые алгоритмы, основанные на идеях Ли и разработанные Ю. Л. Зиманом и Г. Г. Рябовым. Данные алгоритмы получили широкое рас-

пространение в существующих САПР, поскольку они позволяют легко учитывать технологическую специфику печатного монтажа со своей совокупностью конструктивных ограничений. Эти алгоритмы всегда гарантируют построение трассы, если путь для нее существует.

2) Ортогональные алгоритмы, обладающие большим быстродействием, чем алгоритмы первой группы. Реализация их на ЭВМ требует в 75–100 раз меньше вычислений по сравнению с волновыми алгоритмами. Такие алгоритмы применяют при проектировании печатных плат со сквозными металлизированными отверстиями. Недостатки этой группы алгоритмов связаны с получением большого числа переходов со слоя на слой, отсутствием 100 %-й гарантии проведения трасс, большим числом параллельно идущих проводников.

3) Алгоритмы эвристического типа. Эти алгоритмы частично основаны на эвристическом приеме поиска пути в лабиринте. При этом каждое соединение проводится по кратчайшему пути, обходя встречающиеся препятствия. Возможно применение эволюционных методов для решения задачи трассировки, в частности генетических алгоритмов. Трудность заключается в постановке задачи трассировки как задачи глобальной оптимизации в терминах и обозначениях метода ГА.

Волновой алгоритм Ли является классическим примером использования методов динамического программирования для решения задач трассировки печатных соединений. Основные принципы построения трасс с помощью динамического алгоритма сводятся к следующему.

Все ячейки монтажного поля подразделяют на занятые и свободные. Занятыми считаются ячейки, в которых уже расположены проводники, построенные на предыдущих шагах, или находятся монтажные выводы элементов, а также ячейки, соответствующие границе платы и запрещенным для прокладывания проводников участкам. Каждый раз при проведении новой трассы можно использовать лишь свободные ячейки, число которых по мере проведения трасс сокращается.

На множестве свободных ячеек коммутационного поля моделируют волну влияния из одной ячейки в другую, соединяемых впоследствии общим проводником. Первую ячейку, в которой зарождается волна влияний, называют источником, а вторую – приемником волны. Чтобы иметь возможность следить за прохождением фронта волны влияний, его фрагментам на каждом этапе присваивают некоторые веса:

$$P_k = P_{k-1} + \varphi(f_1, f_2, \dots, f_g),$$

где P_k и P_{k-1} – веса ячеек k -го и $(k-1)$ -го фронтов; $\varphi(f_1, f_2, \dots, f_g)$ – весовая функция, являющаяся показателем качества проведения пути, каждый параметр которой $f_i (i = 1, 2, \dots, g)$ характеризует путь с точки зрения одного из критериев качества (длины пути, числа пересечений и т. п.). На P_k накладывают одно ограничение – веса ячеек предыдущих фронтов не должны быть больше

весов ячеек последующих фронтов. Фронт распространяется только на соседние ячейки, которые имеют с ячейками предыдущего фронта либо общую сторону, либо хотя бы одну общую точку. Процесс распространения волны продолжается до тех пор, пока её расширяющийся фронт не достигнет приемника или на Θ -ом шаге не найдется ни одной свободной ячейки, которая могла бы быть включена в очередной фронт, что соответствует случаю невозможности проведения трассы при заданных ограничениях.

Если в результате распространения волна достигла приемника, то осуществляют «проведение пути», которое заключается в движении от приемника к источнику по пройденным на этапе распространения волны ячейкам, следя за тем, чтобы значения P_k монотонно убывали. В результате получают путь, соединяющий эти две точки. Из описания алгоритма следует, что все условия, необходимые для проведения пути, закладываются в правила приписания веса ячейкам.

Чтобы исключить неопределенность при проведении пути для случая, когда несколько ячеек имеют одинаковый минимальный вес, вводят понятие путевых координат, задающих предпочтительность проведения трассы. Каждое направление кодируют двоичным числом по $\text{mod } q$, где q – число просматриваемых соседних ячеек. И чем более предпочтительно то или иное направление, тем меньший числовой код оно имеет. Например, если задаться приоритетным порядком проведения пути сверху, справа, снизу и слева, то коды соответствующих путевых координат будут 00, 01, 10, и 11. Приписывание путевых координат производят на этапе распространения волны. При проведении пути движение от ячейки к ячейке осуществляют по путевым координатам. Этапы волнового алгоритма:

1. На масштабной сетке коммутационного поля (КП) отмечают занятые и свободные дискреты (трасса может проходить только через свободные дискреты; все ранее проложенные проводники через дискреты и контактные площадки – это занятые дискреты).

2. Вводится весовая функция $F(f_1, f_2, \dots, f_n)$ как критерий качества пути (например, f_1 – допустимая длина, f_2 – число пересечений, f_3 – число изгибов, f_n – число переходных отверстий).

3. Распространение волны для каждой цепи от начала А (источника) во все стороны до конца В (приемника) заключается в присвоении дискретам, соседним с ранее рассмотренными, определенного значения весовой функции F . Процесс распространения волны продолжается до тех пор, пока расширяющийся фронт волны не достигает дискрета В (конца цепи). Это означает, что путь построить можно. Если на очередном этапе распространения волны не окажется ни одного свободного дискрета, то путь построить нельзя.

4. Строится путь трассы от дискрета В по пройденным дискретам. Для устранения неопределенности проведения трассы в случае, если несколько соседних дискретов имеют одинаковый вес, используются весовые координаты (приоритет поворотов: вверх, направо, вниз, налево), которые указывают предпочтительность направления трассы.

Рассмотрим алгоритм на примере [табл. 15.1](#).

Таблица 15.1

2	1	3	2	1	2	3	1	2	3	1	2	3	1	2	3	1
1	3	2	1	3	1	2	3	1	2	3	1	2	3	1	2	3
3	2	1	3	2	3	1	2	*	3	1	2	3	1	2	3	1
2	1	3	2	1	2	3	1	*	1	2	3	1	2	3	1	2
1	3	2	1	A	1	2	3	*	2	3	1	2	3	1	2	3
2	1	3	2	1	2	3	1	*	3	1	2	3	1	2	3	1
3	2	1	3	2	3	1	2	*	1	2	3	1	2	3	1	2
1	3	2	1	3	1	2	3	*	3	1	2	3	1	2	3	1
2	1	3	2	1	*	*	*	*	2	3	1	2	3	1	2	3
3	2	1	3	2	3	1	2	3	1	2	3	1	2	3	1	2
1	3	2	1	3	1	2	3	1	2	3	*	2	3	1	2	3
2	1	3	2	1	2	3	1	2	3	1	*	3	B	2	3	1
3	2	1	3	2	3	1	2	3	*	*	*	1	2	3	1	2
1	3	2	1	3	1	2	3	1	2	3	1	2	3	1	2	3
2	1	3	2	1	2	3	1	2	3	1	2	3	1	2	3	1

Выбираем точку A(5, 10) в качестве источника, точку B(14, 3) в качестве приемника.

Вес дискреты A равен 0. Соседним дискретам присваиваем веса, равные $i_l, 2, 3$ и т. д. (Для присвоения весов дискретам достаточно трех весов 1, 2, 3.) Распространяем волну во все стороны от клетки A до тех пор, пока фронт не достигнет точки B. Волна распространяется только по свободным дискретам (в таблице запрещенные дискреты помечены символом *). В обратном направлении строим путь от точки B до точки A, проходя дискреты в обратном направлении по весовым значениям 3–2–1–3–2–1 и так далее при условии приоритетов: вверх, влево, вниз, вправо (построенный путь отмечен жирным шрифтом).

Для повышения быстродействия волновых алгоритмов применяется одновременное распространение волны от источника и приемника до их встречи, распространение волны только в области прямоугольника, создающего искусственные границы по дискретам начала и конца цепи, распростра-

нение волны не по всему множеству клеток, а по потоковой сети, замедление волны, сжатие и растяжение сети.

Следует отметить, что на распространение волны затрачивается около 90 % времени вычислений, а на проведение цепи – 10 %. Существенными недостатками волнового алгоритма являются малое быстродействие и большой объем памяти, необходимый для хранения информации о текущем состоянии всех ячеек коммутационного поля, возможность построения лишь соединений типа «ввод-вывод». Попытки устранить указанные недостатки привели к созданию ряда модификаций волнового алгоритма.

Лучевой алгоритм трассировки предусматривает исследование свободных клеток коммутационного поля (КП) по заранее заданным направлениям – подобно лучам. Для построения пути от точки А до точки В распространяют лучи одновременно из обоих источников до встречи их в некоторой площадке С. Обычно выбирают два луча, идущих в двух направлениях из каждой точки ([табл. 15.2](#)).

Таблица 15.2

									*										
									*										
				А	→	→	→	*				*	*	*	*				
				↓			↓	*				*	↑						
				↓			↓	*				*	↑						
				↓	*	*	*	*					↑						
				↓									↑						
				↓	←	←	←	←	←	←	←	←	↑						
				↓							*	↑	↑						
				↓							*	←	В						
				↓					*	*	*								
				↓															

Обычно с помощью лучевых алгоритмов можно проложить 50–60 % трасс, остальные проводятся с помощью волнового алгоритма. Лучевой алгоритм экономичнее волнового по затратам машинного времени.

Канальные алгоритмы трассировки основаны на прокладке трасс по укрупненным дискретам коммутационного поля (КП), представляющим собой каналы. Канал характеризуется пропускной способностью – допустимым количеством проходящих через него магистралей. Каналы образуются, как правило, между рядами установленных элементов или компонентов. В вертикальных и горизонтальных каналах трассировка осуществляется в различ-

ных слоях КП. Канальные алгоритмы основаны на определении максимального потока в сети с ограниченными пропускными способностями каналов.

Планирование конкуренции различных цепей на одни и те же каналы КП позволяет существенно повысить эффективность алгоритма. Каждой из рассматриваемых цепей ставится в соответствие прямоугольник минимальной площади, охватывающий все контакты цепи. Эти прямоугольники налагаются на поверхность КП и каждому дискрету КП присваивается вес, равный числу прямоугольников, покрывающих данный участок. Затем каждой цепи присваивается вес, равный минимальному значению веса участка поля, входящего в прямоугольник этой цепи. После чего порядок трассировки определяется по уменьшению веса, т. е. цепи, полностью принадлежащие областям с наибольшей плотностью проводников, будут трассироваться в первую очередь.

При разработке топологии многослойных схем (МПП, ИС, ГИС) задаче трассировки монтажных соединений предшествует этап расслоения цепей.

Если расслоение цепей выполняется до трассировки, то производится анализ всех цепей с целью определения конфликтующих соединений, совместимости или несовместимости цепей на одном слое. Одним из способов такого анализа является решение задачи разбиения графа схемы на минимальное число подграфов с последующей реализацией этих подграфов на отдельных слоях. Более простой путь – определение степени совместимости цепей на одном слое с учетом метрических параметров: для двухслойных печатных плат (ДПП) применяется распределение горизонтальных и вертикальных соединений на разные стороны ДПП, для многослойных (МПП) строятся минимальные охватывающие прямоугольники и вычисляются коэффициенты несовместимости. Задача расслоения на заданное число слоев r сводится к получению такой раскраски графа в r цветов, при которой сумма весов ребер, соединяющих вершины одного цвета (суммарная несовместимость цепей одного слоя), минимальна.

Если расслоение цепей производится после трассировки всех цепей на одной плоскости со всеми получившимися пересечениями цепей (естественно, с соответствующей минимизацией числа пересечений, числа изгибов и длины соединений), то решается задача раскраски полученного графа схемы в r цветов с последующим заполнением слоев оставшимися соединениями и назначением контактных переходов в точках изгиба проводников.

Следует отметить, что решение задачи трассировки соединений с предварительным расслоением для многослойных печатных плат приводит к лучшему использованию КП и уменьшению числа слоев, а для ДПП наиболее эффективна организация процесса расслоения и минимизации числа переходов после трассировки всех цепей на КП.

Лекция 16. Контроль полученных конструктивных решений

Контроль соответствия конструкции исходной принципиальной электрической схеме. Анализ помехоустойчивости. Оценка тепловых режимов конструкции. Анализ механических характеристик конструкции.

Требования рынка в настоящий момент предъявляют жесткие ограничения на временные затраты, связанные с созданием электронных устройств. На [рис. 16.1](#) представлена взаимосвязь времени выхода на рынок и отладки физического прототипа: 1 – время выхода новой продукции на рынок; 2 – время, необходимое для отладки прототипа; 3 – время, отводимое для других стадий создания изделия; 4 – ситуация, при которой время отладки прототипа диктует сроки появления продукции на рынке; 5 – ситуация, при которой прототип не может быть отлажен, что препятствует выходу продукции на рынок. Аналогичную взаимосвязь времени выхода на рынок и отладки прототипа можно представить и для СУ в целом [6].

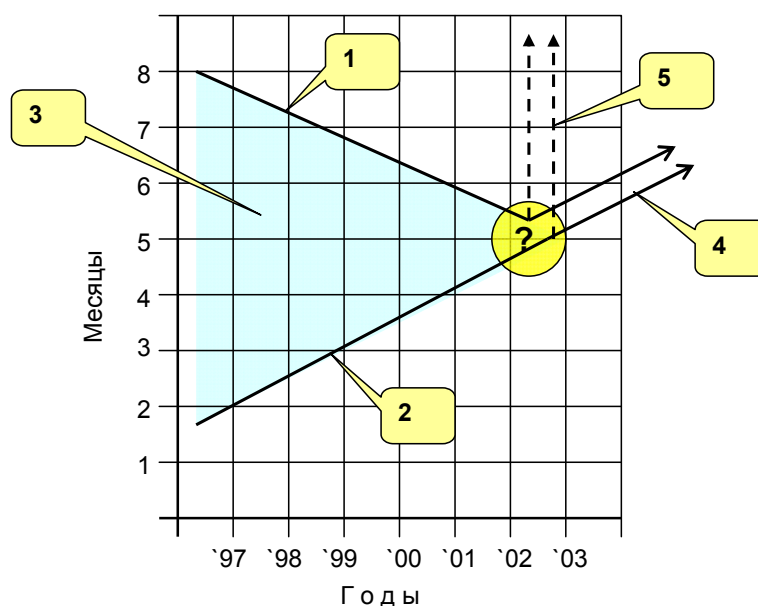


Рис. 16.1

Для ускорения процедуры проектирования и отладки прототипов необходимо применять научно обоснованные методы проектирования, адекватные модели и средства автоматизации проектирования.

При обнаружении дефекта на стадии опытной эксплуатации устройств СУ возникает необходимость повторного проектирования и изготовления, например, печатного узла, что экономически неэффективно, поэтому анализ и верификация результатов является важной задачей этапа конструкторского проектирования (относится к задачам второй группы КП).

Анализ и верификация результатов КП содержит:

- контроль соответствия конструкции исходной принципиальной электрической схеме и контроль выполнения заданных конструктивно-технологических ограничений;
- анализ помехоустойчивости;
- оценку тепловых режимов конструкции;
- анализ механических характеристик конструкции.

Контроль соответствия конструкции исходной принципиальной электрической схеме

Для контроля соответствия конструкции исходной принципиальной электрической схеме и контроля выполнения заданных конструктивно-технологических ограничений в состав САПР включаются DRC- и ERC-утилиты.

DRC-утилита (Design Rules Check) выполняет проверку печатной платы; сравнение списка соединений текущей печатной платы с принципиальной схемой; выявляет наличие нарушения зазоров; а также выполняет ограничения на ширину трасс, длины цепей и др.

ERC-утилита (Electrical Rules Check) выявляет ошибки принципиальной схемы: соединения выводов несовместимых типов; неподключенные выходы компонентов; неподключенные цепи; входящие в состав шины цепи, которые встречаются только один раз и др.

Соответствие схеме проверяется путем восстановления принципиальной схемы по топологии печатной платы.

Анализ помехоустойчивости

Задача анализа помехоустойчивости элементов и узлов непосредственно связана с этапом коммутационно-монтажного проектирования. Повышение быстродействия электронных устройств, повышение плотности цепей и разветвленности затрудняют конструктивную реализацию межэлементных связей, оптимальную в смысле помехоустойчивости, на всех уровнях конструкторского проектирования. Вместе с тем именно печатный монтаж является основным в современных конструкциях и занимает большую площадь ячеек в печатных конструкциях. Внутриблочные и межблочные связи, осуществляемые проводами и кабелями, в значительной степени определяют объем блоков и шкафов электронной аппаратуры.

Повышение роли межэлементных соединений на всех уровнях иерархии конструкции СУ выдвигает на первый план задачу анализа помех в цепях связи и сигнальных цепях. Основные причины появления помех, связанные с конструктивными особенностями реализации межсхемных соединений: ис-

кажения и задержки сигналов при прохождении по электрически «длинным» цепям – линиям связи, в которых время распространения сигнала много больше его фронта; перекрестные наводки в цепях. «Длинные» цепи характерны для печатного, межъячеечного и межстоечного монтажа.

Проблема помехоустойчивости межэлементных соединений косвенно решается на этапах размещения и трассировки введением критериев оптимальности: минимизации суммарной длины соединений, ограничения максимальной длины, ограничения числа перегибов и др. Чтобы провести конкретную оценку помехоустойчивости, нужно для разработанной конструкции узла выявить наиболее опасные (в смысле помех) цепи и сделать анализ переходных процессов в них с учетом взаимосвязи со схемными элементами. При этом методы и модели анализа помехоустойчивости можно свести к трем группам.

1. Электродинамические модели на основе решений уравнений Максвелла, которые моделируют электрические и магнитные поля в различных местоположениях в частотной или временной области.

2. Модели на основе теории цепей, которые строятся на основе решения дифференциальных уравнений, соответствуют различным элементам цепи и включают соотношения токов и напряжений, которые позволяют рассчитать напряжения и токи в различных узлах цепи в частотной или временной области.

3. Программы *поведенческого моделирования*, которые используют модели, основанные на таблицах, линиях передачи и других моделях пассивных элементов, базирующиеся на передаточных функциях, позволяющих быстро рассчитать напряжения и токи в различных узлах, как правило, во временной области.

Применимость моделей оценивается на основе «электрического размера» $k = l/\lambda$, где l – максимальный линейный размер платы (например – диагональ); λ – минимальная длина волны в спектре сигнала. При $k = 0,1$ – объект электрически мал и для его анализа можно использовать теорию цепей. В остальных случаях следует использовать полные электродинамические модели.

Например, при длине печатной платы $l = 250$ мм и при частоте 300 МГц ($\lambda = 1$ м) получаем $k = 0,25/1 = 0,25$, т. е. проектирование платы должно вестись с учетом электродинамических процессов, происходящих в них.

Верификация проектных решений позволяет установить работоспособность (в части целостности сигнала) электронного устройства СУ на виртуальном прототипе. Разновидности верификации:

- параметрическая (функциональная) – учет всего разнообразия параметров;
- временная – учет только тех параметров, которые влияют на временные соотношения (временные диаграммы);
- целостности сигнала – учет параметров, существенно влияющих на форму фронта сигнала;

- электромагнитной совместимости (ЭМС) – учет параметров, влияющих на характеристики ЭМС (помехозащищенность и помехоэмиссию).

Содержание основных задач анализа помехоустойчивости при проектировании печатных плат и печатных узлов представлено на [рис. 16.2](#). Разработка и реализация этих задач в виде компонентов виртуальных прототипов различной степени адекватности будет способствовать повышению качества проектирования, сокращению сроков разработки конкурентоспособных изделий.

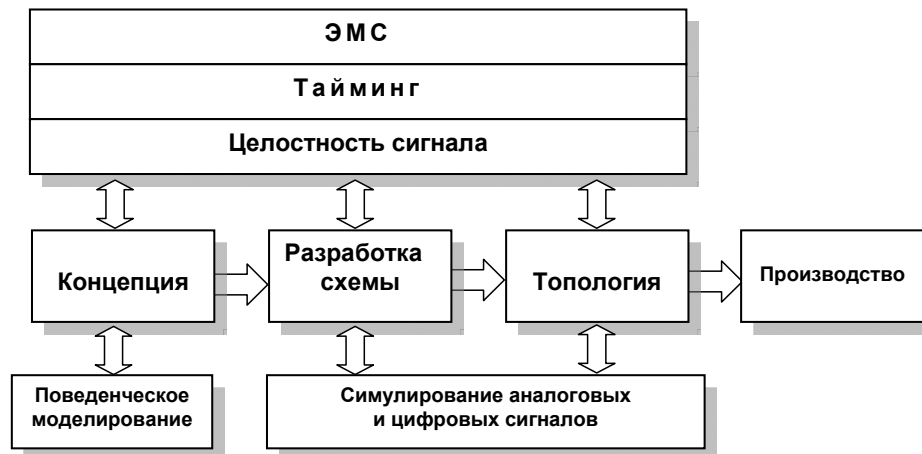


Рис. 16.2

Перечень основных задач верификации при создании электронных устройств:

Электромагнитная совместимость:

- уровень излучения от электронного модуля;
- кондуктивные помехи по шинам питания;
- восприимчивость к излучаемым помехам от внешних источников.

Тайминг:

- определение задержек распространения сигнала в линиях передачи;
- определение помех отражения и согласование линий передач;
- учет разности синхронных сигналов на выходах микросхемы;
- анализ влияния емкости и индуктивности печатных проводников на системную задержку.

Целостность сигнала:

- определение рациональной структуры многослойных печатных плат;
- анализ линий передач в виде дифференциальных пар;
- управление волновым сопротивлением линий передач;
- анализ перекрестных помех;
- анализ влияния нагрузки и согласования линии передачи;
- влияние «отрыва» заземления при работе цифровых микросхем;
- влияние потерь в линиях передачи печатных плат.

Задачи целостности сигнала предусматривают рассмотрение цифрового сигнала как аналогового с произвольными искажениями его формы, которые вызваны физическими особенностями распространения сигнала в реальной конструкции платы.

Два важных фактора влияют на рассмотрение проблем целостности сигнала:

- повышение частоты ведёт к увеличению скоростей изменения токов dI/dt и напряжений dV/dt в цепях аппаратуры; это означает, что проблемы, не оказывающие никакого влияния на низкочастотные проекты, могут иметь катастрофические последствия в проектах следующего поколения быстродействующих узлов;
- эффективное решение проблем целостности сигнала базируется на понятиях полных сопротивлений межсоединений; если мы имеем глубокое представление о полном сопротивлении и сможем установить при конструировании соответствие параметров конструкции печатной платы и соответствующих полных сопротивлений, то можно устранить проблемы целостности сигнала на этапе проектирования.

Результатами выполнения этих задач являются:

- для концептуальной стадии:
 - возможны рекомендации о реализуемости требований технического задания по быстродействию;
 - рекомендации по выбору материалов и технологии изготовления;
- для стадии схемотехнического проектирования:
 - уточнение требований к электрическим параметрам микросхем;
 - получение рекомендаций по установке помехоподавляющих элементов;
 - получение рекомендаций по выбору корпусов микросхем;
- для стадии топологического проектирования:
 - выработка топологических норм и рекомендаций для трассировки платы;
 - получение данных для расположения компонентов на плате;
 - определение требований к шинам питания и заземления и рекомендации по их расположению;
 - определение структуры МПП; определение требований к экранам и их расположению.

Для анализа *перекрестных помех* математическую модель длинной линии можно представить с помощью системы дифференциальных уравнений в частных производных (ДУЧП). Рассмотрим получение такой системы для одномерного случая. Проводник, изолированный относительно корпуса ди-

электриком, разбивается на одинаковые дифференциальные элементы Δx , которые можно представить в виде модели (рис. 16.3).

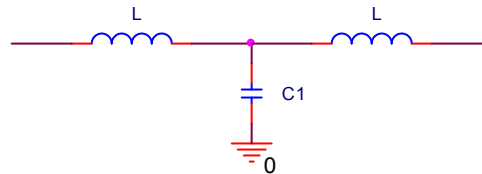


Рис.16.3

Градиенты тока и напряжения вдоль проводника: $-\frac{\partial U}{\partial x} = L \frac{\partial i}{\partial t}$; $-\frac{\partial i}{\partial x} = C \frac{\partial U}{\partial t}$, где x, t – координаты соответственно расстояния и времени; L, C – индуктивность и емкость, распределенные на единицу длины. Объединяя эти уравнения, получают телеграфное уравнение $\frac{\partial^2 U}{\partial x^2} = LC \frac{\partial^2 U}{\partial t^2}$, которое служит для математического моделирования длинной линии.

При анализе перекрестных помех в двух близко расположенных цепях наибольшее применение нашла математическая модель в виде системы ДУЧП, описывающая взаимодействие двух линий передачи с идентичными параметрами:

$$\begin{cases} \frac{\partial i_a}{\partial x} = -C \frac{\partial U_a}{\partial t} + C_B \frac{\partial U_{II}}{\partial t}; \frac{\partial i_{II}}{\partial x} = C_B \frac{\partial U_a}{\partial t} - C \frac{\partial U_{II}}{\partial t} \\ \frac{\partial U_a}{\partial x} = -L \frac{\partial i_a}{\partial t} - L_B \frac{\partial i_{II}}{\partial t}; \frac{\partial U_{II}}{\partial x} = -L_B \frac{\partial i_a}{\partial t} - L \frac{\partial i_{II}}{\partial t} \end{cases}$$

где U_a, U_{II}, i_a, i_{II} – напряжения и токи, относящиеся соответственно к активной линии, содержащей источник воздействия, и пассивной; L_B и C_B – соответственно взаимные индуктивность и емкость на единицу длины.

Основная трудность при получении описанных выше моделей связана с оперативным получением погонных значений L, C, L_B и C_B и выделением наиболее опасных цепей по результатам коммутационно-монтажного проектирования. Метод подпространства является одним из наиболее эффективных методов определения взаимных емкостей. Сущность метода заключается в разбиении всей поверхности проводника на небольшие области – подпространства. Форма подпространства выбирается обычно в виде прямоугольника с отношением длины к ширине 1:1. Точность определения взаимных емко-

стей зависит от размеров проводников, расстояния между ними и числа подпространств.

Моделирование *целостности сигнала*, выполняемое на уровне печатной платы, начинается с составления описаний электрических характеристик кристаллов микросхем, закладываемых в модель, корпусов микросхем и печатных дорожек. Целью этого этапа является получение обоснованных описаний компонентов схемы – описаний, достаточно точных для качественного моделирования и при этом не настолько сложных, чтобы с ними было трудно работать. Среди всей информации выделяют необходимые параметры. Эти параметры могут быть представлены в виде файла описания схемы в стандарте SPICE (более подходит для моделирования БИС) или файла спецификации стандарта IBIS.

Говоря о моделях целостности сигнала, мы имеем в виду кривую сигнала во времени, на которой видны «звон», перекрестные помехи или «дребезг» земли. Такие модели могут быть рассчитаны с помощью специализированного программного обеспечения, предназначенного для анализа целостности сигналов, например, программа ICX компании Mentor Graphics.

Рассмотрим недостатки алгоритмов, используемых в программах моделирования типа SPICE. Одним из факторов, существенно осложняющих работу SPICE, является сложность схемы. При моделировании реальных схем в SPICE приходится иметь дело с сотнями или тысячами узлов схемы и вольт-амперных характеристик. Для обработки таких массивов данных может потребоваться очень сложная программа, но методология моделирования и в этом случае ничем не отличается от той, которая используется для моделирования простой двухэлементной цепи. Независимо от числа узлов схемы, программа SPICE все равно работает методом задания начальных значений и подбора подходящего решения путем итерации.

Недостатки SPICE: моделирует режим работы схемы в дискретные моменты времени. Если временной шаг выбран слишком большим, то результаты не будут верными. Если временной шаг выбран слишком маленьким, то на расчет уйдет слишком много времени.

В некоторых версиях SPICE установлен пониженный предел минимально допустимого шага. Если этот предел слишком высок для расчета супербыстродействующей схемы, то данная версия SPICE не будет пригодна для ее моделирования.

На каждом шаге программа SPICE должна принимать решение: продолжить или прекратить итерационный процесс. Решение принимается путем проверки выполнения правила Кирхгофа для токов при заданной точности равенства нулю суммы токов, втекающих в узел или вытекающих из него.

Может произойти так, что итерационная процедура не обеспечит сходимости. SPICE может оказаться не в состоянии найти решение, особенно в тех случаях, когда возникают коммутационные «выбросы» напряжения на

элементах схемы или имеются изломы вольт-амперных характеристик. В таких случаях программа иногда в состоянии определить, что она работает неверно, а иногда – нет.

SPICE оперирует только теми данными, которые получает от пользователя. Если выборка параметров не учитывает всех существенных паразитных элементов схемы, то результаты моделирования, полученные SPICE, не будут верны. Эта проблема выявляется только при сравнении расчетных результатов с работой реальной схемы.

SPICE великолепно справляется с моделированием режимов работы интегральных схем. Он вполне способен моделировать работу каждого транзистора в логической схеме, учитывая даже функциональную структуру логической схемы, заложенную в кристалл. Если попытаться построить модель всей схемы на этом уровне, учитывая все до единого транзисторы в каждой из микросхем, установленных на большой плате, то моделирование может затянуться на долгое время.

Если действительно нужно моделировать каждый транзистор, то нет иного выхода, кроме как включить компьютер и оставить его «перемалывать» данные (иногда этот процесс занимает недели). Но если цель состоит всего лишь в том, чтобы промоделировать «звон» и перекрестные помехи в линиях, соединяющих интегральные схемы, то стоит подумать об использовании технологии моделирования, изложенной в стандарте I/O Buffer Information Specification (IBIS). Модель по этой технологии строится проще и рассчитывается быстрее, чем полная модель на уровне отдельного транзистора. Она обеспечивает достаточно точную оценку «звона» и перекрестных помех.

Таким образом, прохождение сигналов через активные компоненты моделировать не нужно. Сами компоненты представляются в виде своих входных и выходных импедансов – IBIS – моделей стандарта IBIS2. Эти модели автоматически подсоединяются к описанию печатной платы в соответствии с именами компонентов.

Технология IBIS реализуется на основе предварительно подготовленных таблиц вольт-амперных характеристик и ряда дискретизированных сигналов, характеризующих интегральные схемы, которые работают в общей схеме. Суть метода состоит в том, что программа последовательно проходит каждую ветвь, запуская каждый источник сигнала, рассчитывая сигналы на входах всех приемников для положительных и отрицательных переходов входных сигналов. Затем путем анализа полученного набора принятых сигналов выделяются значения выброса, «звона», времени установления и перекрестных помех в каждой ветви, соответствующие наихудшему случаю.

В состав современных САПР ЭУ входят утилиты, выполняющие анализ помехоустойчивости. Так, САПР PCAD содержит утилиту Signal Integrity, которая представляет сегменты проводников печатной платы в виде

линии передачи и моделирует переходные процессы при воздействии импульсных сигналов.

Пакет программ Omega Plus содержит модули: моделирование электромагнитного излучения ПП; анализ целостности сигналов.

Оценка тепловых режимов конструкции

Оценка тепловых режимов конструкции учитывается в критериях оптимальности при решении задач компоновки и размещения через учет допустимых мощностей рассеивания конструктивных элементов и двух рядом стоящих элементов, коэффициентов теплоотдачи элементов и др.

Тепловые модели конструкции узла получаются в результате идеализации конструкции и протекающих в ней процессов переноса теплоты. В большинстве случаев тепловые модели представлены в виде уравнений, описывающих температурное поле в конструкции. Для упрощения моделей всю конструкцию разделяют на отдельные условно изотермические участки, такое же допущение делается для температурного поля. Процессы переноса теплоты между изотермическими поверхностями описываются системой нелинейных алгебраических уравнений, которая формируется на основе тепловых эквивалентных схем конструкций. При этом используются обозначения, применяемые в электрических системах.

Анализ механических характеристик конструкции заключается в определении процессов, происходящих в конструкции при статических, линейных динамических, вибрационных и других механических нагрузках. Эта задача практически не связана с выполнением коммутационно-монтажного проектирования и выполняется в САЕ-части интегрированных САПР методом конечных элементов.

ТЕМА 7. АВТОМАТИЗАЦИЯ ИСПЫТАНИЙ ССУ

Лекция 17. Методы и алгоритмы испытаний ССУ

Испытания как часть процесса проектирования СУ. Автоматизация испытаний. Обзор современных автоматизированных систем управления испытаниями.

Испытания как часть процесса проектирования СУ

Под испытаниями в процессе проектирования понимают совокупность проектных процедур, связанных с экспериментальной проверкой макетов, опытных образцов СУ и ее устройств на соответствие требованиям ТЗ и с экспериментальной оценкой возможностей этих объектов проектирования [16].

Этап испытаний является неотъемлемой частью процесса проектирования. Испытания прототипов, макетов начинаются уже на первых витках проектирования, а испытаниями опытных образцов завершается процесс проектирования. На [рис. 17.1](#) показаны основные подэтапы испытаний и их место в процессе проектирования [4]. Испытания проводятся с первых этапов и продолжают до окончания проекта.

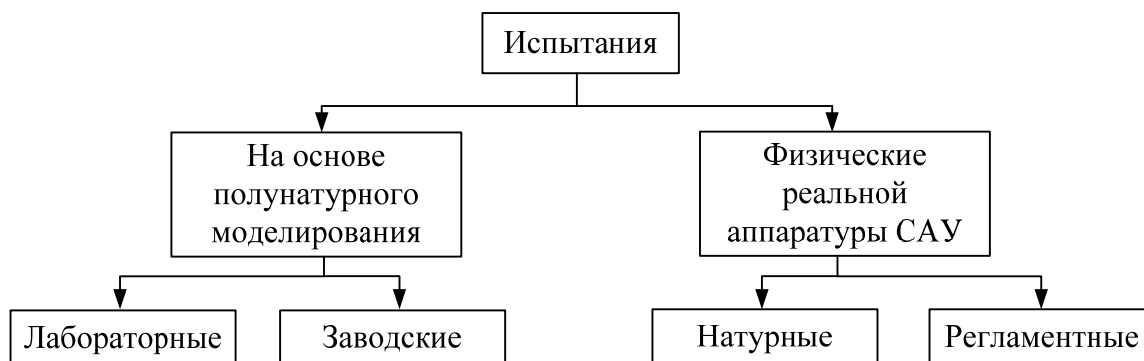


Рис. 17.1

Вначале испытаниям подвергают макетные образцы СУ и ее устройств, составленные из элементов прототипов проектируемой СУ, затем технологические образцы, изготавливаемые на опытном производстве по эскизной документации.

Во время *лабораторных испытаний* детально проверяют методами физического эксперимента (испытанием металлом) функционирование, динамические и точностные характеристики, стабильность, надежность и другие характеристики предполагаемых в проекте структур и схем СУ и ее уст-

роЙств. При этом задаются завышенные по отношению к требованиям ТЗ возмущения и условия эксплуатации. Это делается, чтобы выявить уже на ранних этапах проектирования непригодные проектные решения и в то же время, чтобы убедиться в правильности принципиальных проектных решений. Стенды с аппаратурой СУ, подлежащей лабораторным испытаниям, как правило, составляют и сохраняют в течение всего процесса проектирования. По результатам лабораторных испытаний оформляют протоколы, которые включают в отчеты по эскизному проекту и в техническую документацию проекта в целом.

Заводские испытания проводят после получения опытных образцов, в том числе готовых комплектующих СУ устройств и элементов, изготовления не выпускаемых промышленностью устройств и элементов СУ. Эти испытания осуществляют на заводе. Они предназначены для проверки функционирования СУ и всех ее характеристик после сборки и стыковки устройств системы в заводских условиях. Аналогичные испытания проводят по каждому из устройств СУ. Для СУ на этом этапе важнейшее значение имеет соединение (стыковка) отдельных устройств. Сложные СУ нуждаются в совместных испытаниях входящих в них устройств, так как эти устройства проектируют и изготавливают на разных предприятиях (для СУ ЛА, например, это навигационные приборы, БЦВМ, гидроприводы; для СУ турбогенератора – электрические машины, микропроцессоры, насосы, сервоприводы). Такие испытания, называемые также *межведомственными*, служат для выявления работоспособности СУ в целом, установления соответствия ТЗ стыковочным характеристикам устройств СУ, локализации мест неисправностей и т. д.

Протоколы заводских испытаний входят в техническую документацию проекта.

Завершают этот этап *натурные испытания СУ*. Так как при всех перечисленных видах испытаний не удастся воспроизвести реальную обстановку эксплуатации СУ (полет ЛА, работу турбогенератора в городской сети и т. д.), то во время таких испытаний, проводимых специальной комиссией с участием заказчика и государственной приемки, СУ работает в тех же условиях и на тех же объектах, для которых она проектируется (полет опытного самолета, движение шагающего робота по полной программе, работа ТГ под нагрузкой и т. д.).

Протоколы результатов натурных испытаний являются одним из основных документов проекта. В процессе серийного производства и эксплуатации СУ она подвергается также периодическим контрольным и *регламентным испытаниям* для установления ее работоспособности и прогнозирования возможных дефектов на заданный период работы. По результатам всех видов испытаний вводятся многочисленные «обратные связи» в процессе проектирования – исправляют и корректируют техническую документацию, отбрасывают неверные проектные решения.

Испытания – весьма трудоемкий и дорогой этап проектирования, в котором помимо проектировщиков-испытателей «участвуют» контрольно-измерительная аппаратура, опытное производство и заводы-изготовители, специально создаваемые для испытаний подвижные объекты (ЛА, суда, колесные машины). После испытаний, если они прошли успешно, проект (вся техническая документация и опытные образцы СУ) поступают в серийное производство, на первых порах в основном мелкосерийное и среднесерийное.

Автоматизация испытаний

Автоматизация испытаний имеет цели:

- разработка инструментов САПР СУ проектировщика-испытателя на всех этапах проектирования;
- проведение испытаний в режимах, максимально приближенных к реальным условиям эксплуатации СУ;
- сокращение расходов на испытания и сроков получения информации по результатам испытаний;
- проведение идентификации проектируемых СУ и их устройств;
- повышение производительности труда и улучшение условий труда испытателей.

Анализ процесса проектирования СУ показывает, что наибольший вес по трудоемкости и значимости имеют проектные процедуры, связанные с испытаниями макетов и опытных образцов. Возросшие требования к СУ, связанные с ростом их сложности и выполняемой функциональной нагрузки, приводят к значительному увеличению и усложнению испытаний. Объем испытаний современных ЛА на порядок превышает объем таких испытаний, проводившихся в 1950-е годы. При этом наибольший вес имеют натурные испытания СУ в составе разнообразного оборудования судов, космических аппаратов и других подвижных объектов. Аналогично состояние при испытаниях СУ турбогенератора, шагового робота и др. Расходы ресурсов на натурные испытания очень велики (построение специальных ЛА, судов для испытаний, создание специальной контрольно-измерительной аппаратуры, многочисленный персонал специалистов-испытателей и т. д.). В соответствии с этим автоматизация испытаний СУ направлена на снижение затрат на испытания и исключение из них наиболее дорогих и тяжелых проектных процедур.

Все применяемые методы испытаний можно разделить на две группы ([рис. 17.1](#)): физические испытания реальной аппаратуры (макеты, экспериментальные и опытные образцы) и испытания на основе полунатурного моделирования.

Способы физических испытаний реальной аппаратуры СУ существенно зависят от типа СУ и условий ее эксплуатации, а применение средств САПР сводится в основном к обработке результатов таких испытаний.

Основной способ испытаний, применяемый в подсистемах САПР, – полунатурное моделирование. Организация испытаний СУ на основе полунатурного моделирования отвечает поэлементному комплексированию устройств в систему, так что можно провести оценку каждого из устройств в отдельности, прежде чем они будут объединены с другими. При этом схема испытаний СУ одновременно использует адекватные математические модели части элементов и устройств испытываемой СУ наряду с ее реальной аппаратурой, макетами, экспериментальными и опытными образцами.

Испытание СУ с участием в контуре управления человека, например, СУ ЛА, позволяет оценить взаимодействие СУ с человеком-оператором, определить величины рабочих нагрузок на оператора, построить тренажер для обработки действий оператора. Программа летных испытаний СУ ЛА обычно на этапе ее проектирования ограничена несколькими летными экспериментами, поэтому применение полунатурного моделирования позволяет при планировании этой программы достичь оптимального распределения работ между испытателями СУ ЛА с использованием моделей и натурными (летными) испытаниями. По данным фирмы «Боинг», натурные испытания удастся сократить на 30 % за счет полунатурного моделирования.

Полунатурное моделирование является ведущей проектной процедурой испытаний. Это моделирование выполняется на специальных программно-аппаратных моделирующих комплексах (ПАМК) в составе САПР. Состав ПАМК формируется в зависимости от объекта испытаний. Важным принципом, закладываемым в организацию испытаний СУ с помощью полунатурного моделирования, является принцип испытаний на основе комплексного воздействия возмущений.

Оценка результатов комплексных лабораторных и заводских испытаний, их устройств и элементов обычно проводится на основе суперпозиции реакций на отдельные возмущающие воздействия, такие как удар, вибрация, качка, перегрузка и др. Эти воздействия воспроизводятся при так называемых частичных испытаниях. Но здесь весьма сложно оценить степень приближения таких испытаний к реальным условиям работы СУ, когда все возмущения действуют одновременно.

Комплексные испытания иногда проводят по схеме простого механического соединения моделирующих стендов (МС). В то же время комплексные испытания оказываются возможными на основе ПАМК, что связано с заданием части возмущающих воздействий непосредственно в СУ «электрически» (программно), а часть – «механически» через МС (имитированием качки, вибрации, ударных воздействий и др.). Однако в такой схеме невозможно практически оценить достоверность испытаний из-за неконтролируемого взаимодействия отдельных механических элементов испытательного оборудо-

дования и сложности соблюдения условий подобия. В то же время комплексные испытания оказываются возможными на основе ПАМК. Исходным является то обстоятельство, что часть возмущающих воздействий можно задавать непосредственно в СУ «электрически», используя программную реализацию возмущений, которые получены по априорным данным при натуральных и частичных испытаниях, а часть – «механически» через МС.

Алгоритм испытаний СУ по схеме [рис. 17.2](#) строится на основе эквивалентирования электрическими возмущениями механических, гидравлических, тепловых воздействий на устройства СУ со стороны внешней среды и внутренних источников. Применение принципа испытаний на комплексное воздействие может выявить скрытые при частичных испытаниях дефекты СУ и осуществить полную имитацию внешних условий. Стандартизация таких испытательных процедур позволит решить на первых этапах проектирования многие вопросы, которые проявляются зачастую только на стадии натуральных испытаний.

Автоматизация испытаний на основе полунатурного моделирования сводится к формированию возмущающих воздействий, задаваемых на реальную аппаратуру (РА) СУ через МС или вводимых в нее «электрически»; измерению сигналов о состоянии СУ, снимаемых с датчиков «промежуточных точек» СУ; сбору и накоплению данных испытаний, обработке результатов и изготовлению проектных документов по испытаниям. В состав САПР СУ должны входить программные средства, обеспечивающие комплексную автоматизацию (на этапах подготовки, проведения и анализа) испытаний от элементов и устройств до СУ в целом. В отличие от обслуживающих и проектирующих подсистем САПР в состав подсистемы испытаний должны входить базовые инструменты испытателя программно-технические средства: для формирования возмущающих воздействий; для управления испытаниями; для обработки результатов испытаний.

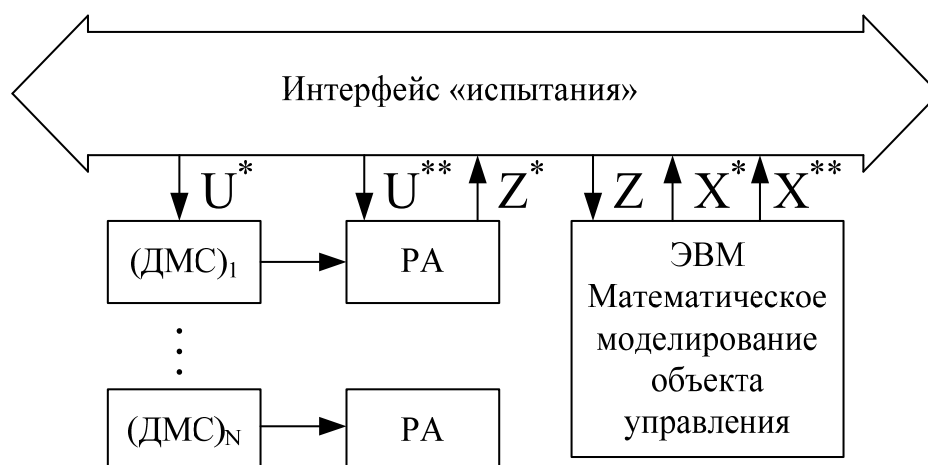


Рис. 17.2

Для формирования возмущающих воздействий, поступающих на входы МС из технических средств, в которых выполняется математическое моделирование объекта управления, приходится проводить преобразование координат. Так, при воспроизведении с помощью МС физической качки ЛА с большими углами разворота по курсу необходима реализация алгоритмов преобразования координат [4]:

$$\psi_C = \arcsin\left(\frac{\cos \mathcal{G}_0 \cos \psi_0}{\cos \psi_C}\right);$$

$$\mathcal{G}_C = \arccos\left(\frac{\cos \mathcal{G}_0 \cos \psi_0}{\cos \psi_C}\right);$$

$$\gamma_C = \arcsin\left(\frac{\cos \gamma_0 \sin \mathcal{G}_0 \sin \psi_0 + \sin \gamma_0 \sin \psi_0}{\cos \psi_C}\right);$$

где $\psi_C, \gamma_C, \mathcal{G}_C$ – углы поворота стенда; $\psi_0, \gamma_0, \mathcal{G}_0$ – углы поворота объекта, например, углы курса, тангажа и крена ЛА, $\psi_C \neq \pi/2$.

Алгоритм формирования детерминированных испытательных воздействий, составленных с учетом собственного движения МС, имеет вид

$$\varphi_C(t) = \varphi_{BX}(0)h(t) + \int_0^t \Omega_{BX}(\tau)h(t-\tau)d\tau$$

$$\Omega_C(t) = \varphi_{BX}(0)\dot{h}(t) + \Omega_{BX}(0)h(t) + \int_0^t h(\tau)\varepsilon_{BX}(t-\tau)d\tau;$$

$$\varepsilon_C(t) = \varphi_{BX}(0)\ddot{h}(t) + \Omega_{BX}(0)\dot{h}(t) + \varepsilon_{BX}(0)h(t) + \int_0^t \dot{h}(\tau)\dot{\varepsilon}_{BX}(t-\tau)d\tau,$$

$\varphi_{BX}(0) = \text{const}, \Omega_{BX}(0) = \text{const}, \varepsilon_{BX}(0) = \text{const}$ – начальные значения входного сигнала на МС φ_{BX} , его скорости Ω_{BX} и ускорения ε_{BX} ; $h(t), \dot{h}(t), \ddot{h}(t)$ – импульсная переходная функция и ее производные.

Методы и алгоритмы обработки результатов испытаний аналогичны методам обработки результатов анализа СУ.

Методы и алгоритмы управления испытаниями служат для управления процессами испытаний в реальном масштабе времени, т. е. в темпе работы реальной аппаратуры (РА). Эти методы и алгоритмы включают в себя логические операции по синхронизации взаимодействия РА, вычислительной техники, МС, различных имитаторов и преобразователей, участвующих в испытаниях.

Обзор современных автоматизированных систем управления испытаниями

Среди современных автоматизированных систем, позволяющих проводить полунатурное моделирование, следует отметить системы MATLAB, LABView, отраслевые системы управления испытаниями [3; 21].

Среда графического программирования National Instruments *LabVIEW* предназначена для создания автоматизированных систем сбора данных и управления приборами, применяется как в исследовательских и испытательных лабораториях, так и на технологических производственных линиях [20].

Данная платформа полностью перекрывает потребности трех базовых областей применений:

- автоматизированные системы измерения и тестирования;
- промышленные системы контроля и управления;
- проектирование и отладка встраиваемых систем.

С помощью пакета NI SignalExpress возможно интерактивно использовать виртуальные приборы для проведения необходимых измерений и анализа сигналов.

С помощью пакета NI TestStand можно разработать структурированную последовательность испытаний, представляющих собой отдельные программы LabVIEW (либо модули других систем программирования), со сложной логикой принятия решений «тест прошел/не прошел» для управления общим ходом испытания.

Кроме того, NI TestStand легко интегрируется в единые информационные системы предприятия (ERP) для предоставления результатов в базы данных, либо для отслеживания испытуемых изделий через автоматизированные системы управления производством (MES).

Для проведения автоматизированных измерений LabVIEW содержит пакеты анализа, оптимизированные для различных измерительных задач:

- тестирование линий связи;
- измерение виброакустических сигналов;
- мониторинг состояния машин и механизмов – специализированные алгоритмы порядкового анализа вращающихся частей механизмов (взйвлет-анализ, совместный частотно-временной анализ);
- обработка изображений – средства для автоматизированного визуального контроля и приложений машинного зрения.

Разработана платформа для промышленных системы измерения и управления на базе LabVIEW. Для создания приложений управления LabVIEW содержит отдельный набор специализированных библиотек, дополняющих графическую платформу методами управления, функциями распределенного мониторинга и управления, АСУТП, а также возможностями управления в реальном времени.

Достоинства LabVIEW:

- полноценный язык программирования;

- интуитивно понятный процесс графического программирования;
- широкие возможности сбора, обработки и анализа данных, управления приборами, генерации отчетов и обмена данных через сетевые интерфейсы;
- драйверная поддержка более 2000 приборов;
- возможности интерактивной генерации кода;
- высокая скорость выполнения откомпилированных программ;
- совместимость с операционными системами Windows2000/NT/XP, Mac OS X, Linux и Solaris.

В ОАО «Информационные спутниковые системы» (г. Железногорск) разработана и используется система управления электрическими испытаниями спутников «РАСКАТ» [18].

В ходе создания спутников связи и навигации одним из важнейших является этап комплексных электрических испытаний. Цель испытаний – проверка работоспособности и логики функционирования объекта контроля (ОК) во всех штатных и аварийных режимах всех комплектов бортовой аппаратуры (БА). Идеология проведения испытаний заключается в выдаче на ОК управляющих воздействий и контроле параметров функционирования ОК. Задачи, решаемые на этом этапе, достаточно сложны и объёмны. Соответственно и к средствам автоматизации (и, в частности, к программному обеспечению) предъявляются объёмные комплексные требования.

В качестве ОК может выступать собственно космический аппарат (КА), составная часть КА, БА, контрольно-проверочная аппаратура (КПА) или вспомогательное оборудование, которое можно абстрагировать следующим образом. Логика функционирования ОК через совокупность интерфейсов, обеспечивающих доступ к ОК, оценивается через текущее состояние ОК по вектору характеризующих параметров и управляется вектором команд. Исходя из указанного представления ОК и цели испытаний разрабатываются унифицированные аппаратно-программные средства (АПС), обеспечивающие подготовку, проведение и мониторинг испытаний широкого класса ОК. В ОАО «Информационные спутниковые системы» подобной системой является автоматизированная система испытаний и телеметрического обеспечения (АСУИТО) «РАСКАТ».

Структурно программные средства обеспечения испытаний АСУИТО «РАСКАТ» состоят из нескольких больших программных комплексов. Комплекс подготовки данных испытаний (КПДИ) обеспечивает настройку конкретного рабочего места испытаний на интерфейсы и логику функционирования заданного ОК. Комплекс включает в себя редакторы базы команд и параметров, вторичных параметров, мнемосхем мониторинга и управления, циклограмм испытаний на проблемно-ориентированном языке, а также программы, контролирующие все подготовленные данные на непротиворечивость, полноту и т. д.

Комплекс программ нижнего уровня (КПНУ) и программное обеспечение различных КПА обеспечивают преобразование разнообразных нестандартных интерфейсов связи с ОК в унифицированный интерфейс связи

с комплексом автоматизированного проведения испытаний (КАПРИ). КАПРИ обеспечивает все функции автоматизации проведения и мониторинга испытаний в реальном времени

При проведении испытаний КАПРИ осуществляет:

- автоматизированный сбор информации о состоянии ОК посредством КПНУ (период опроса 0,5 с, количество параметров до 10 000);
- обработку принятой информации;
- управление ОК ручное и автоматическое (параллельное выполнение до 10 циклограмм);
- дежурный контроль параметров (автоматизированный мониторинг до 2000 параметров);
- отображение мнемосхем, графиков и т. д.;
- протоколирование хода работы и функционирования системы;
- архивирование протокола, обработанной информации (архив параметров), необработанной информации (принятых пакетов), принятых отчётов, сформированных массивов.

Комплекс мониторинга испытаний (КМИ) обеспечивает удалённый мониторинг испытаний как в реальном времени (до 20 терминалов), так и постсессионный (без ограничений).

Для обеспечения испытаний широкого круга ОК принят следующий подход к унификации. Программное обеспечение является единым для всех рабочих мест (РМ), входные данные (ВД) для ПО индивидуальные, т. е. отдельный комплект ВД для проведения испытаний каждого ОК на соответствующем РМ.

При разработке ПО принят принцип «поглощения». Это означает, что при разработке очередного РМ, если существующие возможности ПО не удовлетворяют требованиям РМ и ОК, то оно дорабатывается и распространяется на все РМ всех ОК.

Масштабируемость ПО обеспечивается ВД, которые являются индивидуальными для каждого ОК и отражают его информационную мощность (количество параметров и команд, информативность), конфигурацию АПС нижнего уровня. А также необходимость использования для каждого конкретного РМ того или иного комплекса программ (КПНУ, КМИ).

Таким образом, АСУИТО «РАСКАТ» является мощным унифицированным программным средством, обеспечивающим комплексную автоматизацию (на этапах подготовки, проведения и анализа) испытаний широкого класса изделий от систем – приборов и компонентов до КА в целом.

ЗАКЛЮЧЕНИЕ

В курсе лекций рассмотрены основные подходы, применяемые при формировании математического описания средств и систем управления (СУ), использования современных методов их анализа и синтеза на этапах функционального и конструкторского проектирования.

Современные технические системы не поддаются исследованию (в нужной полноте и точности) обычными теоретическими методами. Прямой натурный эксперимент над ними может быть долог, дорог и зачастую невозможен по разным причинам. Цена ошибок и просчетов при проектировании недопустимо высока. В этой связи значительно повышается роль математического описания систем управления с целью получения эффективных надежных алгоритмов и программ реализации представления систем управления (СУ) и их анализа при проектировании с применением интегрированных САПР.

Математическое (информационное) моделирование СУ становится неизбежной составляющей научно-технического прогресса, основой математического обеспечения CALS-технологий в рамках получения единой информационной модели СУ в процессе проектирования и жизненного цикла СУ в целом.

Технические средства и системы управления на современной элементной базе являются трудноформализуемыми объектами и требуют применения методов анализа и синтеза, основанных на вновь создаваемых методах, например, эволюционных алгоритмах, в сочетании с классическими функциональным анализом, теорией множеств и др.

При этом, как подтверждено отечественным и мировым опытом, необходимым условием широкого и эффективного внедрения методов и средств автоматизации является простота и доступность их использования проектировщиком. Поэтому дальнейшим направлением развития вопросов автоматизированного проектирования СУ будут вопросы адаптации методов теории решения изобретательских задач, искусственного интеллекта к решению задач параметрического и структурного синтеза, задач конструкторского проектирования устройств СУ и систем в целом через разработку интеллектуальных компонентов в составе САПР.

СПИСОК ЛИТЕРАТУРЫ

1. Ли, К. Основы САПР (CAD/CAM/CAE) / К. Ли. – СПб. : Питер, 2004. – 560 с.
2. Норенков, И. П. Информационная поддержка наукоемких изделий . CALS-технологии / И. П. Норенков, П. К. Кузьмик. – М. : Изд-во МГТУ, 2002. – 320 с.
3. Латышев, П. Н. Каталог САПР. Программы и производители / П. Н. Латышев. – М. : Солон-Пресс, 2006. – 608 с.
4. Сольнищев, Р. И. Автоматизация проектирования систем автоматического управления : учебник для вузов / Р. И. Сольнищев. – М. : Высш. шк., 1991. – 335 с.
5. Афанасьев, В. Н. Математическая теория конструирования систем управления / В. Н. Афанасьев, В. Б. Колмановский, В. Р. Носов. – М. : Высшая школа, 1998. – 574 с.
6. Норенков, И. П. Основы теории и проектирования САПР / И. П. Норенков, В. Б. Маничев. – М. : Высш. шк., 1990. – 335 с.
7. Корячко, В. П. Теоретические основы САПР / В. П. Корячко, М. М. Курейчик. – М. : Энергоатомиздат, 1987. – 400 с.
8. Рождественский, Д. А. Автоматизация проектирования систем и средств управления : учеб. пособие Ч. 1 / Д. А. Рождественский. – Томск : Том. межвуз. центр дистанционного образования, 2004. – 67 с.
9. Рождественский, Д. А. Автоматизация проектирования систем и средств управления : учеб. пособие Ч. 2 / Д. А. Рождественский. – Томск : Том. межвуз. центр дистанционного образования, 2004. – 131 с.
10. Норенков, И. П. Основы автоматизированного проектирования / И. П. Норенков. – М. : Изд-во МГТУ, 2000. – 360с.
11. Самарский, А. А. Математическое моделирование : Идеи. Методы. Примеры / А. А. Самарский, А. П. Михайлов. – М. : Физматлит, 2005. – 320 с.
12. Зарубин, В. С. Математическое моделирование в технике : учебник для вузов / В. С. Зарубин; под ред. В. С. Зарубина, А. П. Крищенко. – М. : Изд-во МГТУ, 2001. – 496 с.
13. Егупова, Н. Д. Методы современной теории управления / под ред. Н. Д. Егупова. – М. : Изд-во МГТУ, 2000. – 748 с.
14. Дьяконов, В. П. MATLAB 6.5 SP1/7/7 SP1/7 SP2 + Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики / В. П. Дьяконов, В. В. Круглов. – М. : Солон-Пресс, 2006. – 456 с.
15. Джонс, М. Т Программирование искусственного интеллекта в приложениях / М. Тим Джонс. – М. : ДМК Пресс, 2006. – 312 с.
16. Дьяконов, В. П. Matlab. Анализ, идентификация и моделирование систем : специальный справочник / В. П. Дьяконов. – СПб. : Питер, 2002. – 370 с.
17. Кофанов, Ю. Н. Автоматизация проектирования РЭС. Топологическое проектирование печатных плат : учеб. пособие / Ю. Н. Кофанов,

- А. В. Сарафанов, С. И. Трегубов. – Красноярск : ИПЦ КГТУ, 2000. – 215 с.
18. Баев, М. Ю. Программное обеспечение автоматизации испытаний спутников связи и навигации [Электронный ресурс] / М. Ю. Баев, А. В. Барков, В. А. Дубенко, А. А. Киселёв, А. А. Колташев, С. Г. Кочура, В. В. Кривцов, В. В. Максимов, Л. В. Федоренко // Электрон. дан. Режим доступа : http://glonass.npopm.ru/?cid=td3_11 Загл. с экрана.
19. Джонс, Дж. К. Методы проектирования / Дж. К. Джонс. – М. : Мир, 1986. – 450 с.
20. LabVIEW.ru - National Instruments // Режим доступа : www.labview.ru
21. Сообщество пользователей Matlab и Simulink // Режим доступа: <http://www.exponenta.ru>
22. Тищенко, Н. М. Введение в проектирование систем управления / Н. М. Тищенко. – М. : Энергоатомиздат, 1986.
23. Норенков, И. П. Основы теории и проектирования САПР / И. П. Норенков, В. Б. Маничев. – М. : Высш. шк., 1990. – 335 с.
24. Норенков, И. П. Введение в автоматизированное проектирование технических устройств и систем : учеб. пособие для вузов / И. П. Норенков. – М. : Высш. шк., 1986.
25. Корячко, В. П. Теоретические основы САПР / В. П. Корячко, В. М. Курейчик. – М. : Энергоатомиздат, 1987. – 400 с.
26. Джонс, Дж. К. Методы проектирования / Дж. К. Джонс. – М. : Мир, 1986.
27. Автоматизированное проектирование систем управления / под ред. М. Джемшиди. – М. : Машиностроение, 1989.
- Журналы:
28. CAD/CAM/CAE Observer.
29. Информационные технологии.
30. Информационные технологии в проектировании.
31. Информационно-измерительные и управляющие системы.
32. Информационные технологии.
33. Математическое моделирование.
34. Системы управления и информационные технологии.