

Лекция 1

ВВЕДЕНИЕ В АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ.

Человечество вступило в XXI в., в котором придется решать ряд сложных проблем, связанных с экологией, поиском новых источников энергии, материалов, технологий, соответствующих постиндустриальному обществу. Определяющая роль в решении названных проблем отводится информационным технологиям.

Среди информационных технологий автоматизация проектирования занимает особое место. В-первых, автоматизация проектирования - синтетическая дисциплина, ее составными частями являются многие другие современные информационные технологии. Так, техническое обеспечение систем автоматизированного проектирования (САПР) основано на использовании вычислительных сетей и телекоммуникационных технологий, в САПР используются персональные компьютеры и рабочие станции, есть примеры применения мейнфреймов. Математическое обеспечение САПР отличается богатством и разнообразием используемых методов вычислительной математики, статистики, математического программирования, дискретной математики, искусственного интеллекта. Программные комплексы САПР относятся к числу наиболее сложных современных программных систем, основанных на операционных системах Unix, Windows-NT, языках программирования C, C++, Java и других, современных CASE-технологиях, реляционных и объектно-ориентированных системах управления базами данных (СУБД), стандартах открытых систем и обмена данными в компьютерных средах.

Во-вторых, знание основ автоматизации проектирования и умение работать со средствами САПР требуются практически любому инженеру-разработчику. Компьютерами насыщены проектные подразделения, конструкторские бюро и офисы. Работа конструктора за обычным кульманом, расчеты с помощью логарифмической линейки или оформление отчета на пишущей машинке стали анахронизмом. Предприятия, ведущие разработки без САПР или лишь с малой степенью их использования, оказываются неконкурентоспособными вследствие как больших материальных и временных затрат на проектирование, так и невысокого качества проектов.

Появление первых программ для автоматизации проектирования за рубежом и в нашей стране относится к началу 1960-х годов. Тогда были созданы программы для решения задач строительной механики, анализа электронных схем, проектирования печатных плат. Дальнейшее развитие САПР шло по пути создания аппаратных и программных средств машинной графики, повышения вычислительной эффективности программ моделирования и анализа, расширения областей применения САПР, упрощения пользовательского интерфейса, внедрения в САПР элементов искусственного интеллекта.

К настоящему времени создано большое число программно-методических комплексов для САПР с различными степенью специализации и прикладной ориентацией. В результате автоматизация проектирования стала необходимой составной частью подготовки инженеров разных специальностей; инженер, не владеющий знаниями и не умеющий работать в САПР, не может считаться полноценным специалистом.

Системный подход к проектированию

Понятие инженерного проектирования

Проектирование технического объекта - создание, преобразование и представление в принятой форме образа этого еще 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 объекты строительства, промышленные изделия или процессы. Проектирование включает в себя разработку технического предложения и (или) технического задания (ТЗ), отражающих эти потребности, и реализацию ТЗ в виде проектной документации.

Обычно ТЗ представляют в виде некоторых документов, и оно является исходным (первичным) 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, окончательное 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.

Преобразование исходного описания в окончательное порождает ряд промежуточных описаний, подводящих итоги решения некоторых задач и используемых при обсуждении и принятии 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 автоматического (без участия человека на промежуточных этапах). Система, реализующая автоматизированное проектирование, представляет собой **систему автоматизированного проектирования САПР** (в англоязычном написании **CAD System -Computer Aided Design System**).

Автоматическое проектирование возможно лишь в отдельных частных случаях для сравнительно несложных объектов. Превалирующим в настоящее время является автоматизированное проектирование.

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

Принципы системного подхода

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 применяют 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, определение атрибутов, анализ влияния внешней среды.

Системный подход рассматривают как направление научного познания и социальной политики. Он является базой для обобщающей дисциплины «**Теория систем**» (другое используемое название - «**Системный анализ**»). **Теория систем** - дисциплина, в которой конкретизируются положения системного подхода; она посвящена исследованию и проектированию сложных экономических, социальных, технических систем, чаще всего слабоструктурированных. Характерными примерами таких систем являются производственные системы. При проектировании систем цели достигаются в многошаговых процессах принятия решений. Методы принятия решений часто выделяют в самостоятельную дисциплину, называемую «**Теория принятия решений**».

В технике дисциплину, в которой исследуются сложные технические системы, их проектирование и которая аналогична теории систем, чаще называют **системотехникой**. Предметом системотехники являются, во-первых, организация процесса создания, использования и развития технических систем, во-вторых, методы и принципы их проектирования и исследования. 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.

Блочнo-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. 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 также следующие особенности.

1. Структуризация процесса проектирования, выражаемая декомпозицией проектных задач и документации, выделением стадий, этапов, проектных процедур. Эта структуризация является сущностью блочно-иерархического подхода к проектированию.

2. Итерационный характер проектирования.

3. Типизация и унификация проектных решений и средств проектирования.

Основные понятия системотехники

В теории систем и системотехнике введен ряд терминов, среди них к базовым нужно отнести следующие понятия.

Система — множество элементов, находящихся в отношениях и связях между собой.

Элемент - такая часть системы, представление о которой нецелесообразно подвергать при проектировании дальнейшему членению.

Сложная система - система, характеризуемая большим числом элементов и, что наиболее важно, большим числом взаимосвязей элементов. Сложность системы определяется также видом взаимосвязей элементов, свойствами целенаправленности, целостности, иерархичности, многоаспектности. Очевидно, что современные автоматизированные информационные системы и, в частности, САПР являются сложными в силу наличия у них перечисленных свойств и признаков.

Подсистема—часть системы (подмножество элементов и их взаимосвязей), которая имеет свойства системы.

Надсистема - система, по отношению к которой рассматриваемая система является подсистемой.

Структура - отображение совокупности элементов системы и их взаимосвязей; понятие структуры отличается от понятия самой системы также тем, что при описании структуры принимают во внимание лишь типы элементов и связей без конкретизации значений их параметров.

Параметр — величина, выражающая свойство или системы, или ее части, или влияющей на систему среды. Обычно в моделях систем в качестве параметров рассматривают величины, не изменяющиеся в процессе исследования системы. Параметры подразделяют на *внешние, внутренние и выходные*, выражающие свойства элементов системы, самой системы, внешней среды соответственно. Векторы внутренних, выходных и внешних параметров далее обозначены $X = (x_1, x_2, \dots, x_n)$, $Y = (y_1, y_2, \dots, y_m)$, $Q = (q_1, q_2, \dots, q_k)$ соответственно.

Фазовая переменная — величина, характеризующая энергетическое или информационное наполнение элемента или подсистемы.

Состояние — совокупность значений фазовых переменных, зафиксированных в одной временной точке процесса функционирования.

Поведение (динамика) системы — изменение состояния системы в процессе функционирования.

Система без последствия — ее поведение при $t > t_0$ определяется заданием состояния в момент t_0 и вектором внешних воздействий $Q(t)$. В системах с последствием, кроме того, нужно знать предысторию поведения, т.е. состояния системы в моменты, предшествующие t_0 .

Вектор переменных V , характеризующих состояние (вектор переменных состояния), - неизбыточное множество фазовых переменных, задание значений которых в некоторый момент времени полностью определяет поведение системы в дальнейшем (в автономных системах без последствия).

Пространство состояний — множество возможных значений вектора переменных состояния.

Фазовая траектория — представление процесса (зависимости $V(t)$) в виде последовательности точек в пространстве состояний.

К характеристикам сложных систем, как сказано выше, часто относят следующие понятия.

Целенаправленность — свойство искусственной системы, выражающее назначение системы. Это свойство необходимо для оценки эффективности вариантов системы.

Целостность — свойство системы, характеризующее взаимосвязанность элементов и наличие зависимости выходных параметров от параметров элементов, при этом большинство выходных параметров не является простым повторением или суммой параметров элементов.

Иерархичность — свойство сложной системы, выражающее возможность и целесообразность ее иерархического описания, т. е. представления в виде нескольких уровней, между компонентами которых имеются отношения целое — часть.

Составными частями системотехники являются следующие основные разделы:

- иерархическая структура систем, организация их проектирования;
- анализ и моделирование систем;
- синтез и оптимизация систем.

Моделирование имеет две четко различимые задачи:

1 — создание моделей сложных систем (в англоязычном написании - **modeling**);

2 - анализ свойств систем на основе исследования их моделей (**simulation**).

Синтез также подразделяют на две задачи:

1 — синтез структуры проектируемых систем (структурный синтез);

2 — выбор численных значений параметров элементов систем (параметрический синтез). Эти задачи относятся к области принятия проектных решений.

Моделирование и оптимизацию желательно выполнять с учетом статистической природы систем. Детерминированность - лишь частный случай. При проектировании характерны нехватка достоверных исходных данных, неопределенность условий принятия решений. Учет статистического характера данных при моделировании в значительной мере основан на методе статистических испытаний (методе Монте-Карло), а принятие решений - на использовании нечетких множеств, экспертных систем, эволюционных вычислений.

Примеры.

1. Компьютер является сложной системой в силу наличия у него большого числа элементов, разнообразных связей между элементами и подсистемами, свойств целенаправленности, целостности, иерархичности. К подсистемам компьютера относятся процессор (процессоры), оперативная память, кэш-память, шины, устройства ввода-вывода. В качестве надсистемы могут выступать вычислительная сеть, автоматизированная и (или) организационная система, к которым принадлежит компьютер. Внутренние параметры - времена выполнения арифметических операций, чтения (записи) в накопителях, пропускная способность шин и др. Выходные параметры - производительность компьютера, емкость оперативной и внешней памяти, себестоимость, время наработки на отказ и др. Внешние параметры — напряжение питания сети и его стабильность, температура окружающей среды и др.

2. Для двигателя внутреннего сгорания подсистемами являются коленчатый вал, механизм газораспределения, поршневая группа, системы смазывания и охлаждения. Внутренние параметры - число цилиндров, объем камеры сгорания и др. Выходные параметры - мощность двигателя, КПД, расход топлива и др. Внешние параметры - характеристики топлива, температура воздуха, нагрузка на выходном валу.

3. Подсистемы электронного усилителя - усилительные каскады; внутренние параметры - сопротивления резисторов, емкости конденсаторов, параметры транзисторов; выходные параметры - коэффициент усиления на средних частотах, полоса пропускания, входное сопротивление; внешние параметры - температура окружающей среды, напряжения источников питания, сопротивление нагрузки.

Стадии 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, 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 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**. Примерами проектных процедур могут служить подготовка **детализировочных** чертежей, анализ кинематики, моделирование переходного процесса, оптимизация параметров и другие проектные задачи. В свою очередь, проектные процедуры можно расчленить 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. Проектирование сводится к выполнению некоторых последовательностей проектных процедур - **маршрутов проектирования**.

Иногда разработку ТЗ на проектирование называют **внешним** проектированием, а реализацию ТЗ - **внутренним** проектированием.

Содержание технических заданий на проектирование

В ТЗ на проектирование объекта указывают, по крайней мере, следующие данные.

1. Назначение объекта.

2. Условия эксплуатации. Наряду с качественными характеристиками (представленными в вербальной форме) имеются числовые параметры, называемые **внешними** параметрами, для которых указаны области допустимых значений. Примеры внешних параметров: температура окружающей среды, внешние силы, электрические напряжения, нагрузки и т. п.

3. Требования к **выходным** параметрам, т. е. к величинам, характеризующим свойства объекта, интересующие потребителя. Эти требования выражены в виде **условий работоспособности**

$$y_i R T,$$

где y_i - i -й выходной параметр; $R \in \{ =, <, >, \leq, \geq \}$ - вид отношения; T - норма i -го выходного параметра. В случае $R =$ (равно) нужно задать требуемую точность выполнения равенства.

Примеры условий работоспособности:

- расход топлива на 100 км пробега автомобиля < 8 л;
- коэффициент усиления усилителя на средних частотах > 300 ;
- быстродействие процессора > 40 Мфлопс.

Лекция 2

Структура процесса проектирования Иерархическая структура проектных спецификаций и иерархические уровни проектирования

При использовании блочно-иерархического подхода к проектированию представления о проектируемой системе расчленяют на *иерархические уровни*. На верхнем уровне используют наименее детализированное представление, отражающее только самые общие черты и особенности проектируемой системы. На следующих уровнях степень подробности описания возрастает, при этом рассматривают уже отдельные блоки системы, но с учетом воздействий на каждый из них его соседей. Такой подход позволяет на каждом иерархическом уровне формулировать задачи приемлемой сложности, поддающиеся решению с помощью имеющихся средств проектирования. Разбиение на уровни должно быть таким, чтобы документация на блок любого уровня была обозрима и воспринимаема одним человеком.

Другими словами, блочно-иерархический подход есть *декомпозиционный* подход (его можно назвать также *диакоптическим*), который основан на разбиении сложной задачи большой размерности на последовательно и (или) параллельно решаемые группы задач малой размерности, что существенно сокращает требования к используемым вычислительным ресурсам или время решения задач.

Можно говорить об иерархических уровнях не только спецификаций, но и проектирования, понимая под каждым из них совокупность спецификаций некоторого иерархического уровня совместно с постановками задач, методами получения описаний и решения возникающих проектных задач.

Список иерархических уровней в каждом приложении может быть специфичным, но для большинства приложений характерно следующее наиболее крупное выделение уровней:

- *системный* уровень, на котором решают наиболее общие задачи проектирования систем, машин и процессов; результаты проектирования представляют в виде структурных схем, генеральных планов, схем размещения оборудования, диаграмм потоков данных и т. п.;
- *макроуровень*, на котором проектируют отдельные устройства, узлы машин и приборов; результаты представляют в виде функциональных, принципиальных и кинематических схем, сборочных чертежей и т. п.;
- *микроуровень*, на котором проектируют отдельные детали и элементы машин и приборов.

В каждом приложении число выделяемых уровней и их наименования могут быть различными. Так, в радиоэлектронике *микроуровень* часто называют *компонентным*, *макроуровень* - *схемотехническим*. Между схемотехническим и системным уровнями вводят уровень, называемый функционально-логическим. В вычислительной технике системный уровень подразделяют на уровни проектирования ЭВМ (вычислительных систем) и вычислительных сетей. В машиностроении имеются уровни деталей, узлов, машин, комплексов.

В зависимости от последовательности решения задач иерархических уровней различают нисходящее, восходящее и смешанное проектирование (стили проектирования). Последовательность решения задач от нижних уровней к верхним характеризует *восходящее проектирование*, обратная последовательность приводит к *нисходящему проектированию*, в смешанном стиле имеются элементы как восходящего, так и нисходящего проектирования. В большинстве случаев для сложных систем предпочитают нисходящее проектирование. Отметим, однако, что при наличии заранее спроектированных составных блоков (устройств) можно говорить о смешанном проектировании.

Неопределенность и нечеткость исходных данных при нисходящем проектировании (так как еще не спроектированы компоненты) или исходных требований при восходящем проектировании (поскольку ТЗ имеется на всю систему, а не на ее части) обуславливают необходимость прогнозирования недостающих данных с последующим их уточнением, т. е. последовательного приближения к окончательному решению (итерационность проектирования).

Наряду с декомпозицией описаний на иерархические уровни применяют разделение представлений о проектируемых объектах на аспекты.

Аспект описания (страта) — описание системы или ее части с некоторой оговоренной точки зрения, определяемой функциональными, физическими или иного типа отношениями между свойствами и элементами.

Различают *функциональный*, *информационный*, *структурный* и *поведенческий* (процессный) аспекты.

- **Функциональное** описание относят к функциям системы и чаще всего представляют его функциональными схемами.
- **Информационное** описание включает в себя основные понятия предметной области (сущности), словесное пояснение или числовые значения характеристик (атрибутов) используемых объектов, а также описание связей между этими понятиями и характеристиками. Информационные модели можно представлять графически (графы, диаграммы сущность - отношение), в виде таблиц или списков.
- **Структурное** описание относится к морфологии системы, характеризует составные части системы и их соединения и может быть представлено структурными схемами, а также различного рода конструкторской документацией.
- **Поведенческое** описание характеризует процессы функционирования (алгоритмы) системы и (или) технологические процессы создания системы.

Иногда аспекты описаний связывают с подсистемами, функционирование которых основано на различных физических процессах.

Отметим, что в общем случае выделение страт может быть неоднозначным. Так, помимо указанного подхода очевидна целесообразность выделения таких аспектов, как функциональное (разработка принципов действия, структурных, функциональных, принципиальных схем), конструкторское (определение форм и пространственного расположения компонентов изделий), алгоритмическое (разработка алгоритмов и программного обеспечения) и технологическое (разработка технологических процессов) проектирование систем. Примерами страт в случае САПР могут служить также рассматриваемые далее виды обеспечения автоматизированного проектирования.

Классификация моделей и параметров, используемых при автоматизированном проектировании

В автоматизированных проектных процедурах вместо еще не существующего проектируемого объекта оперируют некоторым *квазиобъектом-моделью*, которая отражает некоторые интересующие исследователя свойства объекта. Модель может быть физическим объектом (макет, стенд) или спецификацией. Среди моделей-спецификаций различают упомянутые выше функциональные, поведенческие, информационные, структурные модели (описания). Эти модели называют *математическими*, если они формализованы средствами аппарата и языка математики.

В свою очередь, математические модели могут быть геометрическими, топологическими, динамическими, логическими и т. п., если они отражают соответствующие свойства объектов. Наряду с математическими моделями при проектировании используют рассматриваемые ниже функциональные IDEF0-модели, информационные модели в виде диаграмм сущность - отношение, геометрические модели-чертежи. В дальнейшем, если нет специальной оговорки, под словом «модель» будем подразумевать математическую модель (МО).

Математическая функциональная модель в общем случае представляет собой алгоритм вычисления вектора выходных параметров Y при заданных векторах параметров элементов X и внешних параметров Q .

Математические модели могут быть символическими и численными. При использовании *символических* моделей оперируют не значениями величин, а их символическими обозначениями (идентификаторами). *Численные* модели могут быть *аналитическими*, т. е. их можно представить в виде явно выраженных зависимостей выходных параметров Y от *параметров внутренних* X и внешних Q , или *алгоритмическими*, в которых связь Y , X и Q задана неявно в виде алгоритма моделирования. Важнейший частный случай алгоритмических моделей - *имитационные*, они отображают процессы в системе при наличии внешних воздействий на систему. Другими словами, имитационная модель - это алгоритмическая поведенческая модель.

Классификацию математических моделей выполняют также по ряду других признаков. Так, в зависимости от принадлежности к тому или иному иерархическому уровню выделяют модели уровней системного, функционально-логического, *макроуровня* (сосредоточенного) и *микроуровня* (распределенного).

По характеру используемого для описания математического аппарата различают модели лингвистические, теоретико-множественные, абстрактно-алгебраические, нечеткие, автоматные и т. п.

Например, на системном уровне преимущественно применяют модели систем массового обслуживания и сети Петри, на функционально-логическом уровне - автоматные модели на основе аппарата передаточных функций или конечных автоматов, на *макроуровне* - системы алгебраических и

дифференциальных уравнений, на *микроуровне* - дифференциальные уравнения в частных производных. Особое место занимают геометрические модели, используемые в системах конструирования.

Кроме того, введены понятия полных моделей и макромоделей, моделей статических и динамических, детерминированных и стохастических, аналоговых и дискретных, символических и численных.

Полная модель объекта в отличие от *макромодели* описывает не только процессы на внешних выводах моделируемого объекта, но и внутренние для объекта процессы.

Статические модели описывают статические состояния, в них не присутствует время в качестве независимой переменной. *Динамические* модели отражают поведение системы, т. е. в них обязательно используется время.

Стохастические и детерминированные модели различают в зависимости от учета или неучета случайных факторов.

В *аналоговых* моделях фазовые переменные - непрерывные величины, в *дискретных* - дискретные, в частном случае дискретные модели являются *логическими (булевыми)*, в них состояние системы и ее элементов описывается булевыми величинами. В ряде случаев полезно применение *смешанных* моделей, в которых одна часть подсистем характеризуется аналоговыми моделями, другая - логическими.

Информационные модели относятся к информационной страте автоматизированных систем, их используют прежде всего при инфологическом проектировании баз данных для описания связей между единицами информации.

Наибольшие трудности возникают при создании моделей слабоструктурированных систем, что характерно прежде всего для системного уровня проектирования. Здесь значительное внимание уделяется экспертным методам. В теории систем сформулированы общие рекомендации по подбору экспертов при разработке модели, организации экспертизы, по обработке полученных результатов. Достаточно общий подход к построению моделей сложных слабоструктурированных систем выражен в методиках **IDEF**.

Обычно в имитационных моделях фигурируют фазовые переменные. Так, на макроуровне имитационные модели представляют собой системы алгебро-дифференциальных уравнений

$$\Phi(dV/dt, V, t) = 0 \text{ при } t = 0, V = V_0, \quad (1.1)$$

где V —вектор фазовых переменных; t - время; V_0 —вектор начальных условий. К примерам фазовых переменных можно отнести токи и напряжения в электрических системах, силы и скорости - в механических, давления и расходы—в гидравлических.

Выходные параметры систем могут быть двух типов. Во-первых, это параметры-функционалы, т. е. функционалы зависимостей $V(Y)$ в случае использования (1.1). Примеры таких параметров: амплитуды сигналов, временные задержки, мощности рассеивания и т. п. Во-вторых, это параметры, характеризующие способность проектируемого объекта работать при определенных внешних условиях. Эти выходные параметры являются граничными значениями диапазонов внешних переменных, в которых сохраняется работоспособность объекта.

Типовые проектные процедуры

Создать проект объекта (изделия или процесса) означает выбрать структуру объекта, определить значения всех его параметров и представить результаты в установленной форме. Результаты (проектная документация) могут быть выражены в виде чертежей, схем, пояснительных записок, программ для программно-управляемого технологического оборудования и других документов на бумаге или на машинных носителях информации.

Разработка (или выбор) структуры объекта есть проектная процедура, называемая *структурным синтезом*, а расчет (или выбор) значений параметров элементов X - процедура *параметрического синтеза*.

Задача структурного синтеза формулируется в системотехнике как *задача принятия решений (ЗПР)*. Ее суть заключается в определении цели, множества возможных решений и ограничивающих условий.

Классификацию ЗПР осуществляют по ряду признаков. По числу критериев различают задачи одно- и многокритериальные. По степени неопределенности различают ЗПР детерминированные, ЗПР в условиях риска (при наличии в формулировке задачи случайных параметров), ЗПР в условиях неопределенности, т. е. при неполноте или недостоверности исходной информации.

Реальные задачи проектирования, как правило, являются многокритериальными. Одна из основных проблем постановки многокритериальных задач - установление правил предпочтения вариантов. Способы сведения многокритериальных задач к однокритериальным и последующие пути решения изучаются в дисциплинах, посвященных методам оптимизации и математическому программированию.

Наличие случайных факторов усложняет решение ЗПР. Основные подходы к решению ЗПР в условиях риска заключаются или в решении «для наихудшего случая», или в учете в целевой функции математического ожидания и дисперсии выходных параметров. В первом случае задачу решают как детерминированную при завышенных требованиях к качеству решения, что является главным недостатком подхода. Во втором случае достоверность результатов решения намного выше, но возникают трудности с оценкой целевой функции. Применение метода Монте-Карло в случае алгоритмических моделей становится единственной альтернативой, и, следовательно, для решения требуются значительные вычислительные ресурсы.

Существуют две группы ЗПР в условиях неопределенности. Одна из них решается при наличии противодействия разумного противника. Такие задачи изучаются в теории игр, для задач проектирования в технике они не характерны. Во второй группе противодействие достижению цели оказывают силы природы. Для их решения полезно использовать теорию и методы нечетких множеств.

Например, при синтезе структуры автоматизированной системы постановка задачи должна включать в качестве исходных данных следующие сведения:

- множество выполняемых системой функций (другими словами, множество работ, каждая из которых может состоять из одной или более операций); возможно, что в этом множестве имеется частичная упорядоченность работ, которая может быть представлена в виде ориентированного графа, где вершины соответствуют работам, а дуги - отношениям порядка;
- типы допустимых для использования серверов (машин), выполняющих функции системы;
- множество внешних источников и потребителей информации;
- во многих случаях задается также некоторая исходная структура системы в виде взаимосвязанной совокупности серверов определенных типов; эта структура может рассматриваться как обобщенная избыточная или как вариант первого приближения;
- различного рода ограничения, в частности ограничения на затраты материальных ресурсов и (или) на времена выполнения функций системы.

Задача заключается в синтезе (или коррекции) структуры, определении типов серверов (программно-аппаратных средств), распределении функций по серверам таким образом, чтобы достигался экстремум целевой функции при выполнении заданных ограничений.

Конструирование, разработка технологических процессов, оформление проектной документации — частные случаи структурного синтеза.

Задачу параметрического синтеза называют *параметрической оптимизацией* (или оптимизацией), если ее решают как задачу математического программирования, т. е.

$$\text{extr } F(\mathbf{X}), \mathbf{X} \in \mathbf{D}_X,$$

где $F(\mathbf{X})$, - целевая функция; \mathbf{X} - вектор управляемых (называемых также проектными или варьируемыми) параметров; $\mathbf{D}_X = \{\mathbf{X} \mid \varphi(\mathbf{X}) < 0, \psi(\mathbf{X}) = 0\}$ - допустимая область; $\varphi(\mathbf{X})$ и $\psi(\mathbf{X})$ - функции-ограничения.

Пример. Электронный усилитель: управляемые параметры \mathbf{X} = (параметры резисторов, конденсаторов, транзисторов); выходные параметры $\mathbf{Y} = (f_B, f_H)$, -верхняя и нижняя граничные частоты полосы пропускания; K -коэффициент усиления на средних частотах; $R_{вх}$ -входное сопротивление). В качестве целевой функции $F(\mathbf{X})$ можно выбрать параметр f_B , а условия работоспособности остальных выходных параметров отнести к функциям-ограничениям.

Следующая после синтеза группа проектных процедур - процедуры анализа. Цель *анализа* - получение информации о характере функционирования и значениях выходных параметров \mathbf{Y} при заданных структуре объекта, сведениях о внешних параметрах \mathbf{Q} и параметрах элементов \mathbf{X} . Если заданы фиксированные значения параметров \mathbf{X} и \mathbf{Q} , то имеет место процедура *одно-вариантного анализа*, которая сводится к решению уравнений математической модели, например такой, как модель (1.1), и вычислению вектора выходных параметров \mathbf{Y} . Если заданы статистические сведения о параметрах \mathbf{X} и нужно получить оценки числовых характеристик распределений выходных параметров (например, оценки математических ожиданий и дисперсий), то это процедура *статистического анализа*. Если требуется рассчитать матрицы абсолютной \mathbf{A} и (или) относительной \mathbf{B} чувствительности, то имеет место задача *анализа чувствительности*.

Элемент A_{ji} матрицы A называют **абсолютным коэффициентом чувствительности**, он представляет собой частную производную j -го выходного параметра y_j по i -му параметру x_i . Другими словами, A_{ji} является элементом вектора градиента y -го выходного параметра. На практике удобнее использовать безразмерные **относительные коэффициенты чувствительности** B_{ji} , характеризующие степень влияния изменений параметров элементов на изменения выходных параметров:

$$B_{ji} = A_{ji} x_{i \text{ ном}} / y_{j \text{ ном}}$$

где $x_{i \text{ ном}}$ и $y_{j \text{ ном}}$ - номинальные значения параметров x , и y соответственно.

В процедурах **многовариантного анализа** определяется влияние внешних параметров, разброса и нестабильности параметров элементов на выходные параметры. Процедуры статистического анализа и анализа чувствительности - характерные примеры процедур многовариантного анализа.

Системы автоматизированного проектирования и их место среди других автоматизированных систем

Структура САПР

Как и любая сложная система, САПР состоит из подсистем. Различают подсистемы проектирующие и обслуживающие.

Проектирующие подсистемы непосредственно выполняют проектные процедуры. Примерами проектирующих подсистем могут служить подсистемы геометрического трехмерного моделирования механических объектов, изготовления конструкторской документации, схемотехнического анализа, трассировки соединений в печатных платах.

Обслуживающие подсистемы обеспечивают функционирование проектирующих подсистем, их совокупность часто называют системной средой (или оболочкой) САПР. Типичными обслуживающими подсистемами являются подсистемы управления проектными данными, подсистемы разработки и сопровождения программного обеспечения **CASE (Computer Aided Software Engineering)**, обучающие подсистемы для освоения пользователями технологий, реализованных в САПР.

Структурирование САПР по различным аспектам обуславливает появление **видов обеспечения САПР**. Принято выделять семь видов обеспечения САПР:

- **техническое (ТО)**, включающее различные аппаратные средства (ЭВМ, периферийные устройства, сетевое коммутационное оборудование, линии связи, измерительные средства);
- **математическое (МО)**, объединяющее математические методы, модели и алгоритмы для выполнения проектирования;
- **программное**, представляемое компьютерными программами САПР;
- **информационное**, состоящее из базы данных, СУБД, а также включающее другие данные, используемые при проектировании; отметим, что вся совокупность используемых при проектировании данных называется информационным фондом САПР, а база данных вместе с СУБД носит название банка данных;
- **лингвистическое**, выражаемое языками общения между проектировщиками и ЭВМ, языками программирования и языками обмена данными между техническими средствами САПР;
- **методическое**, включающее различные методики проектирования, иногда к нему относят также математическое обеспечение;
- **организационное**, представляемое штатными расписаниями, должностными инструкциями и другими документами, регламентирующими работу проектного предприятия.

Разновидности САПР

Классификацию САПР осуществляют по ряду признаков, например по приложению, целевому назначению, масштабам (комплексности решаемых задач), характеру базовой подсистемы - ядра 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 системами **MCAD (Mechanical CAD)**.
2. САПР для радиоэлектроники: системы **ECAD (Electronic CAD)** или **EDA (Electronic Design Automation)**.
3. САПР в области архитектуры и строительства.

Кроме того, известно большое число специализированных САПР, или выделяемых в указанных группах, или представляющих самостоятельную ветвь 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, обеспечивающие разные аспекты (страты) проектирования. Так, в составе МСАД появляются рассмотренные выше САЕ/CAD/CAM-системы.

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*) обеспечений.

По характеру базовой подсистемы различают следующие разновидности САПР:

1. САПР на базе подсистемы 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. К этой группе систем относится большинство САПР в области машиностроения, построенных на базе графических ядер.

В настоящее время широко используют унифицированные графические ядра, применяемые более чем в одной САПР (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 они имеются также при проектировании объектов, подобных щитам управления в системах автоматизации.

3. САПР на базе конкретного прикладного пакета. Фактически это автономно используемые ПМК, например, имитационного моделирования производственных процессов, расчета прочности по МКЭ, 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

Понятие о CALS-технологиях

Технологии комплексной компьютеризации сфер промышленного производства, цель которых— унификация и стандартизация спецификаций промышленной продукции на всех этапах ее жизненного цикла, называют CALS-технологиями. 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. Терминология в области CALS еще окончательно не установилась. Так, первоначально аббревиатура CALS расшифровывалась как **Computer Aided Logistics Systems**, т. е. автоматизированная логистическая поддержка. 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, применяют и более соответствующую предмету расшифровку аббревиатуры CALS — **Continuous Acquisition and Lifecycle Support**. В русском языке понятию CALS соответствует ИПИ (информационная поддержка изделий) или КСПИ (компьютерное сопровождение и поддержка изделий).

Применение CALS-технологий позволяет существенно сократить объемы проектных работ, так как описания многих составных частей оборудования, машин и систем, проектировавшихся ранее, хранятся в унифицированных форматах данных сетевых серверов, доступных любому пользователю технологий 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. Предполагается, что успех на рынке сложной технической продукции будет немалым вне технологий CALS.

Развитие **CALS**-технологий 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**-технологий 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 современных **CALS**-технологий. 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.

Таким образом, информационная интеграция является неотъемлемым свойством **CALS**-систем. Поэтому в основу **CALS**-технологий 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.

1.4. Особенности проектирования автоматизированных систем Этапы проектирования

К проектированию **АС** непосредственное отношение имеют два направления деятельности:

- 1) собственно проектирование **АС** конкретных предприятий (отраслей) на базе готовых программных и аппаратных компонентов с помощью специальных инструментальных средств разработки;
- 2) проектирование упомянутых компонентов **АС** и инструментальных средств, ориентированных на многократное применение при разработке многих конкретных автоматизированных систем.

Сущность первого направления можно охарактеризовать словами «*системная интеграция*» (другое близкое понятие имеет название *консалтинг*). Разработчик **АС** должен быть специалистом в области системотехники, хорошо знать соответствующие международные стандарты, состояние и тенденции развития информационных технологий и программных продуктов, владеть инструментальными средствами разработки приложений (**CASE**-средствами) и быть готовым к восприятию и анализу автоматизируемых процессов в сотрудничестве со специалистами-прикладниками.

Существует ряд фирм, специализирующихся на разработке проектов **АС** (например, **Price Waterhouse, Jet Info, Consistent Software, Interface** и др.)

Второе направление в большей мере относится к области разработки **МО** и **ПО** для реализации функций **АС** — моделей, методов, алгоритмов, программ на базе знания системотехники, методов анализа и синтеза проектных решений, технологий программирования, операционных систем и т. п. Существует ряд общеизвестных технологий (методик) проектирования **ПО АС**, среди которых прежде всего следует назвать компонентно-ориентированную разработку — технологию индустриальной разработки программных систем.

Для каждого класса **АС** (**САПР**, **ERP**, геоинформационные системы и т. д.) можно указать фирмы, специализирующиеся на разработке программных (а иногда и программно-аппаратных) систем. Многие из них на основе одной из базовых технологий реализуют свой подход к созданию **АС** и придерживаются стратегии либо тотального поставщика, либо открытости и расширения системы приложениями и дополнениями третьих фирм.

В России действует государственный стандарт на стадии создания автоматизированных систем ГОСТ 34.601-90. Существует и международный стандарт на стадии жизненного цикла программной

продукции (ISO 12207:1995). Как собственно АС, так и компоненты АС являются сложными системами, и при их проектировании нужно использовать один из стилей проектирования:

- **нисходящее (Top-of-Design)**; четкая реализация нисходящего проектирования приводит к *спиральной модели* разработки ПО, на каждом витке спирали блоки предыдущего уровня детализируются, используются обратные связи (альтернативой является так называемая *каскадная модель*, относящаяся к поочередной реализации частей системы);
- **восходящее (Bottom-of-Design)**;
- **эволюционное (Middle-of-Design)**.

Чаще всего применяют нисходящий стиль блочно-иерархического проектирования.

Рассмотрим этапы нисходящего проектирования АС.

Верхний уровень проектирования АС часто называют **концептуальным** проектированием. Концептуальное проектирование выполняют в процессе предпроектных исследований, формулировки ТЗ, разработки эскизного проекта и прототипирования (согласно ГОСТ 34.601-90, эти стадии называют формированием требований к АС, разработкой концепции АС и эскизным проектом).

Предпроектные исследования проводят путем анализа (обследования) деятельности предприятия (компании, учреждения, офиса), на котором создается или модернизируется АС. При этом нужно получить ответы на вопросы: что не устраивает в существующей технологии? Что можно улучшить? Кому это нужно и, следовательно, каков будет эффект? Перед обследованием формируются и в процессе его проведения уточняются цели обследования - определение возможностей и ресурсов для повышения эффективности функционирования предприятия на основе автоматизации процессов управления, проектирования, документооборота и т. п. Содержание обследования - выявление структуры предприятия, выполняемых функций, информационных потоков, имеющихся опыта и средств автоматизации. Обследование проводят системные аналитики (системные интеграторы) совместно с представителями организации-заказчика.

На основе анализа результатов обследования строят модель, отражающую деятельность предприятия на данный момент (до реорганизации). Такую модель называют «*As Is*» (как есть). Далее разрабатывают исходную концепцию АС. Эта концепция включает в себя предложения по изменению структуры предприятия, взаимодействию подразделений, информационным потокам, что выражается в модели «*To Be*» (как должно быть).

Результаты анализа конкретизируются в ТЗ на создание АС. В ТЗ указывают потоки входной информации, типы выходных документов и предоставляемых услуг, уровень защиты информации, требования к производительности (пропускной способности) и т. п. ТЗ направляют заказчику для обсуждения и окончательного согласования.

Эскизный проект (техническое предложение) представляют в виде проектной документации, описывающей архитектуру системы, структуру ее подсистем, состав модулей. Здесь же содержатся предложения по выбору базовых программно-аппаратных средств, которые должны учитывать прогноз развития предприятия.

После принятия эскизного проекта разрабатывают **прототип** АС, представляющий собой набор программ, эмулирующих работу готовой системы. Благодаря прототипированию можно не только разработчикам, но и будущим пользователям АС увидеть контуры и особенности системы и, следовательно, заблаговременно внести коррективы в проект.

При концептуальном проектировании применяют ряд спецификаций, среди которых центральное место занимают модели преобразования, хранения и передачи информации. Модели, полученные в процессе обследования предприятия, являются моделями его функционирования. В процессе разработки АС модели, как правило, претерпевают существенные изменения (переход от «*As Is*» к «*To Be*») и в окончательном виде модель «*To Be*» рассматривают в качестве модели проектируемой АС.

Различают **функциональные, информационные, поведенческие и структурные** модели.

- **Функциональная** модель системы описывает совокупность выполняемых системой функций.
- **Информационная** модель отражает структуры данных - их состав и взаимосвязи.
- **Поведенческая** модель описывает информационные процессы (динамику функционирования), в ней фигурируют такие категории, как состояние системы, событие, переход из одного состояния в другое, условия перехода, последовательность событий, осуществляется привязка ко времени.
- **Структурная** модель характеризует морфологию системы (ее построение) - состав подсистем, их взаимосвязи.

Содержанием последующих этапов нисходящего проектирования (согласно ГОСТ 34.601—90, это стадии разработки технического проекта, рабочей документации, ввода в действие) являются уточнение перечней приобретаемого оборудования и готовых программных продуктов, построение

системной среды, детальное мифологическое проектирование баз данных и их первоначальное наполнение, разработка собственного оригинального ПО, которая, в свою очередь, делится на ряд этапов нисходящего проектирования. Эти работы составляют содержание *рабочего проектирования*. После этого следуют закупка и инсталляция программно-аппаратных средств, внедрение и опытная эксплуатация системы.

Открытые системы

Одной из главных тенденций современной индустрии информатики является создание *открытых систем*. Свойство открытости означает, во-первых, переносимость (мобильность) ПО на различные аппаратные платформы, во-вторых, приспособленность системы к ее модификациям (модифицируемость или собственно открытость) и комплексированию с другими системами в целях расширения ее функциональных возможностей и (или) придания системе новых качеств (интегрируемость).

Переход к открытым информационным системам позволяет существенно ускорить научно-технический прогресс в результате замены длительной и дорогостоящей разработки новых систем по полному циклу их компоновкой из ранее спроектированных подсистем или быстрой модернизацией уже существующих систем (реинжиниринг).

Открытость подразумевает выделение в системе интерфейсной части (входов и выходов), обеспечивающей сопряжение с другими системами или подсистемами, причем для комплексирования достаточно располагать сведениями только об интерфейсных частях сопрягаемых объектов. Если же интерфейсные части выполнены в соответствии с заранее оговоренными правилами и соглашениями, которых должны придерживаться все создатели открытых систем определенного приложения, то проблема создания новых сложных систем существенно упрощается. Из этого следует, что основой создания открытых систем являются стандартизация и унификация в области информационных технологий.

Значительное развитие концепция открытости получила в области построения вычислительных сетей, что нашло выражение в эталонной модели взаимосвязи открытых систем, поддерживаемой рядом международных стандартов. Идеи открытости широко используются при построении программного, информационного и лингвистического обеспечений АС; в результате повышается степень универсальности программ и расширяются возможности их адаптации к конкретным условиям.

Аспекты открытости отражены в стандартизации:

- *API (Application Program Interface)* - интерфейсов прикладных программ с операционным окружением, в том числе системных вызовов и утилит операционной системы (ОС), т. е. связей с ОС;
- межпрограммного интерфейса, включая языки программирования;
- сетевого взаимодействия;
- пользовательского интерфейса, в том числе средств графического взаимодействия пользователя с ЭВМ;
- средств защиты информации.

Стандарты, обеспечивающие открытость ПО, в настоящее время разрабатываются такими организациями, как **ISO (International Standard Organization)**, **IEEE (Institute of Electrical and Electronics Engineers)**, **EIA (Electronics Industries Association)** и др.

Стандарты **POSIX (Portable Operating System Interface)** предназначены для **API** и составляют группу стандартов **IEEE 1003**. В этих стандартах содержатся перечень и правила вызова интерфейсных функций, определяются способы взаимодействия прикладных программ с ядром ОС на языке **C** (что означает преимущественную ориентацию на **ОС Unix**), даны расширения для взаимодействия с программами на других языках, способы тестирования интерфейсов на соответствие стандартам **POSIX**, правила административного управления программами и данными и т. п.

Ряд стандартов **ISO** посвящен языкам программирования. Имеются стандарты на языки **C (ISO 9899)**, **Fortran (ISO 1539)**, **Pascal (ISO 7185)** и др.

Среди других стандартов, способствующих открытости ПО АС, следует отметить стандарты графического пользовательского интерфейса, хранения и передачи графических данных, построения баз данных и файловых систем, сопровождения и управления конфигурацией программных систем и др.

Важное значение для создания открытых систем имеют унификация а стандартизация средств межпрограммного интерфейса, или, другими словами, необходимо наличие профилей АС для информационного взаимодействия программ, входящих в АС. *Профилем* открытой системы называют

совокупность стандартов и других нормативных документов, обеспечивающих выполнение системой заданных функций.

Так, в профилях АС могут фигурировать язык *Express* стандарта **STEP**, спецификация графического пользовательского интерфейса *Motif*, унифицированный язык *SQL* обмена данными между различными СУБД, стандарты сетевого взаимодействия, в профили **MCAD** может входить формат *IGES* и в случае **ECAD** - формат *EDIF* и т. п.

Упражнения и вопросы для самоконтроля

1. Дайте определение понятия «проектирование».
2. Что является предметом изучения в теории систем?
3. Назовите признаки, присущие сложной системе.
4. Приведите примеры иерархической структуры технических объектов, их внутренних, внешних и выходных параметров.
5. Приведите примеры условий работоспособности.
6. Почему проектирование обычно имеет итерационный характер?
7. Назовите основные стадии проектирования технических систем. Чем обусловлено прототипирование?
8. Дайте характеристику этапов жизненного цикла промышленной продукции.
9. Назовите основные типы промышленных АС и виды их обеспечения.
10. Какие причины привели к появлению и развитию CALS-технологий?
11. Что понимают под комплексной АС?
12. Дайте определение профиля открытой системы.
13. Чем обеспечивается открытость систем?

Лекция 3

ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

2.1. Структура технического обеспечения

Требования, предъявляемые к техническому обеспечению

Техническое обеспечение САПР включает в себя различные технические средства (**hardware**), используемые для выполнения автоматизированного проектирования, а именно ЭВМ, периферийные устройства, сетевое оборудование, а также оборудование некоторых вспомогательных систем (например, измерительных), поддерживающих проектирование.

Используемые в САПР технические средства должны обеспечивать:

- 1) выполнение всех необходимых проектных процедур, для которых имеется соответствующее ПО;
- 2) взаимодействие между проектировщиками и ЭВМ, поддержку интерактивного режима работы;
- 3) взаимодействие между членами коллектива, работающими над общим проектом.

Первое из этих требований выполняется при наличии в САПР вычислительных машин и систем с достаточными производительностью и емкостью памяти.

Второе требование относится к пользовательскому интерфейсу и выполняется за счет включения в САПР удобных средств ввода-вывода данных и прежде всего устройств обмена графической информацией.

Третье требование обуславливает объединение аппаратных средств САПР в вычислительную сеть.

Канал передачи данных — средство двустороннего обмена данными, включающее в себя аппаратуру окончания канала данных (АКД) и линию связи. **Линией связи** называют часть физической среды, используемую для распространения сигналов в определенном направлении; примерами линий связи могут служить коаксиальный кабель, витая пара проводов, волоконно-оптическая линия связи (ВОЛС). Близким является понятие **канала (канала связи)**, под которым понимают средство односторонней передачи данных. Примером канала связи может быть полоса частот, выделенная одному передатчику при радиосвязи. В некоторой линии можно образовать несколько каналов связи, по каждому из которых передается своя информация. При этом говорят, что линия разделяется между несколькими каналами.

Типы сетей

Существуют два метода разделения линии передачи данных: *временное мультиплексирование* (иначе разделение по времени, или **TDM— Time Division Method**), при котором каждому каналу выделяется некоторый квант времени, и *частотное разделение (FDM — Frequency Division Method)*, при котором каналу выделяется некоторая полоса частот.

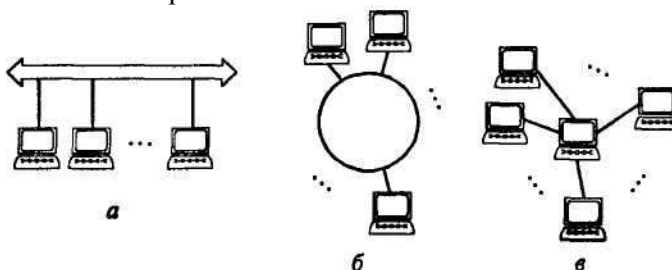


Рис. 2.2. Варианты топологии локальных вычислительных сетей:

a — шинная; *b* — кольцевая; *c* — звездная

В САПР небольших проектных организаций, насчитывающих не более единиц - десятков компьютеров, которые размещены на малых расстояниях один от другого (например, в одной или нескольких соседних комнатах), объединяющая компьютеры сеть является локальной. **Локальная вычислительная сеть (ЛВС)**, или **LAN (Local Area Network)**, имеет линию связи, к которой подключаются все узлы сети. При этом топология соединений узлов (рис. 2.2) может быть шинная (**bus**),

кольцевая (**ring**), звездная (**star**). Протяженность линии и число подключаемых узлов в ЛВС ограничены.

В более крупных по масштабам проектных организациях в сеть включены десятки-сотни и более компьютеров, относящихся к разным проектным и управленческим подразделениям и размещенных в помещениях одного или нескольких зданий. Такую сеть называют **корпоративной**. В ее структуре можно выделить ряд ЛВС, называемых **подсетями**, и средства связи ЛВС между собой. В эти средства входят коммутационные серверы (блоки взаимодействия подсетей). Если коммутационные серверы объединены отделенными от ЛВС подразделений каналами передачи данных, то они образуют новую подсеть, называемую **опорной** (или транспортной), а вся сеть оказывается иерархической структуры.

Если здания проектной организации удалены друг от друга на значительные расстояния (вплоть до их расположения в разных городах), то корпоративная сеть по своим масштабам становится **территориальной сетью (WAN— Wide Area Network)**. В территориальной сети различают **магистральные** каналы передачи данных (магистральную сеть), имеющие значительную протяженность, и каналы передачи данных, связывающие ЛВС (или совокупность ЛВС отдельного здания или кампуса) с магистральной сетью и называемые **абонентской линией** или соединением «**последней мили**».

Развитие виртуальных предприятий, работающих на основе **CALS**-технологий, с необходимостью подразумевает информационные обмены через территориальные сети, как правило, через Internet. Нужно, однако, отметить, что использование сетей общего пользования существенно усложняет задачу обеспечения информационной безопасности.

Структура **ТО САПР** для крупной организации представлена на рис. 2.3. Здесь показана типичная структура крупных корпоративных сетей САПР, называемая архитектурой **клиент — сервер**. В сетях клиент — сервер выделяется один или несколько узлов, называемых **серверами**, которые выполняют в сети управляющие или общие для многих пользователей проектные функции, а остальные узлы (рабочие места) являются терминальными, их называют **клиентами**, в них работают пользователи. В общем случае сервером называют совокупность программных средств, ориентированных на выполнение определенных функций, но если эти средства сосредоточены на конкретном узле вычислительной сети, то тогда понятие «сервер» относится именно к узлу сети.

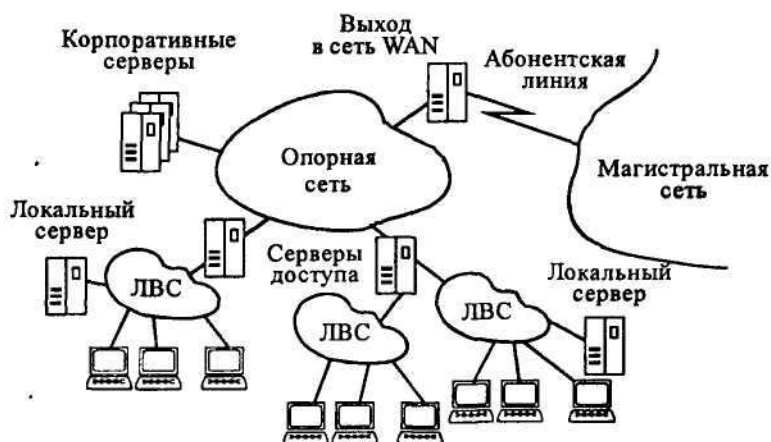


Рис. 2.3. Структура корпоративной сети САПР

Сети клиент—сервер различают по характеру распределения функций между серверами, другими словами, их классифицируют по типам серверов. Различают **файл-серверы** для хранения файлов, разделяемых многими пользователями, **серверы баз данных АС**, **серверы приложений** для решения конкретных прикладных задач, **коммутационные серверы** (называемые также блоками взаимодействия сетей или серверами доступа) для взаимосвязи сетей и подсетей, **специализированные серверы** для выполнения определенных телекоммуникационных услуг, например серверы электронной почты.

В соответствии со способами коммутации различают сети с **коммутацией каналов** и **коммутацией пакетов**. В первом случае при обмене данными между узлами **A** и **B** в сети создается физическое соединение между **A** и **B**, которое во время сеанса связи используется только этими абонентами. Примером сети с коммутацией каналов может служить телефонная сеть. Здесь передача информации происходит быстро, но каналы связи используются неэффективно, так как при обмене данными возможны длительные паузы и канал «простаивает». При коммутации пакетов физического

соединения, которое в каждый момент сеанса связи соединяло бы абонентов *A* и *B*, не создается. Сообщения разделяются на порции, называемые **пакетами**, которые передаются в разветвленной сети от *A* к *B* или обратно через промежуточные узлы с возможной буферизацией (временным запоминанием) в них. Таким образом, любая линия может разделяться многими сообщениями, попеременно пропуская при этом пакеты разных сообщений с максимальным заполнением упомянутых пауз.

Эталонная модель взаимосвязи открытых систем

Для удобства модернизации сложных информационных систем их делают максимально открытыми, т. е. приспособленными для внесения изменений в некоторую часть системы при сохранении неизменными остальных частей. В отношении вычислительных сетей реализация концепции открытости привела к появлению **эталонной модели взаимосвязи открытых систем (ЭМВОС)**, предложенной Международной организацией стандартизации (**ISO — International Standard Organization**). В этой модели дано описание общих принципов, правил, соглашений, обеспечивающих взаимодействие информационных систем и называемых **протоколами**.

Информационную сеть в ЭМВОС рассматривают как совокупность функций (протоколов), которые подразделяют на группы, называемые **уровнями**. Именно разделение на уровни позволяет вносить изменения в средства реализации одного уровня без перестройки средств других уровней, что значительно упрощает и удешевляет модернизацию средств по мере развития техники.

Различают семь уровней ЭМВОС (OSI).

1. На **физическом (physical)** уровне осуществляется представление информации в виде электрических или оптических сигналов, преобразование формы сигналов, выбор параметров физических сред передачи данных, организуется передача информации через физические среды.
2. На **канальном (link)** уровне выполняется обмен данными между соседними узлами сети, т. е. узлами, непосредственно связанными физическими соединениями без других промежуточных узлов. Отметим, что пакеты канального уровня обычно называют **кадрами**.
3. На **сетевом (network)** уровне происходит формирование пакетов по правилам тех промежуточных сетей, через которые проходит исходный пакет, и **маршрутизация** пакетов, т. е. определение и реализация маршрутов, по которым передаются пакеты. Другими словами, маршрутизация сводится к образованию логических каналов. **Логическим каналом** называют виртуальное соединение двух или более объектов сетевого уровня, при котором возможен обмен данными между этими объектами. Понятию логического канала необязательно соответствует физическое соединение линий передачи данных между связываемыми пунктами. Это понятие введено для абстрагирования от физической реализации соединения. Еще одной важной функцией сетевого уровня после маршрутизации является контроль нагрузки на сеть с целью предотвращения перегрузок, отрицательно влияющих на работу сети.
4. На **транспортном (transport)** уровне обеспечивается связь между оконечными пунктами (в отличие от предыдущего сетевого уровня, на котором обеспечивается передача данных через промежуточные компоненты сети). К функциям транспортного уровня относятся мультиплексирование и демultipлексирование (сборка-разборка сообщений на пакеты в конечных пунктах), обнаружение и устранение ошибок в переданных данных, задание требуемого уровня услуг (например, заказанных скорости и надежности передачи).
5. На **сеансовом (session)** уровне определяются тип связи (дуплекс или полудуплекс), начало и окончание заданий, последовательность и режим обмена запросами и ответами взаимодействующих партнеров.
6. На **представительном (presentation)** уровне реализуются функции представления данных (кодирование, форматирование, структурирование). Например, на этом уровне выделенные для передачи данные преобразуются из одного кода в другой, в частности, с целью шифрования.
7. На **прикладном (application)** уровне определяются и оформляются в сообщения те данные, которые подлежат передаче по сети.

2.2. Аппаратура рабочих мест в автоматизированных системах проектирования и управления

Вычислительные системы в САПР

В зависимости от назначения существуют АРМ конструктора, АРМ технолога, АРМ руководителя проекта и т. п. Они могут различаться составом периферийных устройств, характеристиками ЭВМ.

Специально выпускаемые ЭВМ как серверы высокой производительности обычно имеют структуру симметричной многопроцессорной вычислительной системы. В них системная память разделяется всеми процессорами, каждый процессор может иметь свою сверхоперативную память сравнительно небольшой емкости, число процессоров невелико (единицы, редко более десяти). Например, сервер **Enterprise 250** (Sun Microsystems) имеет один-два процессора, его цена в зависимости от комплектации колеблется в диапазоне 24 ... 56 тыс. долл., а сервер **Enterprise 450** с четырьмя процессорами стоит от 82 до 95 тыс. долл.

Периферийные устройства

Для ввода графической информации с имеющихся документов в САПР используют дигитайзеры и сканеры.

Дигитайзер применяют для ручного ввода. Он имеет вид кульмана, по его электронной доске перемещается курсор, на котором расположены визир и кнопочная панель. Курсор имеет электромагнитную связь с сеткой проводников в электронной доске. При нажатии кнопки в некоторой позиции курсора происходит занесение в память информации о координатах этой позиции. Таким образом может осуществляться ручная «сколка» чертежей.

Для автоматического ввода информации с имеющихся текстовых или графических документов используют сканеры планшетного или протяжного типа. Способ считывания оптический. В сканирующей головке размещаются оптоволоконные самофокусирующиеся линзы и фотоэлементы. Разрешающая способность в разных моделях составляет от 300 до 800 точек на дюйм (этот параметр часто обозначают dpi). Считанная информация имеет растровую форму, программное обеспечение сканера представляет ее в одном из стандартных форматов, например **TIFF**, **GIF**, **PCX**, **JPEG**, и для дальнейшей обработки может выполнить векторизацию — перевод графической информации в векторную форму, например в формат **DXF**.

Для вывода информации применяют принтеры и плоттеры. Первые из них ориентированы на получение документов малого формата (А3, А4), вторые — на вывод графической информации на широкоформатные носители.

2.3. Методы доступа в локальных вычислительных сетях

Множественный доступ с контролем несущей и обнаружением конфликтов

Доступом к сети называют взаимодействие станции (узла сети) со средой передачи данных для обмена информацией с другими станциями. Управление доступом к сети — это установление последовательности, в которой станции получают доступ к среде передачи данных.

Различают случайные и детерминированные методы доступа. Среди случайных методов наиболее известен метод множественного доступа с контролем несущей и обнаружением конфликтов (МДКН/ОК). Англоязычное название метода — *Carrier Sense Multiple Access /Collision Detection (CSMA/CD)*. Этот метод основан на контроле наличия электрических колебаний (несущей) в линии передачи данных и устранении конфликтов, возникающих в случае попыток одновременного начала передачи двумя или более станциями, путем повторения попыток захвата линии через случайный отрезок времени.

Среди детерминированных методов преобладают маркерные методы доступа. Маркерный метод — метод доступа к среде передачи данных в ЛВС, основанный на передаче полномочий передающей станции с помощью специального информационного объекта, называемого маркером. Под полномочием понимается право инициировать определенные действия, динамически предоставляемые объекту, например станции данных в информационной сети.

2.4. Локальные вычислительные сети Ethernet

Состав аппаратуры

Одной из первых среди ЛВС шинной структуры была создана сеть **Ethernet**, разработанная фирмой **Xerox**. В этой сети был применен метод доступа **МДКН/ОК**. Позднее Ethernet стала основой стандарта **IEEE 802/3**. Другой вариант шинных ЛВС соответствует стандарту **IEEE 802/4**, описывающему сеть с эстафетной передачей маркера.

Технология **Ethernet** наиболее распространена в ЛВС. Так, по данным на 1996 г. 85 % всех компьютеров в ЛВС были в сетях **Ethernet**.

Сетевой контроллер реализует принятый метод доступа к каналу, а также в случае метода МДКН /ОК осуществляет действия по выработке сигнала затора, по задержке в передаче при наличии конфликта или при занятом моноканале, по формированию кадров, кодированию (декодированию) электрических сигналов в (из) специальный последовательный код, называемый манчестерским, по распознаванию адреса в передаваемых по сети сообщениях.

Шестибайтовый MAC-адрес — уникальный номер сетевой платы, он назначается изготовителем по выданной ему лицензии на определенный диапазон адресов.

Разновидности сетей Ethernet

1. **Thick Ethernet** (шина с «толстым» кабелем); принятое обозначение варианта **10Base-5**, где первый элемент «10» характеризует скорость передачи данных по линии 10 Мбит/с, последний элемент «5» — максимальную длину сегмента кабеля (в сотнях метров), т. е. 500 м. Другие параметры сети: максимальное число сегментов 5; максимальное число узлов на одном сегменте 100; минимальное расстояние между узлами 2,5 м. Здесь под *сегментом* кабеля понимается часть кабеля, используемая в качестве линии передачи данных и имеющая на концах согласующие элементы (*терминаторы*) для предотвращения отражения сигналов.
2. **Thin Ethernet** (шина с «тонким» кабелем); принятое обозначение **10Base-2**: максимальное число сегментов 5; максимальная длина сегмента 185 м; максимальное число узлов на одном сегменте 30; минимальное расстояние между узлами 0,5 м; скорость передачи данных по линии 10 Мбит/с.
3. **TwistedPair Ethernet**; принятое обозначение **10Base-T**; это кабельная сеть с использованием витых пар проводов и концентраторов, называемых также распределителями или хабами (hubs). В состав сетевого оборудования входят активные (АН) и пассивные (РН) концентраторы (active and passive hubs), различие между которыми заключается в наличии или отсутствии усиления сигналов и в количестве портов.
4. **Fiber Optic Ethernet** (шина на основе оптоволоконного кабеля), обозначение **10Base-F**; применяется для соединений точка—точка, например для соединения двух конкретных распределителей в кабельной сети. Максимальные длины—в пределах 2 ... 4 км.
5. **RadioEthernet** (стандарт **IEEE 802/11**). Среда передачи данных — радиоволны, распространяющиеся в эфире. Структура сети может быть «постоянной» при наличии базовой кабельной сети с точками доступа от узлов по радиоканалам или «временной», когда обмены между узлами происходят только по радиоканалам. Скорость передачи данных составляет 11 Мбит/с при работе в диапазоне волн 2,4 ГГц, возможно увеличение скорости до 54 Мбит/с при переходе в диапазон 5 ГГц.
6. **Fast Ethernet**, иначе называемая **100Base-X** или **100Base-T** (стандарт **IEEE 802/3u**). Информационная скорость 100 Мбит/с. В этой сети применен метод доступа МДКН /ОК. Используется для построения скоростных ЛВС (последовательно включается не более двух хабов),
7. **Gigabit Ethernet (1000Base-X)**. В этом варианте получены гигабитные скорости. В соответствии со стандартом **IEEE 802.3z** имеются разновидности на ВОЛС с длиной волны 830 или 1270 нм (соответственно 1000Base-SX и 1000Base-LX) на расстояниях до 550 и 5000 м, на витой паре категории 5 (1000Base-CX) на расстояниях до 25 м. Скорость до 1 Гбит/с. Такая скорость достигается благодаря следующим решениям.

2.5. Сети кольцевой топологии

Сеть Token Ring

В кольцевых локальных сетях сигналы циркулируют по кольцу, состоящему из ряда отрезков линии связи, которые соединяют пары соседних узлов. Эти отрезки соединяются в узлах через повторители сигналов, выполняющие функции приема и передачи сигналов как из кольца и в кольцо, так и между АКД и линией. Повторители вносят некоторую задержку в передачу сигналов, поэтому общая задержка зависит от числа станций, включенных в кольцо.

Сеть FDDI

Сеть FDDI относится к высокоскоростным сетям, имеет кольцевую топологию, использует ВОЛС и специфический вариант маркерного метода доступа.

В основном варианте сети применено двойное кольцо на ВОЛС. Обеспечивается информационная скорость **100 Мбит/с**. Расстояние между крайними узлами — до 200 км, между соседними станциями — не более 2 км. Максимальное число узлов 500. В ВОЛС применяются волны длиной 1300 нм.

Два кольца ВОЛС используются одновременно. Станции можно подключать к одному из колец или к обоим сразу. Использование конкретным узлом обоих колец позволяет получить для этого узла суммарную пропускную способность 200 Мбит/с.

Проводные и беспроводные линии связи.

Проводные линии связи в вычислительных сетях представлены коаксиальными кабелями и витыми парами проводов.

Используются коаксиальные кабели: «толстый» диаметром 12,5 мм и «тонкий» диаметром 6,25 мм. «Толстый» кабель имеет меньшее затухание, лучшую помехозащищенность, что обеспечивает возможность работы на больших расстояниях, но он плохо гнется, что затрудняет прокладку соединений в помещениях, и дороже «тонкого».

Существуют экранированные (*STP — Shielded Twist Pair*) и неэкранированные (*UTP — Unshielded Twist Pair*) пары проводов. Экранированные пары сравнительно дороги, поэтому чаще используются неэкранированные пары, имеющие несколько категорий (типов).

Обычный телефонный кабель — пара категории 1. Пара категории 2 может использоваться в сетях с пропускной способностью до 4 Мбит/с. **Витые пары** имеют категории, начиная с третьей. Для сетей **Ethernet** (точнее, для ее варианта 10Base-T) разработана пара категории 3, а для сетей **Token Ring** — пара категории 4. Более совершенными являются неэкранированные витые пары категорий 5 и 6.

Пару категории 5 применяют при частотах до 100 МГц, в ней проводник представлен медными жилами диаметром **0,51 мм**, навитыми по определенной технологии и заключенными в термостойкую изолирующую оболочку. В высокоскоростных ЛВС на **UTP** длины соединений обычно не превышают 100 м. Затухание в паре категории 5 на 100 МГц и при длине 100 м составляет около **24 дБ**, при 10 МГц и 100 м — около **7 дБ**.

Примерами пар категорий 6 и 7 могут служить кабели, выпускаемые фирмой **РКС**, в них размещается по четыре пары проводов, каждая со своим цветом полиэтиленовой изоляции. В случае категории 6 оболочка кабеля имеет диаметр 5 мм, используются медные проводники диаметром 0,5 мм, затухание на 100 МГц — около **22 дБ**. В случае категории 7 каждая пара дополнительно заключена в экранирующую алюминиевую фольгу, диаметр оболочки увеличен до 8 мм, затухание на 100 МГц составляет около **20 дБ**, на 600 МГц - **50 дБ**.

Радиоканалы

В беспроводных радиоканалах передача информации осуществляется с помощью радиоволн. В информационных сетях используются диапазоны от сотен мегагерц до десятков гигагерц.

Для организации канала передачи данных в диапазонах дециметровых волн (**902... 928 МГц** и **2,4... 2,5 ГГц**) необходима регистрация в Госсвязьнадзоре. Работа в диапазоне **5,725 ... 5,85 ГГц** пока лицензирования не требует.

Цифровые каналы

Для передачи аналоговых сигналов по цифровым каналам связи применяют импульсно-кодую модуляцию (ИКМ), или РСМ (**Pulse Code Modulation**). Этот вид модуляции сводится к измерению амплитуды аналогового сигнала в моменты времени, отстоящие друг от друга на dt , и к кодированию этих амплитуд цифровым кодом. Согласно *теореме Котельникова*, величину dt определяют следующим образом: для неискаженной передачи должно быть не менее двух отсчетов на период колебаний, соответствующий высшей составляющей в частотном спектре сигнала.



Рис. 2.7. Манчестерское кодирование

В *цифровых каналах* для представления двоичной информации преимущественно используют самосинхронизирующийся *манчестерский код*. Пример манчестерского кода представлен на рис. 2.7. Самосинхронизация избавляет от необходимости иметь дополнительную линию связи для передачи синхронизирующих импульсов. Самосинхронизация обеспечивается благодаря формированию синхроимпульсов из перепадов, имеющихся в каждом такте манчестерского кода.

Наиболее перспективными сетями интегрального обслуживания являются сети с цифровыми каналами передачи данных, например сети *ISDN (Integrated Service Digital Network)*.

Сети *ISDN* могут быть коммутируемыми и некоммутируемыми. Различают обычные *ISDN* со скоростями от **56 кбит/с** до **1,54 Мбит/с** и широкополосные *ISDN (Broadband ISDN, или B-ISDN)* со скоростями **155 ... 2048 Мбит/с**.

Организация дуплексной связи

Для организации дуплексной связи, т. е. одновременной передачи информации по линии в обоих направлениях, используют следующие способы:

- четырехпроводная линия связи — одна пара проводов для прямой и другая — для обратной передачи, что, естественно, дорого;
- частотное разделение — прямая и обратная передачи ведутся на разных частотах, но при этом полоса для каждого направления сужается более чем вдвое по сравнению с полосой симплексной (однонаправленной) связи;
- эхо-компенсация—при установлении соединения с помощью посылки зондирующего сигнала определяются параметры (запаздывание и мощность) эха — отраженного собственного сигнала; в дальнейшем из принимаемого сигнала вычитается эхо собственного сигнала.

2.7. Стеки протоколов и типы сетей в автоматизированных системах

Протокол TCP

Протоколы, используемые совместно в сетях определенного типа, объединяют в совокупности, называемые *стеками протоколов*. Широко известны стеки протоколов *TCP/IP, SPX/IPX, X.25, Frame Relay (FR), ATM*, семиуровневые протоколы *ЭМВОС*.

Наибольшее распространение получили протоколы *TCP/IP* в связи с их использованием в качестве основных в сети Internet. *TCP/IP* — пятиуровневые протоколы, но базовыми среди них, давшими название всей совокупности, являются протокол транспортного уровня *TCP (Transmission Control Protocol)* и протокол сетевого уровня *IP (Internet Protocol)*. Эти протоколы поддерживаются такими ОС, как Unix и Windows-95/NT.

TCP — дуплексный транспортный протокол с установлением соединения. Под установлением соединения подразумевают установление виртуального канала в сети путем обмена запросом и согласием на соединение между отправителем и получателем сообщения. К другим функциям *TCP* относятся упаковка и распаковка пакетов на концах транспортного соединения; управление потоком - получатель одновременно с подтверждением правильности передачи сообщает размер окна, т. е. число пакетов, которые получатель готов принять, или, что практически то же самое, число пакетов, которые отправитель может послать в сеть, не дожидаясь получения подтверждения об их правильном приеме;

помещение срочных данных между специальными указателями, т. е. возможность управлять скоростью передачи.

В программном обеспечении протокола TCP имеется программа-агент, которая постоянно готова к работе и при приходе запроса и установлении соединения генерирует свою копию для обслуживания создаваемого соединения, а сама программа-родитель ждет новых вызовов.

В схеме установления соединения в сетях клиент — сервер предусмотрена посылка клиентом запроса на соединение (команда **ACTIVE_OPEN**) с указанием адреса сервера, тайм-аута (времени жизни), уровня секретности. Можно сразу же поместить в запрос данные (тогда используется команда **ACTIVE_OPEN_WITH_DATA**). Если сервер готов к связи, он отвечает командой согласия (**OPEN_RECEIVED**), в которой назначает номер соединения. Далее командой **SEND** посылаются данные, а командой **DELIVER** подтверждается их получение. Разъединение выполняется обменом командами **CLOSE** и **CLOSING**.

В одноранговых сетях используется трехшаговая процедура установления соединения. Сначала инициатор **A** посылает запрос на установление прямого соединения, затем приемник **B** отвечает согласием и посылает запрос на установление обратного соединения, узел **A** отвечает на это согласием.

Протокол IP

Сетевой протокол IP — *дейтаграммный* сетевой протокол, т. е. протокол без установления соединения. В дейтаграммных протоколах сообщение разбивается на дейтаграммы. *Дейтаграмма* — это пакет, передаваемый независимо от других частей одного и того же сообщения в вычислительных сетях с коммутацией пакетов. Дейтаграммы одного и того же сообщения могут передаваться в сети по разным маршрутам и поступать к адресату в произвольной последовательности, что требует дополнительных операций по сборке сообщения из дейтаграмм в узле-получателе. На внутренних участках маршрута контроль правильности передачи не предусмотрен и надежность связи обеспечивается лишь контролем в конечном узле.

Четвертая версия протокола **IPv4**. Один из недостатков этой версии — 32-битный размер адреса. Действительно, 32 бита соответствуют $2^{32} \ll 4,3$ млрд адресов, а это в связи с бурным ростом числа компьютеров в Internet уже вызывает затруднения с распределением адресного пространства. Поэтому разработана и постепенно вводится в действие версия **IPv6**, в которой применена другая структура заголовка и адресации. Как частный случай, в структуре **IPv6**-адреса можно разместить **IPv4**-адрес, т. е. сети с протоколами этих версий могут работать совместно. Пока (к 2002 г.) большинство доменов Internet работает по протоколу **IPv4**.

Адресация в TCP/IP

В протоколах TCP/IP различают два типа адресов. На канальном уровне используют адреса, называемые *физическими*. Это шестибайтовые адреса сетевых плат, присваиваемые изготовителем контроллеров (как уже отмечалось, каждый изготовитель вместе с лицензией на изготовление получает уникальный диапазон адресов). На сетевом уровне используют сетевые адреса, иначе называемые *виртуальными* или *логическими*.

Различают понятия сетевых адреса и имени, имеющих цифровое и буквенное выражения соответственно.

Сетевой адрес называют IP-адресом. В **IPv4** это четырехбайтовый код, состоящий из двух частей: адреса сети и адреса узла (заметим, что узел, имеющий IP-адрес, называют хостом). Имя характеризует пользователя. Его составляют в соответствии с доменной системой имен. Соответствие между IP-адресом и IP-именем хоста устанавливается специальной службой имен. В Internet это **DNS (Domain Name Service)**, в семиуровневой модели **ISO** — стандарт **X.500**.

При обращении к сети пользователь, отправляющий сообщение, задает IP-имя получателя. Поскольку маршрутизация в сети осуществляется по IP-адресам, то с помощью серверов **DNS** осуществляется перевод указанного IP-имени в IP-адрес.

В локальной сети, где используются шестибайтовые адреса, называемые **MAC-адресами**, требуется преобразование IP-имен в **MAC-адреса**. Это преобразование выполняется с помощью специального протокола **ARP**, имеющегося в стеке TCP/IP. Для этого создаются **ARP-таблицы** соответствия **IP** и **MAC** адресов данной сети.

Маршрутизация в Internet организована по иерархическому принципу. Имеются уровни ЛВС и корпоративных сетей; маршрутных доменов, в каждом из которых используются единые протоколы и

алгоритмы маршрутизации; административных доменов, каждый из которых соответствует некоторой ассоциации и имеет единое управляющее начало. В маршрутных доменах есть внешние маршрутизаторы для связи с другими маршрутными или административными доменами.

Обращение из некоторого узла к другому узлу в Internet (например, из wwwcdl.bmstu.ru по адресу <http://www.eevl.ac.uk>) происходит следующим образом.

Сначала IP-имя переводится в IP-адрес. Для этого происходит обращение к местному серверу (**bmstu**), и если там сведений о сети назначения нет, то происходит переход к серверу следующего, более высокого уровня (**ru** и далее по иерархии вниз до получения IP-адреса хоста назначения. В местном **DNS**-сервере могут быть сведения об IP-адресах хостов из удаленных доменов, если к ним происходят достаточно частые обращения из данного домена.

После получения IP-адреса узел-отправитель сравнивает номер своей сети (подсети) с номером сети, указанным в IP-адресе получателя в заголовке пакета. Если номера совпадают, то узел-отправитель с помощью имеющейся в его памяти **ARP**-таблицы переводит IP-адрес в **MAC**-адрес, по которому и доставляется пакет средствами канального уровня. Если в **ARP**-таблице строки с нужным **MAC**-адресом не оказалось, то по сети широковещательно, т. е. по всем узлам данной сети, распространяется **ARP**-запрос. Все узлы вскрывают этот запрос, но только узел, имеющий указанный в запросе IP-адрес, откликается своим **MAC**-адресом. Далее пакет отправляется адресату, одновременно строка с найденным **MAC**-адресом заносится в **ARP**-таблицу узла-отправителя. Если номера сетей не совпадают, то пакет пересылается маршрутизатору, который с помощью своей таблицы определяет, через какой из своих портов направлять пакет дальше.

Как отмечено выше, продолжающийся рост числа узлов в Internet привел к появлению версии **IPv6** протокола IP.

В целом IP-заголовок в протоколе **IPv6** состоит из **40** байт и включает следующие поля: *версия протокола (4) — приоритет (4) — параметры обслуживания (24) — длина пакета (16) — тип протокола (8) — тип следующего заголовка (8) — лимит числа переходов (8) — адреса отправителя и получателя (по 128).*

За основным заголовком в **IPv6**-пакете могут следовать дополнительные, используемые для указания пользователю той или иной служебной информации, например способа шифрования или способа фрагментации. Лимит числа переходов — это максимально допустимое число маршрутизаторов на пути дейтаграммы. Превышение этого числа приводит к ликвидации пакета.

Другие протоколы стека TCP/IP

В стек протоколов TCP/IP входит ряд других протоколов. Например, на транспортном уровне это протокол **UDP (User Datagram Protocol)** — транспортный протокол без установления соединения, он значительно проще **TCP**, но его используют чаще всего для сообщений, уместяющихся в один пакет. После оформления **UDP**-пакета он передается с помощью средств IP к адресату, который по заголовку IP-пакета определяет тип протокола и передает пакет не агенту **TCP**, а агенту **UDP**. Агент определяет номер порта и ставит пакет в очередь к этому порту.

На базе протокола **SNMP** разработан ряд мощных средств управления, примерами которых могут служить продукт **ManageWISE** фирмы **Novell** или система **UnicenterTNG** фирмы **Computer Associates**. С их помощью администратор сети может выполнять следующие действия:

- 1 — строить 2D-изображение топологии сети, причем на разных иерархических уровнях, перемещаясь от региональных масштабов до подсетей ЛВС (при интерактивной работе);
- 2 — разделять сеть на домены управления по функциональным, географическим или другим принципам с установлением своей политики управления в каждом домене;
- 3 — разрабатывать нестандартные агенты с помощью имеющихся инструментальных средств.

Сети X.25 и Frame Relay

Сети **X.25**, работающие по одноименному стеку протоколов, предложенному международным телекоммуникационным союзом **ITU (International Telecommunication Union)**, относятся к первому поколению сетей коммутации пакетов. Протоколы **X.25** разработаны еще в 1976 г. В свое время они получили широкое распространение, в России их популярность сохраняется, поскольку эти сети хорошо приспособлены к работе на телефонных каналах невысокого качества, составляющих в России значительную долю каналов связи. С помощью сетей **X.25** удобно соединять локальные сети в территориальную сеть, устанавливая между ними мосты **X.25**.

Стандарт **X.25** относится к трем нижним уровням ЭМВОС, т. е. включает протоколы физического, канального и сетевого уровней. На сетевом уровне используется коммутация пакетов.

В сетях пакетной коммутации **Frame Relay (FR)** в отличие от сетей **X.25** обеспечивается большая скорость передачи данных (до **45 Мбит/с**) за счет исключения контроля ошибок в промежуточных узлах, так как контроль, адресация, инкапсуляция и восстановление выполняются в конечных пунктах, т. е. на транспортном уровне. В промежуточных узлах ошибочные пакеты могут только отбрасываться, а запрос на повторную передачу происходит от конечного узла средствами уровня, выше сетевого. Но для реализации **FR** нужны помехоустойчивые каналы передачи данных.

Сети АТМ

Технология асинхронной передачи данных, реализованная в сетях **АТМ (Asynchronous Transfer Mode)**, относится к перспективным технологиям, обеспечивающим высокие скорости передачи разнородной информации (данных, речевых и видеосигналов) на значительные расстояния. Действительно, передача голосовой и видеоинформации обычно требуется в режиме реального времени, видеоинформация характеризуется большими объемами, и, следовательно, задержки должны быть только малыми (например, для голосовой связи—не более **6 сек**).

Технология **АТМ** представляет собой быструю коммутацию коротких пакетов фиксированной длины (**53 байт**), называемых ячейками. В силу этой причины и саму технологию **АТМ** иногда называют коммутацией ячеек.

Сети **АТМ** относят к сетям с установлением соединения, но возможны варианты и без установления соединения. Соединения могут быть постоянными и динамическими. Первые устанавливаются и разрываются администратором сети, их действие продолжительно, для каждого нового обмена данными между абонентами постоянного соединения не нужно тратить время на его установление. Вторые устанавливаются и ликвидируются автоматически для каждого нового сеанса связи.

Каждое соединение получает свой идентификатор, который указывается в заголовке ячеек. При установлении соединения каждому коммутатору на выбранном пути следования данных передается таблица соответствия идентификаторов и портов коммутаторов. Коммутатор, распознав идентификатор, направляет ячейку в нужный порт. Непосредственное указание в заголовке адресов получателя и отправителя не требуется, заголовок короткий — всего 5 байт.

Высокие скорости в **АТМ** обеспечиваются рядом технических решений.

Во-первых, большое число каналов с временным мультиплексированием (**ТОМ**) можно использовать для параллельной передачи частей одного и того же «объемного» сообщения (*статистическое мультиплексирование*). При этом цикл синхронизации состоит из отдельных участков, длины участка и ячейки совпадают. Под конкретное сообщение можно выделить N интервалов, совокупность которых называют виртуальным каналом. Скорость передачи можно регулировать, изменяя N . Если сеть **АТМ** оказывается перегруженной, то во избежание потери информации возможна буферизация данных для выравнивания загрузки каналов. Регулирование загрузки (управление потоком) осуществляется периодическим включением (обычно через 32 кадра) служебной ЛМ-ячейки в информационный поток. В эту ячейку промежуточные коммутаторы и конечный узел могут вставлять значения управляющих битов, сигнализирующие о перегрузке или недогрузке канала. ЛМ-ячейка от конечного узла передается в обратном направлении источнику сообщения, который может соответственно изменить режим передачи. В частности, применяется режим занятия всех свободных ресурсов при перегрузке. Таким образом, происходит динамическое перераспределение нагрузки.

Во-вторых, отрицательные квитанции при искажениях собственно сообщений (но не заголовков) возможны только от конечного пункта. Это исключает потери времени в промежуточных пунктах на ожидание подтверждений. Такой способ иногда называют *коммутацией кадров* (в отличие от коммутации пакетов). Контрольный код (четырёхбайтный циклический) для информационной части сообщения имеется только в конце последнего пакета сообщения.

В-третьих, упрощена маршрутизация. Собственно установление соединения выполняется аналогично этой процедуре в ТСП/ IP. Однако далее номер рассчи-ганного маршрута помещается в заголовок каждого пакета, и для них не нужно заново определять маршрут по таблицам маршрутизаторов при прохождении через сеть. Такую передачу называют *маршрутизацией от источника*. Другими словами, осуществляется передача с установлением соединения (в отличие, например, от IP). При этом клиент направляет серверу запрос в виде специального управляющего кадра. Кадр проходит через промежуточные маршрутизаторы и (или) коммутаторы, в которых соединению

(каналу) присваивается номер **VPI/VCI** (идентификаторы) маршрута. Если передача адресована нескольким узлам, то соответствующие идентификаторы в коммутаторах приписываются нескольким каналам.

Физические среды для **ATM**-сетей — каналы **SDH** или **G1/G4/L1/E4**), реализуемые на ВОЛС, витой паре или коаксиальном кабеле. При использовании магистральной сети **SDH** для передачи информации по технологиям **ATM** или **FR** сети **ATM** и **FR** называют наложенными вторичными сетями. Доступ к транспортной сети осуществляется через специальные мультиплексоры.

Каналы **ATM** со скоростями **51, 155, 622** и **2488** Мбит/с называют каналами **OC-1, OC-3, OC-12** и **OC-48** соответственно.

Промышленные сети

В интегрированных системах проектирования и управления на уровнях цеховом и ниже используются специальные вычислительные сети АСУТП, называемые промышленными (или **Fieldbus**). В число узлов сети входят компьютеры, выполняющие функции числового управления технологическим оборудованием и функции **SCADA**.

Во встроенных системах использование оборудования разных производителей возможно, только если эти системы являются открытыми, что, в свою очередь, диктует необходимость стандартизации промышленных шин. Однако разнообразие условий работы систем и требований к ним, а также исторические причины развития технологий обуславливают использование на практике ряда унифицированных решений.

Различают последовательные и параллельные шины. Примерами последовательных шин могут служить **Fiber Channel, Fire Wire 1394, USB, Ethernet**, а параллельных шин — **VMEbus, PCI** и др. Шина **VMEbus** стандартизована в 1987 г. (стандарт **IEEE 1014**). Конструктивное оформление выполняется по стандартам Евромеханики (**IEEE 1101.10** и **IEEE 1101.11**). В крейте может быть до **21 слота**, в которых размещаются платы унифицированных размеров. Информационная скорость—до **320 Мбайт/с**. Шина эффективно работает в условиях большого числа прерываний от устройств ввода-вывода, что важно для встроенного оборудования.

Сетевое коммутационное оборудование

Узлы в средах передачи данных, выполняющие функции связи между частями сложной сети (**internetworking**), составляют *сетевое (коммутационное) оборудование*. В сетевое оборудование входят повторители, мосты, концентраторы, коммутаторы, маршрутизаторы, шлюзы, модемы и др.

Повторитель (repeater) — блок взаимодействия, служащий для регенерации электрических сигналов, передаваемых между двумя сегментами ЛВС. Повторители используются в случае, если реализация ЛВС на одном сегменте кабеля (отрезке, моноканале) не допускается из-за ограничений на расстояние или на число узлов, причем при условии, что в соседних сегментах используются один и тот же метод доступа и одни и те же протоколы. Трафик в сегментах, соединенных повторителем, — общий. Повторитель может быть многопортовым. Сигнал, пришедший на один из портов, повторяется на всех остальных портах.

Концентраторы предназначены для объединения в сеть многих узлов. Так, концентраторами являются хабы в сетях **10Base-T** или **Token Ring**. Однако такие концентраторы подобно повторителям создают общую среду передачи данных без разделения трафика.

Мост (bridge) — блок взаимодействия, служащий для соединения разных подсетей, которые могут иметь неодинаковые канальные протоколы.

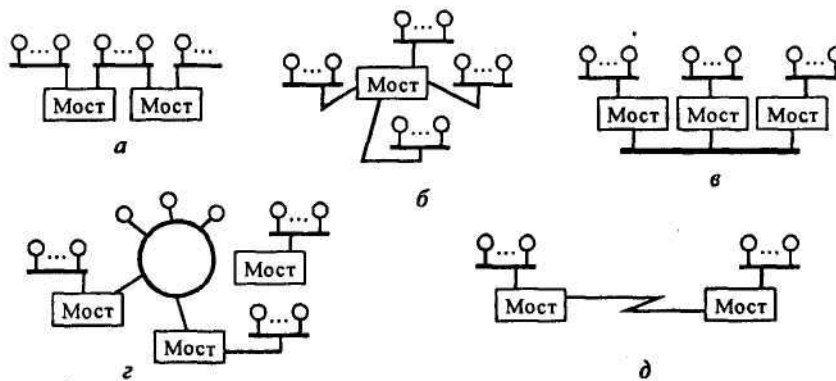


Рис. 2.12. Варианты мостовых соединений:
 — каскадное соединение; б — многопортовый мост; в — соединение через опорную магистраль;
 з — FDDI в качестве опорной сети; д — использование выделенной линии

При малых расстояниях между подсетями связь возможна через серверы подсетей, в которых размещаются интерфейсные платы, называемые внутренними мостами, и соответствующее сетевое программное обеспечение. Возможно применение внешних мостов — специально выделяемых узлов для целей сопряжения по одному в каждой из соединяемых подсетей. Внешние мосты обходятся дороже, но обеспечивают лучшие эксплуатационные характеристики. Важная функция мостов — ограничение трафика, так как локальный трафик одной подсети замыкается в ней, не проходя в другую подсеть.

Обычно мост имеет два порта, хотя существуют и многопортовые мосты. Каждый порт может оказаться входным или выходным.

Коммутаторы в отличие от концентраторов предназначены для объединения в сеть многих узлов или подсетей с разделением трафика между подсетями. Как и в мостах, пакеты передаются только в ту подсеть, для которой они предназначены, что уменьшает общую загрузку сети. Но в отличие от многопортового моста в коммутаторе возможно одновременно иметь много соединений, т. е. обеспечивается параллельная передача сообщений. Коммутаторы используют также для связи нескольких ЛВС с территориальной сетью.

Маршрутизатор (router) — блок взаимодействия, служащий для выбора маршрута передачи данных в корпоративных и территориальных сетях. С помощью маршрутизаторов могут согласовываться не только канальные протоколы, как это имеет место при применении мостов, но и сетевые протоколы. Маршрутизаторы содержат таблицы и протоколы маршрутизации в отличие от других узлов, которые могут содержать лишь локальные таблицы соответствия IP-адресов физическим адресам сетевых контроллеров в локальной сети. Маршрутизаторы могут фильтровать пакеты в соответствии с признаками, отраженными в заголовке пакета, т. е. выполнять роль брандмауэра — устройства, защищающего сеть от нежелательных вторжений извне.

Использование коммутаторов вместо маршрутизаторов (там, где это возможно) позволяет существенно повысить пропускную способность сети.

Шлюз (gateway — межсетевой преобразователь) — блок взаимодействия, служащий для соединения информационных сетей различной архитектуры и с неодинаковыми протоколами. В шлюзах предусмотрено согласование протоколов всех семи уровней ЭМВОС. Примерами шлюзов могут быть устройства, соединяющие ЛВС типа Ethernet с сетью SNA, используемой для связи больших машин фирмы IBM. Часто под шлюзом понимают сервер, имеющий единственный внешний канал передачи данных.

Упражнения и вопросы для самоконтроля

1. Поясните состав и назначение устройств графической рабочей станции.
2. Что такое «растеризация» и «векторизация»?
3. Что такое «промышленный компьютер»? Каковы его особенности?
4. Дайте сравнительную характеристику методов коммутации каналов и коммутации пакетов.
5. В чем заключается сущность методов временного (TDM) и частотного (FDM) разделения каналов?
6. Почему в МДКН/ОК повторные попытки захвата линии разрешаются через случайные интервалы времени?
7. Рассчитайте размер окна столкновений в сети 10Base-5, если линия передачи данных представлена одним сегментом кабеля длиной 500 м.

8. Что такое «стаффинг»?
9. В чем сущность метода предотвращения конфликтов в RadioEthernet?
10. Почему способ кодирования 4b/5b или 8b/10b позволяет увеличить информационную скорость передачи данных?
11. Каким образом реализуется приоритетная передача данных в сети Token Ring?
12. Почему в сетях Ethernet введено ограничение на размер кадра снизу? Рассчитайте нижнюю границу длины кадра для Gigabit Ethernet.
13. Какой может быть максимальная информационная скорость в канале передачи данных с полосой пропускания 4 кГц и отношением сигнал—помеха 130?
14. В чем заключаются преимущества перевода системы сотовой связи в более высокочастотный диапазон?
15. Рассчитайте задержку при передаче сигнала в спутниковых системах с использованием геостационарных орбит (высота спутника 36 тыс. км).
16. Сколько телефонных разговоров одновременно можно передавать по каналу T1?
17. Поясните, как действует схема эхо-компенсации.
18. Каким образом выполняется контроль правильности передачи данных по протоколу TCP?
19. Почему в IP-пакете имеется контрольный код заголовка, а не всего пакета?
20. Что такое «менеджеры» и «агенты» в сетевом программном обеспечении?
21. Назовите факторы, обуславливающие высокие скорости передачи данных в сетях ATM.
22. Что такое «маршрутизация от источника»?
23. Что понимают под виртуальной ЛВС?

Лекция 4

МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

3.1. Компоненты математического обеспечения

Математический аппарат в моделях разных иерархических уровней

К математическому обеспечению (МО) САПР относятся:

- математические модели,
- численные методы,
- алгоритмы выполнения проектных процедур.

Компоненты МО определяются базовым математическим аппаратом, специфичным для каждого из иерархических уровней проектирования.

На *микроуровне* типичные математические модели представлены дифференциальными уравнениями в частных производных вместе с краевыми условиями. К этим моделям, называемым *распределенными*, относятся многие уравнения математической физики. Объектами исследования здесь являются поля физических величин, что требуется при анализе прочности строительных сооружений или машиностроительных деталей, исследовании процессов в жидких средах, моделировании концентраций и потоков частиц в электронных приборах и т. п.

Число совместно исследуемых различных сред (число деталей, слоев материала, фаз агрегатного состояния) в практически используемых моделях микроуровня не может быть большим ввиду сложности вычислительного характера. Резко снизить вычислительные затраты в многокомпонентных средах можно, только применив иной подход к моделированию, основанный на принятии определенных допущений.

Допущение, выражаемое дискретизацией пространства, позволяет перейти к моделям *макроуровня*. Моделями макроуровня, называемыми также *сосредоточенными*, являются системы алгебраических и обыкновенных дифференциальных уравнений, поскольку независимой переменной здесь остается только время t . Упрощение описания отдельных компонентов (деталей) позволяет исследовать модели процессов в устройствах, приборах, механических узлах, число компонентов в которых может достигать до нескольких тысяч.

В тех случаях, когда число компонентов в исследуемой системе превышает некоторый порог, сложность модели системы на макроуровне вновь становится чрезмерной. Поэтому, принимая соответствующие допущения, переходят на *функционально-логический уровень*. На этом уровне используют аппарат передаточных функций для исследования аналоговых (непрерывных) процессов или аппарат математической логики и конечных автоматов, если объектом исследования является дискретный процесс, т. е. процесс с дискретным множеством состояний.

Наконец, для исследования еще более сложных объектов, примерами которых могут служить производственные предприятия и их объединения, вычислительные системы и сети, социальные системы и другие подобные объекты, применяют аппарат теории массового обслуживания, возможно использование и некоторых других подходов, например сетей Петри. Эти модели относятся к *системному* уровню моделирования.

Требования к математическим моделям и численным методам в САПР

Основными требованиями к МО являются требования адекватности, устойчивости, экономичности. Модель всегда лишь приближенно отражает некоторые свойства объекта.

Адекватность имеет место, если модель отражает заданные свойства объекта с приемлемой точностью. Точность математической модели заключается в максимальном учете всех факторов, присутствующих в реальном физическом процессе, чтобы математическая модель позволяла моделировать этот процесс. Адекватность МО относительна, так как оценить ее мы можем только с позиций современного состояния средств измерения. Например в 16 веке Галлилео Галлилей открыл закон свободного падения, в котором описал влияние на падающее тело притяжения Земли. В 17 веке Исаак Ньютон сформулировал закон всемирного тяготения, где учел взаимное влияние двух тел как силу прямопропорциональную их массам и обратнопропорциональную квадрату расстояния между ними.

Этот закон уточнял результат Галилея как частный случай но носил качественный характер и опирался на результаты экспериментов, выполненных на тогдашней лабораторной базе. Только в 19 веке эксперименты Кавендиша и Пуассона дали возможность вычислить в формуле Ньютона **гравитационную постоянную** $G = 6,67428 \cdot 10^{-11} \text{ м}^3 \text{ с}^{-2} \text{ кг}^{-1}$ или $\text{Н м}^2 \text{ кг}^{-2}$. На этом примере мы видим как со временем менялась и уточнялась математическая модель (формула), описывающая реально существующий физический процесс, так как совершенствовались средства измерения при проведении экспериментов. Соответственно менялось наше представление об этом физическом процессе, сам же процесс все время оставался одним и тем же и не зависит от точности его описания.

Адекватность оценивается перечнем отражаемых свойств и областями адекватности. **Область адекватности** — область в пространстве параметров, в пределах которой погрешности модели остаются в допустимых пределах. Например, область адекватности линеаризованной модели поверхности детали определяется системой неравенств

$$\max |\varepsilon_0| \leq \varepsilon_{\text{доп}},$$

где ε_{ij} и $\varepsilon_{\text{доп}}$ — допущенная и предельно допустимая относительные погрешности моделирования поверхности, максимум берется по всем координатам и контролируемым точкам;

$$\varepsilon_{ij} = (x_{ij\text{ист}} - x_{ij\text{мод}}) / x_{ij\text{ист}}$$

$x_{ij\text{ист}}$, $x_{ij\text{мод}}$ — i -я координата j -той точки поверхности в объекте и модели соответственно.

Отметим, что в большинстве случаев области адекватности строятся в пространстве внешних переменных. Так, область адекватности модели электронного радиоэлемента обычно выражает допустимые для применения модели диапазоны изменения моделируемых температур, внешних напряжений, частот.

С областью адекватности тесно связана еще одна характеристика МО — **чувствительность к изменению данных**. При этом различают **устойчивые** и **неустойчивые** МО. Для устойчивых моделей можно гарантировать, что погрешность будет находиться в некоторых заданных пределах, если изменения параметров МО тоже будут происходить в ограниченном диапазоне значений. Математически устойчивость описывается как отношение изменения результата Δf к изменению конкретного параметра Δx :

$$\Delta f = \left| \frac{\partial f}{\partial x} \Delta x \right|$$

Для нескольких параметров x_1, x_2, \dots, x_n устойчивость будет определяться как сумма устойчивостей по каждому параметру:

$$|\partial f / \partial x_1| + |\partial f / \partial x_2| + |\partial f / \partial x_3| + \dots + |\partial f / \partial x_n|.$$

Для устойчивых МО это отношение меньше 1, для неустойчивых — больше 1. Примером устойчивой МО в механике является игрушка-неваляшка, а неустойчивой — спичечный коробок, поставленный на ребро. В первом случае небольшое изменение положения центра тяжести вызовет возврат к положению покоя, а во втором к нарастанию этого смещения. В радиотехнике устойчивость работы усилительных каскадов приемника обеспечивается отрицательной обратной связью, в противном случае в этих каскадах возникает возбуждение (резкий свист). Наоборот для генератора нужно, чтобы состояние было неустойчивым. И электронная схема переключалась поочередно из одного поллярного состояния в другое.

Рассмотрим еще один пример МО, описываемой уравнением

$$x^3 - 3x^2 + 3x - a = 0$$

где a — некоторый параметр.

Исследуем влияние этого параметра на результат — корень уравнения.

Для этого несколько преобразуем уравнение к виду:

$$(x-1)^3 + (1-a) = 0$$

Теперь определим корень при $a=1$. Несложно видеть, что $x_k = 1$.

Изменим a на **0,001 (0,1%** от первоначального значения) до величины $a = 1,001$. Теперь корень уравнения также изменится и будет вычислен как $x_k = 1,1$. (изменится на **10%** от первоначального значения). Налицо неустойчивая МО, так как $\partial x_k / \partial a = 10\% / 0,01\% = 1000$.

Экономичность (вычислительная эффективность) определяется затратами ресурсов, требуемых для реализации модели. Поскольку в САПР используются математические модели, далее речь пойдет о характеристиках именно математических моделей, и экономичность будет

характеризоваться затратами машинного времени и памяти.

Аналогичные требования по точности и экономичности фигурируют при выборе численных методов решения уравнений модели.

Поскольку численные методы применяются для решения МО с помощью компьютера, то для нас важным становится фактор **существования и единственности решения**. Действительно, отсутствие решения приведет к "зависанию" и закликиванию программы, то есть напрасной трате вычислительных ресурсов компьютера, а наличие нескольких решений приводит к неопределенности и отсутствию воспроизводимости результата при серии экспериментов с одними и теми же данными. Поэтому важно еще до начала запуска компьютерной программы моделирования оценить эти факторы и уточнить диапазон возможных изменений параметров.

Пример. Найти решение системы уравнений:

$$\begin{cases} y - e^{-x} = 0 \\ y + ax + b = 0 \end{cases}$$

Если рассматривать каждое уравнение системы, как отдельную функцию, то решением будут координаты общей точки на графиках обеих функций. Если с первой функцией все более или менее ясно – это убывающая функция в области положительных значений x , размещенная в первой четверти. То положение прямой, описываемой вторым уравнением будет существенно зависеть от параметров a и b .

В зависимости от них точка пересечения (решение) может быть одна, две или вообще отсутствовать.

Оценка устойчивости при параметрическом задании функции.

Приведенная ранее формула оценки устойчивости математической модели может быть успешно применена, если в математической модели можно выразить результат через каждый параметр. Однако, произвести преобразования такого рода не всегда представляется возможным. Например, если математическая модель имеет вид:

$$y = e^{-\lambda y}$$

Такой способ задания математических моделей (функций), где связь между результатом y и параметром λ не задается явно называется параметрическим. В этом случае устойчивость определяется по другой формуле. Выполним для данного примера преобразования, перенесем переменные в одну часть уравнения и введем функцию $f(y, \lambda)$ вида:

$$f(y, \lambda) = e^{-\lambda y} - y$$

Теперь воспользуемся известным соотношением для производных:

$$\frac{\partial y}{\partial \lambda} = - \frac{\frac{\partial f}{\partial \lambda}}{\frac{\partial f}{\partial y}}$$

Теперь для приведенной функции устойчивость легко может быть определена:

$$\begin{aligned} \frac{\partial f}{\partial \lambda} &= -y \cdot e^{-\lambda y} \\ \frac{\partial f}{\partial y} &= -\lambda \cdot e^{-\lambda y} - 1 \end{aligned}$$

Теперь получим формулу оценки устойчивости параметрически заданной функции:

$$\frac{\partial y}{\partial \lambda} = - \frac{-y \cdot e^{-\lambda y}}{-\lambda \cdot e^{-\lambda y} - 1}$$

После учета того, что $y = e^{-\lambda y}$ получаем окончательно:

$$\frac{\partial y}{\partial \lambda} = - \frac{-y^2}{-\lambda \cdot y - 1} = - \frac{y^2}{\lambda \cdot y + 1}$$

Полученный результат может быть выражен и в численном виде. Например, если известно, что $\lambda > 0$, то это означает, что $y < 1$ (т.к. $y = e^{-\lambda y}$), а значит y^2 тем более меньше 1. Тогда в полученной формуле можно выполнить замену, которая по крайней мере не уменьшит значение дроби:

$$\left| \frac{y^2}{\lambda \cdot y + 1} \right| \leq \left| \frac{1}{\lambda + 1} \right|$$

Теперь, учитывая, что $\lambda > 0$, получаем, что это математическая модель устойчива.

Процесс формирования математических моделей при проектировании

Вычислительный процесс при анализе состоит из этапов формирования модели и ее исследования (решения). В свою очередь, формирование модели включает две процедуры:

- во-первых, разработку моделей отдельных компонентов,
- во-вторых, формирование модели системы из моделей компонентов.

Первая из этих процедур выполняется предварительно по отношению к типовым компонентам вне маршрута проектирования конкретных объектов. Как правило, модели компонентов разрабатываются специалистами в прикладных областях, причем знающими требования к моделям и формам их представления в САПР. Обычно в помощь разработчику моделей в САПР предлагаются методики и вспомогательные средства, например, в виде программ анализа для экспериментальной отработки моделей. Созданные модели включаются в библиотеки моделей прикладных программ анализа.

На маршруте проектирования каждого нового объекта выполняется вторая процедура (рис. 3.1) — формирование модели системы с использованием библиотечных моделей компонентов. Как правило, эта процедура выполняется автоматически по алгоритмам, включенным в заранее разработанные программы анализа. Примеры таких программ имеются в различных приложениях и прежде всего в отраслях общего машиностроения и радиоэлектроники.

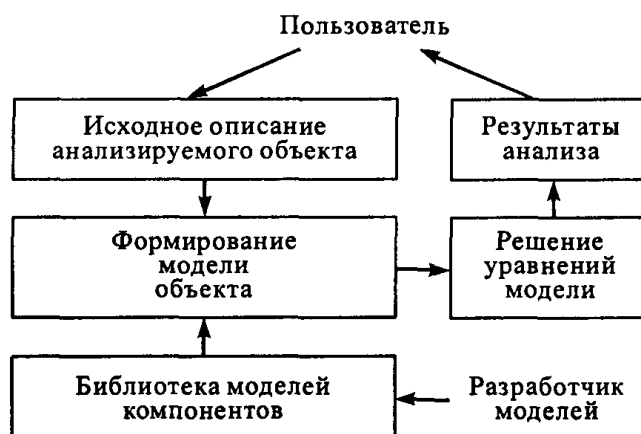


Рис. 3.1. Место процедур формирования моделей на маршрутах проектирования

При применении этих программ пользователь описывает исследуемый объект на входном языке программы анализа не в виде системы уравнений, которая будет получена автоматически, а в виде списка элементов структуры, эквивалентной схеме, эскиза или чертежа конструкции.

Лекция 5

3.2. Математические модели в процедурах анализа на макроуровне

Исходные уравнения моделей

Исходное математическое описание процессов в объектах на макроуровне представлено системами обыкновенных дифференциальных и алгебраических уравнений. Аналитические решения таких систем при типичных значениях их порядков в практических задачах получить не удастся, поэтому в САПР преимущественно используются алгоритмические модели. В этом параграфе изложен обобщенный подход к формированию алгоритмических моделей на макроуровне, справедливый для большинства приложений.

Исходными для формирования математических моделей объектов на макроуровне являются **компонентные** и **топологические** уравнения.

Компонентными уравнениями называют уравнения, описывающие свойства элементов (компонентов), другими словами, это уравнения математических моделей элементов (ММЭ).

Топологические уравнения описывают взаимосвязи в составе моделируемой системы.

В совокупности компонентные и топологические уравнения конкретной физической системы представляют собой исходную **математическую модель системы (ММС)**.

Очевидно, что компонентные и топологические уравнения в системах различной физической природы отражают разные физические свойства, но могут иметь одинаковый формальный вид. Одинаковая форма записи математических соотношений позволяет говорить о формальных аналогиях компонентных и топологических уравнений. Такие аналогии существуют для механических поступательных, механических вращательных, электрических, гидравлических (пневматических), тепловых объектов. Наличие аналогий приводит к практически важному выводу: значительная часть алгоритмов формирования и исследования моделей в САПР оказывается инвариантной и может быть применена к анализу проектируемых объектов в разных предметных областях. Единство математического аппарата формирования ММС особенно удобно при анализе систем, состоящих из физически разнородных подсистем.

В перечисленных выше приложениях **компонентные** уравнения имеют вид

$$\mathbf{F}_k(d\mathbf{V}/dt, \mathbf{V}, t) = \mathbf{0}, \quad (3.1)$$

топологические уравнения —

$$\mathbf{F}_t(\mathbf{V}) = \mathbf{0}, \quad (3.2)$$

где $\mathbf{V} = (v_1, v_2, \dots, v_n)$ — вектор фазовых переменных; t — время.

Фазовая переменная — величина, характеризующая энергетическое или информационное наполнение элемента или системы.

Различают фазовые переменные двух типов, их обобщенные наименования — **фазовые переменные типа потенциала** (например, электрическое напряжение) и **типа потока** (например, электрический ток). Каждое компонентное уравнение характеризует связи между разнотипными фазовыми переменными, относящимися к одному компоненту (например, закон Ома описывает связь между напряжением и током в резисторе), а топологическое уравнение — связи между однотипными фазовыми переменными в разных компонентах.

Модели можно представлять в виде систем уравнений или в графической форме, если между этими формами установлено взаимно однозначное соответствие. В качестве графической формы часто используют эквивалентные схемы.

Примеры компонентных и топологических уравнений

Рассмотрим несколько типов систем.

Электрические системы. В электрических системах фазовыми переменными являются электрические напряжения и токи. Компонентами систем могут быть простые двухполюсные элементы и более сложные двух- и многополюсные компоненты. К простым двухполюсникам относятся следующие элементы: сопротивление, емкость и индуктивность, характеризуемые одноименными параметрами R , C , L . В эквивалентных схемах эти элементы обозначают в соответствии с рис. 3.2, а.

Компонентные уравнения простых двухполюсников:

для сопротивления $u = iR$ (закон Ома); (3.3)

для емкости $i = Cdu/dt$; (3.4)

для индуктивности $u = Ldi/dt$, (3.5)

где u — напряжение (точнее, падение напряжения на двухполюснике); i — ток.

Эти модели лежат в основе моделей других возможных более сложных ком понентов. Большая сложность может определяться нелинейностью уравнений (3.3) — (3.5) (т. е. зависимостью R , C , L от фазовых переменных), или учетом зависимостей параметров R , C , L от температуры, или наличием более двух полюсов. Однако многополюсные компоненты могут быть сведены к совокупности взаимосвязанных простых элементов.

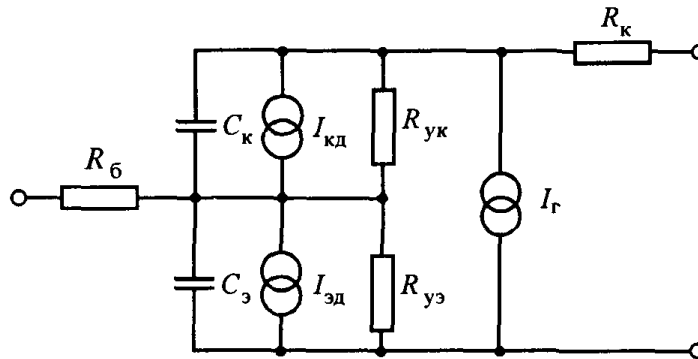


Рис. 3.3. Эквивалентная схема биполярного транзистора

Примером математической модели сложного компонента может служить модель транзистора. На рис.3.3 представлена эквивалентная схема биполярного транзистора, на которой зависимые от напряжений источники тока $i_{эд} = i_{тэ} \exp(u_э/(m\phi_T))$ и $i_{кд} = i_{тк} \exp(u_к/(m\phi_T))$ отображают статические вольт-амперные характеристики $p - n$ переходов;

$i_{тэ}$ и $i_{тк}$ — тепловые токи переходов;

$m\phi_T$ — температурный потенциал;

$u_э$ и $u_к$ — напряжения на эмиттерном и коллекторном переходах;

$C_э$ и $C_к$ — емкости переходов;

$R_уэ$ и $R_ук$ — сопротивления утечки переходов,

$R_б$ и $R_к$ — объемные сопротивления тел базы и коллектора;

$i_г = B i_{эд} - B_u i_{кд}$ — источник тока, моделирующий усилительные свойства транзистора;

B и B_u — прямой и инверсный коэффициенты усиления тока базы.

Здесь $u_э$, $u_к$, $i_{эд}$, $i_{кд}$, $i_г$ — фазовые переменные, а остальные величины — параметры модели транзистора.

Механические системы. Фазовыми переменными в механических поступательных системах являются силы и скорости. Используют одну из двух возможных электромеханических аналогий. В дальнейшем будем использовать ту из них, в которой скорость относят к фазовым переменным типа потенциала, а силу считают фазовой переменной типа потока. Учитывая формальный характер подобных аналогий, в равной мере можно применять и противоположную терминологию.

Компонентное уравнение, характеризующее инерционные свойства тел, в силу второго закона Ньютона имеет вид

$$F = Mdu/dt, \quad (3.8)$$

где F — сила; M — масса; u — поступательная скорость.

Упругие свойства тел описываются компонентным уравнением, которое можно получить из уравнения закона Гука. В одномерном случае (если рассматриваются продольные деформации упругого стержня)

$$G = E\varepsilon, \quad (3.9)$$

где G — механическое напряжение; E — модуль упругости; $\varepsilon = \Delta l/l$ — относительная

деформация; Δl — изменение длины l упругого тела под воздействием G .

Учитывая, что $G = F/S$, где F — сила, S — площадь поперечного сечения тела, и дифференцируя (3.9), имеем

$$dF/dt = (SE/l)d(l)/dt$$

или

$$dF/dt = gu, \quad (3.10)$$

где $g = SE/l$ — жесткость, $u = d(l)/dt$ — скорость.

В механических вращательных системах справедливы компонентные и топологические уравнения поступательных систем с заменой поступательных скоростей на угловые, сил — на вращательные моменты, масс — на моменты инерции, жесткостей — на вращательные жесткости.

Условные обозначения простых элементов механической системы показаны на рис. 3.2, б.

Нетрудно заметить наличие аналогий между электрической и механической системами. Так, токам и напряжениям в первой из них соответствуют силы (либо моменты) и скорости механической системы, компонентным уравнениям (3.4) и (3.5) и фигурирующим в них параметрам C и L , — уравнения (3.8) и (3.10) и параметры M и L_M очевидна аналогия и между топологическими уравнениями.

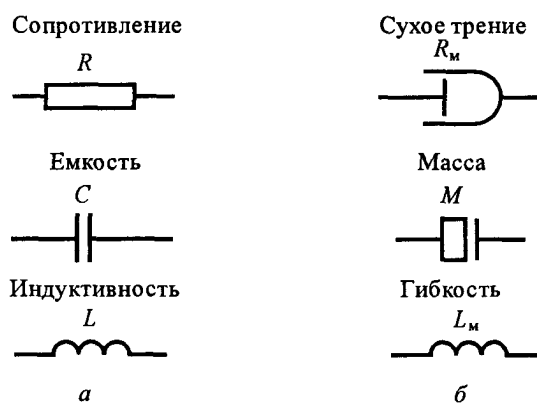


Рис. 3.2. Условные обозначения простых элементов в эквивалентных схемах: а — электрических, гидравлических, тепловых; б — механических

Представление топологических уравнений

Известен ряд методов формирования ММС на макроуровне. Получаемые с их помощью модели различаются ориентацией на те или иные численные методы решения и набором *базисных переменных*, т. е. фазовых переменных, остающихся в уравнениях итоговой ММС. Общей для всех методов является исходная совокупность топологических и компонентных уравнений (3.1) и (3.2).

При записи топологических уравнений удобно использовать промежуточную графическую форму — представление модели в виде эквивалентной схемы, состоящей из двухполюсных элементов. Общность подхода при этом сохраняется, так как любой многополюсный компонент можно заменить подсхемой из двухполюсников. В свою очередь, эквивалентную схему можно рассматривать как направленный граф, дуги которого соответствуют ветвям схемы. Направления потоков в ветвях выбираются произвольно (если реальное направление при моделировании окажется противоположным, то это приведет лишь к отрицательным численным значениям потока).

Пример некоторой простой эквивалентной схемы и соответствующего ей графа приведен на рис. 3.6. Для конкретности и простоты изложения на рисунке использованы условные обозначения, характерные для электрических эквивалентных схем, по той же причине далее в этом параграфе часто применяется электрическая терминология. Очевидно, что поясненные выше аналогии позволяют при необходимости легко перейти к обозначениям и терминам, привычным для механиков.

Для получения топологических уравнений все ветви эквивалентной схемы разделяют на подмножества *хорд* и *ветвей дерева*. Имеется в виду покрывающее (фундаментальное) дерево, т. е. подмножество из $B - 1$ дуг, не образующее ни одного замкнутого контура, где B — число вершин графа (узлов эквивалентной схемы). На рис. 3.6, б показан граф эквивалентной схемы, приведенной на рис. 3.6, а, утолщенными линиями выделено одно из возможных покрывающих деревьев.

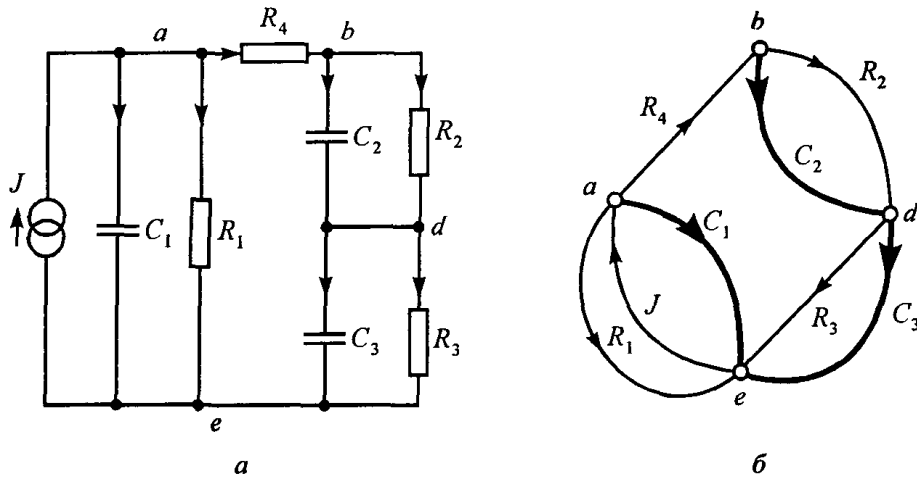


Рис. 3.6. Эквивалентная схема (а) и ее граф (б)

Выбор дерева однозначно определяет векторы напряжений U_x и токов I_x хорд, напряжений $U_{вд}$ и токов $I_{вд}$ ветвей дерева и приводит к записи топологических уравнений в виде

$$U_x + M U_{вд} = 0; \quad (3.13)$$

$$I_{вд} - M^T I_x = 0, \quad (3.14)$$

где M — матрица контуров и сечений; M^T — транспонированная M -матрица.

В M -матрице число строк соответствует числу хорд, число столбцов равно числу ветвей дерева. M -матрица формируется следующим образом. Поочередно к дереву подключаются хорды. Если при подключении к дереву p -й хорды q -я ветвь входит в образовавшийся контур, то элемент матрицы $M_{pq} = +1$ при совпадении направлений ветви и подключенной хорды, $M_{pq} = -1$ при несовпадении направлений. В противном случае $M_{pq} = 0$.

Для схемы на рис. 3.6 M -матрица представлена в виде табл. 3.1

Таблиц а 3. 1

Хорды	Ветви дерева		
	$C1$	$C2$	$C3$
$R1$	-1	0	0
$R2$	0	-1	0
$R3$	0	0	-1
$R4$	-1	+1	+1
J	+1	0	0

Лекция 6

МОДЕЛИРОВАНИЕ И АНАЛИЗ АНАЛОГОВЫХ УСТРОЙСТВ

На функционально-логическом уровне исследуют устройства, в качестве элементов которых принимают достаточно сложные узлы и блоки, считавшиеся системами на макроуровне. Поэтому необходимо упростить представление моделей этих узлов и блоков по сравнению с их представлением на макроуровне. Другими словами, вместо полных моделей узлов и блоков нужно использовать их макромоделли.

Вместо двух типов фазовых переменных в моделях функционально-логического уровня фигурируют переменные одного типа, называемые сигналами.

Физический смысл сигнала, т. е. его отнесение к фазовым переменным типа потока или типа потенциала, конкретизируют в каждом случае исходя из особенностей задачи.

Основой моделирования аналоговых устройств на функционально-логическом уровне является использование аппарата передаточных функций. При этом модель каждого элемента представляют в виде уравнения вход-выход, т. е. в виде

$$\mathbf{V}_{\text{вых}} = f(\mathbf{V}_{\text{вх}})$$

где $\mathbf{V}_{\text{вых}}$ и $\mathbf{V}_{\text{вх}}$ - сигналы на выходе и входе узла соответственно. Если узел имеет более чем один вход и один выход, то скаляры $\mathbf{V}_{\text{вых}}$ и $\mathbf{V}_{\text{вх}}$ становятся векторами. Однако известно, что подобное представление модели возможно, только если узел является безынерционным, т. е. в полной модели узла не фигурируют производные. Следовательно, для получения данного выражения в общем случае требуется предварительная алгебраизация полной модели. Такую алгебраизацию выполняют с помощью интегральных преобразований, например с помощью преобразования Лапласа, переходя из временной области в пространство комплексной переменной p . Тогда в моделях такого типа имеют место не оригиналы, а изображения сигналов $\mathbf{V}_{\text{вых}}(p)$ и $\mathbf{V}_{\text{вх}}(p)$, сами же модели реальных блоков стараются по возможности максимально упростить и представить моделями типовых блоков (звеньев) из числа заранее разработанных библиотечных моделей. Обычно модели звеньев имеют вид

$$\mathbf{V}_{\text{вых}}(p) = \mathbf{h}(p) \mathbf{V}_{\text{вх}}(p),$$

где $\mathbf{h}(p)$ - передаточная функция звена.

В случае применения преобразования Лапласа появляются ограничения на использование нелинейных моделей, а именно в моделях не должно быть нелинейных инерционных элементов.

Другое упрощающее допущение при моделировании на функционально-логическом уровне — неучет влияния нагрузки на характеристики блоков. Действительно, подключение к выходу блока некоторого другого узла не влияет на модель блока.

Собственно получение математической модели системы (ММС) из математических моделей элементов (ММЭ) оказывается вследствие принятых допущений значительно проще, чем на макроуровне: ММС есть совокупность ММЭ, в которых отождествлены сигналы на соединенных входах и выходах элементов. Эта ММС представляет собой систему алгебраических уравнений.

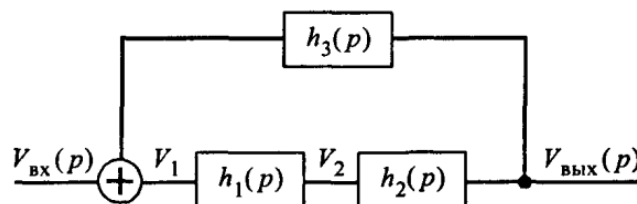
Получение ММС проиллюстрируем простым примером (см. рис.), где показана система из трех блоков с передаточными функциями $\mathbf{h}_1(p)$, $\mathbf{h}_2(p)$ и $\mathbf{h}_3(p)$. ММС имеет вид

$$\begin{aligned} V_2 &= h_1(p)V_1; \\ V_{\text{вых}} &= h_2(p)V_2; \\ V_1 &= V_{\text{вх}}(p) - h_3(p)V_{\text{вых}}; \end{aligned}$$

или

$$V_{\text{вых}} = H(p)V_{\text{вх}}(p);$$

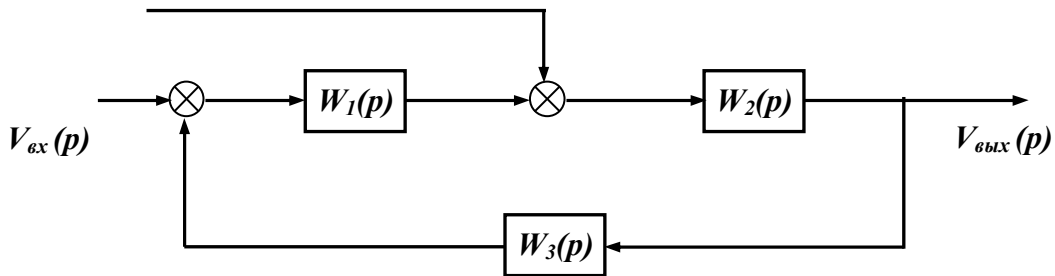
где $H(p) = h_1(p)h_2(p) / (1 - h_1(p)h_2(p)h_3(p))$



Пример схемы из трех блоков

Правила эквивалентных преобразований структурных схем систем автоматического управления

Любая система может быть представлена в виде совокупности динамических звеньев с известными математическими моделями. Например, рассмотрим структуру следующей системы:

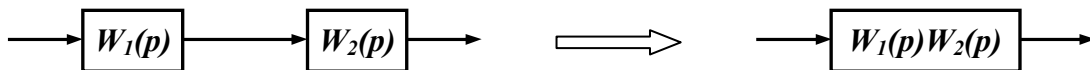


В процессе анализа и синтеза систем необходимо получать передаточные функции, которые связывают выходную переменную $V_{вых}(p)$ со входной $V_{вх}(p)$ и возмущением по известным структурной схеме и передаточным функциям динамических звеньев.

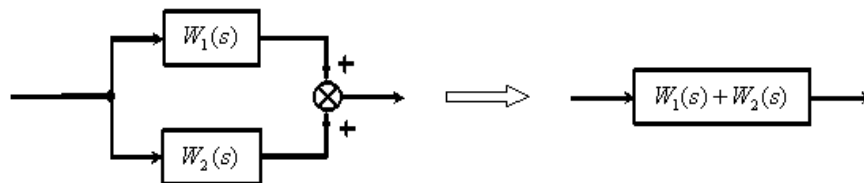
Эта задача решается путем преобразования (сворачивания) структурной схемы к одному динамическому звену с искомой передаточной функцией на основе использования правил эквивалентных преобразований структурных схем и принципа суперпозиции (наложения).

Рассмотрим правила эквивалентных преобразований, не изменяющих свойств систем и необходимых для нахождения передаточной функции:

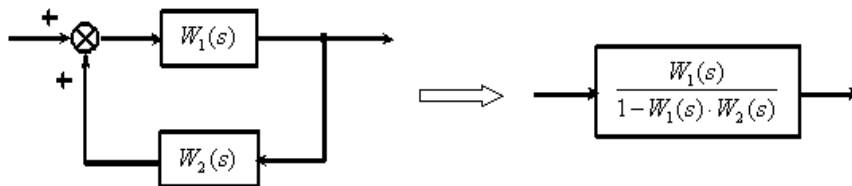
1. Последовательное соединение динамических звеньев.



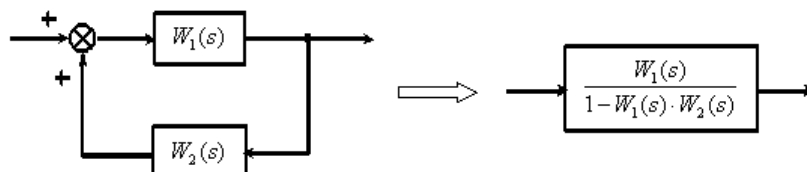
2. Параллельное соединение динамических звеньев.



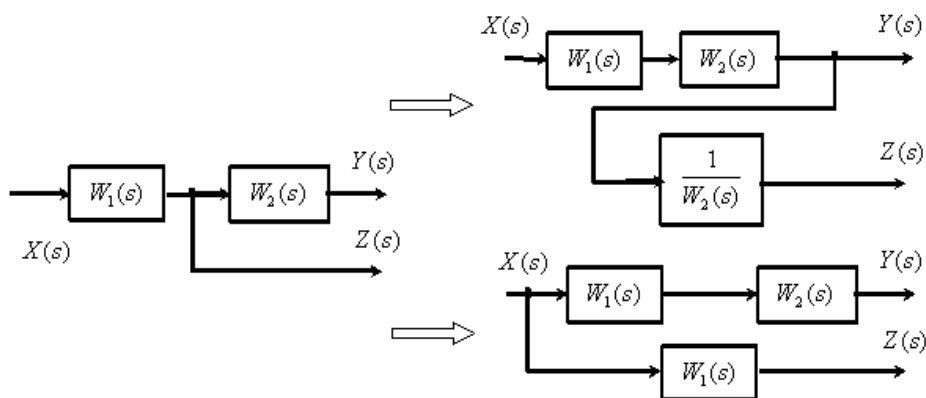
3. Замкнутый контур с отрицательной обратной связью.



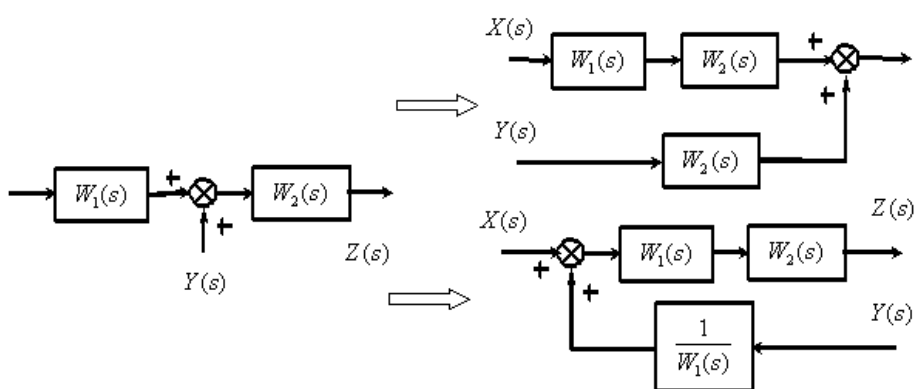
4. Замкнутый контур с положительной обратной связью.



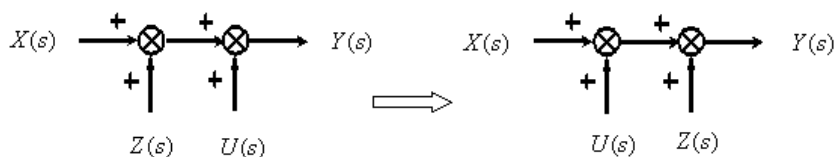
5. Перенос точки ветвления через динамическое звено.



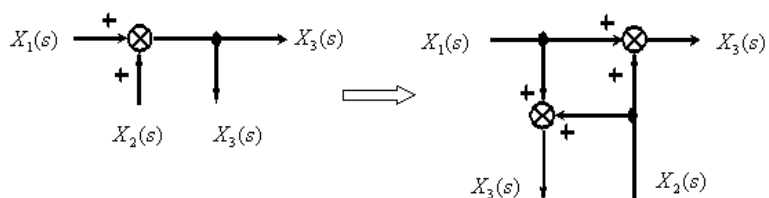
6. Перенос суммирующего звена через динамическое звено.



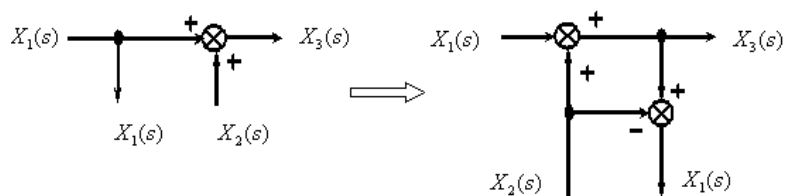
7. Перестановка суммирующих звеньев.



8. Перенос точки ветвления с выхода на вход суммирующего звена

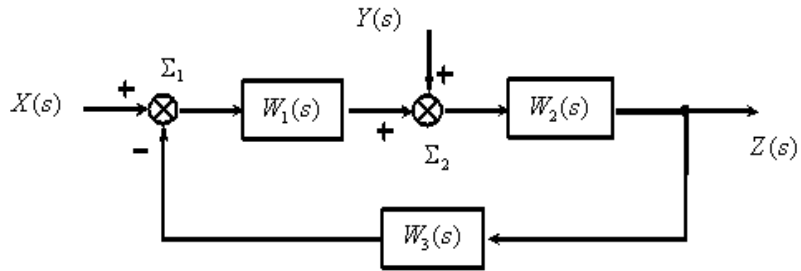


9. Перенос точки ветвления с входа на выход суммирующего звена.



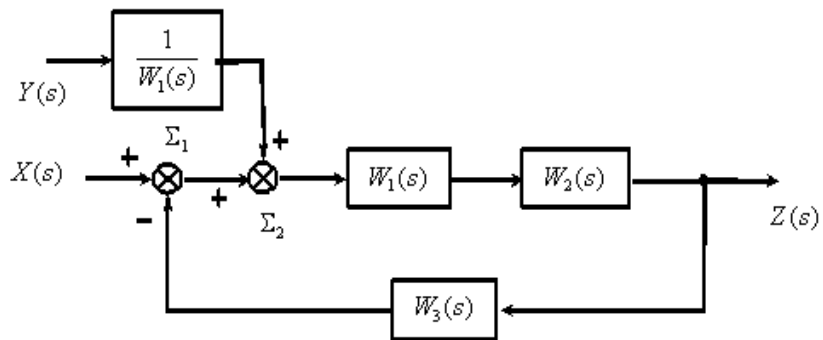
Лекция 7

Применим рассмотренные правила для упрощения структурной схемы, что представлена выше:

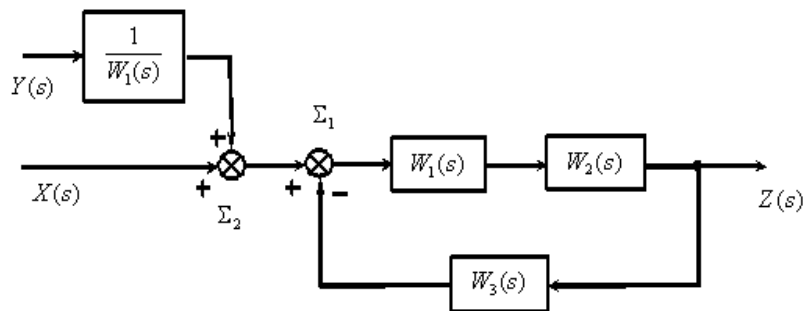


Процесс преобразования, который часто называют свертыванием структурной схемы, выглядит следующим образом.

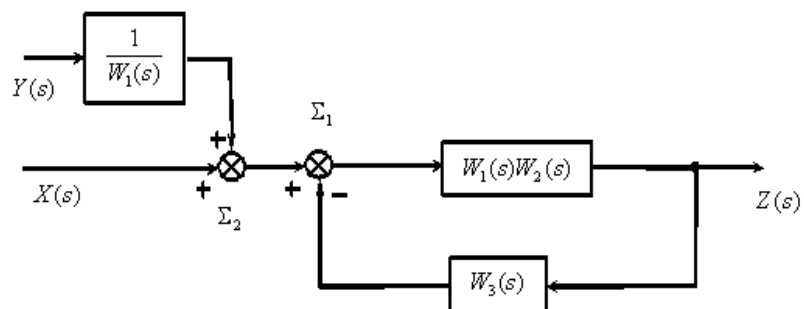
1. Перенесем суммирующее звено Σ_2 через динамическое звено $W_1(s)$:



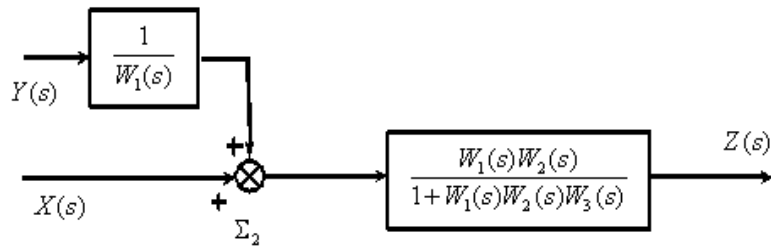
2. Поменяем местами суммирующие звенья Σ_1 и Σ_2 :



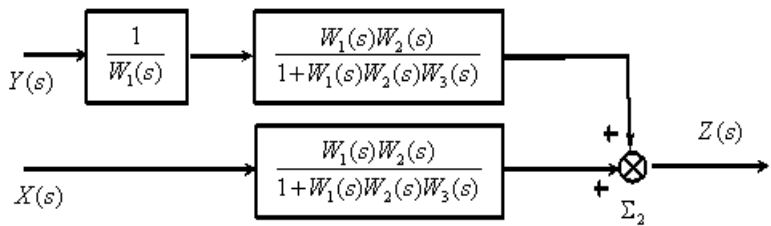
3. Преобразуем последовательно включенные динамические звенья $W_1(s)$ и $W_2(s)$:



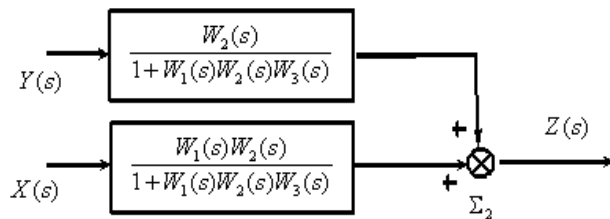
4. Преобразуем замкнутый контур с отрицательной обратной связью $\Sigma_1, W_1(s), W_2(s)$ и $W_3(s)$:



5. Перенесем суммирующее звено Σ_2 вправо :



6. Преобразуем последовательно включенные звенья :



В соответствии с полученной структурной схемой запишем операторное уравнение –

$$Z(s) = X(s) \frac{W_1(s)W_2(s)}{1+W_1(s)W_2(s)W_3(s)} + Y(s) \frac{W_2(s)}{1+W_1(s)W_2(s)W_3(s)} =$$

$$= X(s)W_{zx}(s) + Y(s)W_{zy}(s).$$

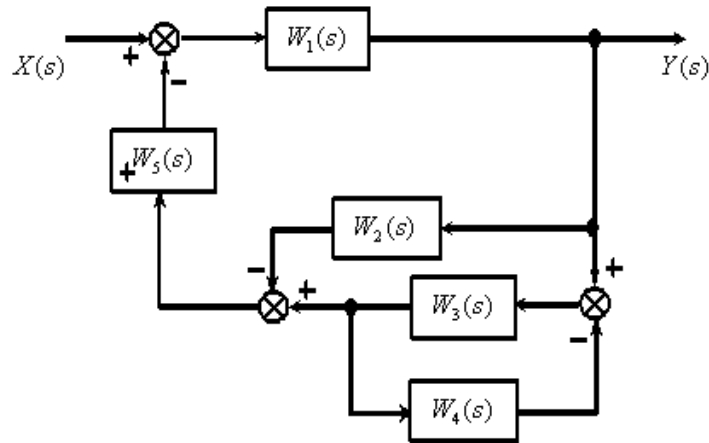
Полученное уравнение показывает, что $Z(s)$ является линейной комбинацией изображений входных сигналов, взятых с коэффициентами $W_{zx}(s)$ и $W_{zy}(s)$.

Теперь становится ясным смысл и самого операторного уравнения, описывающего систему. Он заключается в том, что реакция линейной системы на совместно действующие входные сигналы может быть определена в виде суммы частичных реакций, каждая из которых вычисляется в предположении, что на систему действует только один входной сигнал, а остальные равны нулю.

По сути – это формулировка фундаментального принципа, который называют принципом наложения или суперпозиции. Этот принцип можно рассматривать как дополнение к правилам эквивалентных преобразований структурных схем и активно использовать на практике.

Практически принцип суперпозиции для нахождения конкретной передаточной функции используют следующим образом. Полагают равными нулю все входные сигналы, кроме необходимого сигнала, а затем выполняют преобразование структурной схемы в одно динамическое звено.

Задание 1. Определите передаточную функцию, эквивалентную структурной схеме.



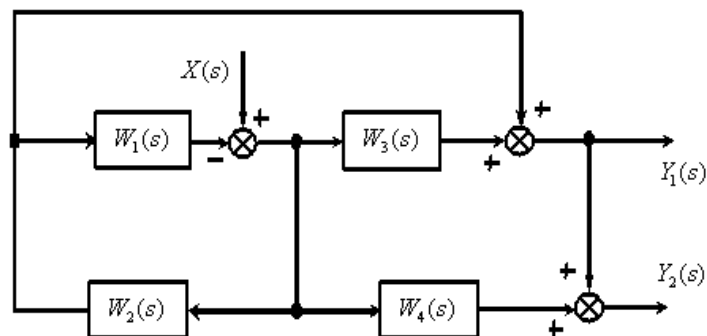
Ответ:

$$W(s) = \frac{W_1(s)(1+W_3(s)W_4(s))}{1+W_3(s)W_4(s)+W_1(s)W_5(s)(W_3(s)-W_2(s)(1+W_3(s)W_4(s)))}$$

Задание 2. Определите передаточные функции

$$W'(s) = \frac{Y_1(s)}{X(s)}, W''(s) = \frac{Y_2(s)}{X(s)},$$

по следующей структурной схеме:



Ответ:

$$W'(s) = \frac{W_2(s)+W_3(s)}{1+W_1(s)W_2(s)}, \quad W''(s) = \frac{W_2(s)+W_3(s)+W_4(s)}{1+W_1(s)W_2(s)}$$

Лекция 8

МАТЕМАТИЧЕСКИЕ МОДЕЛИ ДИСКРЕТНЫХ УСТРОЙСТВ.

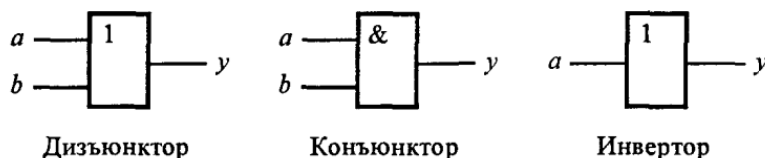
Анализ дискретных устройств на функционально-логическом уровне требуется прежде всего при проектировании устройств вычислительной техники и цифровой автоматики. Здесь дополнительно к допущениям, принимаемым при анализе аналоговых устройств, используют дискретизацию сигналов, причем базовым является двузначное представление сигналов. Удобно этими двумя возможными значениями сигналов считать «истину» (иначе 1) и «ложь» (иначе 0), а сами сигналы рассматривать как булевы величины. Тогда для моделирования можно использовать аппарат математической логики. Находят применение также трех- и более значные модели. Смысл значений сигналов в многозначном моделировании и причины его применения будут пояснены далее на некоторых примерах.

Элементами цифровых устройств на функционально-логическом уровне служат элементы, выполняющие логические функции и возможно функции хранения информации. Простейшими элементами являются дизъюнктор, конъюнктор, инвертор, реализующие соответственно операции:

- дизъюнкции (ИЛИ) $y = a \text{ or } b$,
- конъюнкции (И) $y = a \text{ and } b$,
- отрицания (НЕ) $y = \text{not } a$,

где y — выходной сигнал, a и b — входные сигналы.

Число входов может быть и более двух. Условные схемные обозначения простых логических элементов показаны ниже:



Дизъюнктор

Конъюнктор

Инвертор

Математические модели устройств представляют собой систему математических моделей элементов, входящих в устройство, при отождествлении сигналов, относящихся к одному и тому же соединению элементов.

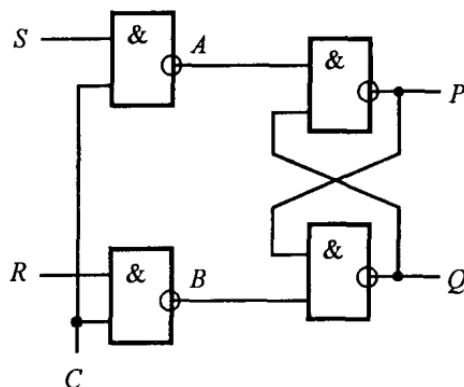


Схема RS-триггера.

Различают **синхронные** и **асинхронные** модели.

Синхронная модель представляет собой систему логических уравнений, в ней отсутствует такая переменная, как время. Синхронные модели используют для анализа установившихся состояний.

Примером синхронной модели может служить следующая система уравнений, полученная для логической схемы триггера:

- $B = \text{not } (R \text{ and } C)$;
- $Q = \text{not } (B \text{ and } P)$;
- $P = \text{not } (A \text{ and } Q)$;
- $A = \text{not } (S \text{ and } C)$.

Асинхронные модели отражают не только логические функции, но и временные задержки в распространении сигналов. Асинхронная модель логического элемента имеет вид

$$y(t + t_{30}) = f(X(t))$$

где t_{30} — задержка сигнала в элементе; f — логическая функция.

Термины «синхронная» и «асинхронная модель» можно объяснить ориентированностью этих моделей на синхронные и асинхронные схемы соответственно.

В синхронных схемах передача сигналов между цифровыми блоками происходит только при подаче на специальные синхровходы тактовых (синхронизирующих) импульсов. Частота тактовых импульсов выбирается такой, чтобы к моменту прихода синхроимпульса переходные процессы от предыдущих передач сигналов фактически закончились. Следовательно, в синхронных схемах расчет задержек не актуален, быстродействие устройства определяется заданием тактовой частоты.

Синхронные модели можно использовать не только для выявления принципиальных ошибок в схемной реализации заданных функций. С их помощью можно обнаруживать места в схемах, опасные с точки зрения возникновения в них искажающих помех. Ситуации, связанные с потенциальной опасностью возникновения помех и сбоев, называют *рисками сбоя*.

Метод минимизации по картам Карно

Данный метод минимизации применим для функций с числом переменных не более 6 и удобен для ручной минимизации, когда человек видит те комбинации, которые можно объединить вместе. Рассмотрим его на конкретном примере.

Пример 1. Рассмотрим функцию

$$f_1(x_1, x_2, x_3) = x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3$$

Множество переменных разобьем на две группы. Одной группе сопоставим строки таблицы, второй — столбцы, так чтобы каждой клетке соответствовала комбинация переменных из этих групп. Карта Карно для нее имеет вид:

$x_1 x_2 / x_3$	0	1
00		
01		1 ₅
11	1 ₂	1 ₁
10	1 ₄	1 ₃

При составлении карты Карно строки именовываются всевозможными комбинациями значений переменных первой группы так, чтобы расстояние между соседними комбинациями было равно 1. Для нашего случая 00@ 01@ 11@ 10 (при каждом последующем переходе изменяется только подчеркнутый символ). Аналогично именовываются столбцы таблицы.

Заполнение карты производится по таблице соответствия исходной функции. В примере конъюнкции $x_1 x_2 x_3$ соответствует клетка 11/1, а клетка 11/0 и так далее. В данной таблице каждая единица имеет порядковый индекс, который соответствует порядковому номеру данной компоненты в исходной функции (расстановка этих индексов совершенно не обязательна и здесь приведена для лучшего понимания).

Для минимизации необходимо попарно «склеить» рядом стоящие единицы, имеющие хотя бы одну общую компоненту. При этом надо стремиться «склеить» в один набор как можно больше клеток.

В данном примере мы можем “склеить” **11,12,13,14** вместе. Это запишется как **x1**, так как содержимое всех этих клеток зависит только от **x1** и не меняется при изменении **x2** или **x3**. На следующем шаге склеим **11** и **15**. В результате получим **x2x3**. Рассуждения аналогичны: при изменении **x1** изменения ячеек с **11** и **15** не происходит.

Резльтирующей минимальной записью исходной функции будет

$$f(x_1, x_2, x_3) = x_1 \vee x_2 x_3$$

Пример 2. Минимизируем функцию пяти переменных:

$$f_2(x_1, x_2, x_3, x_4, x_5) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee \bar{x}_2 x_1 \bar{x}_5$$

$x_4 x_5 x_1 x_2 x_3$	000	001	011	010	110	111	101	100
00							1 ₄	1 _{1,4}
01								1 ₁
11			1 ₃					1 ₂
10			1 ₃				1 ₄	1 _{2,4}

Лекция 9

Определение абстрактного автомата

Математической моделью дискретного управляющего устройства является абстрактный автомат, который задается множеством из шести элементов:

$$S = \{A, Z, W, a_t, w_t, a_1\},$$

где

$A = \{a_1, \dots, a_m, \dots, a_M\}$ - множество состояний (алфавит состояний);

$Z = \{z_1, \dots, z_f, \dots, z_F\}$ - множество входных сигналов (входной алфавит);

$W = \{w_1, \dots, w_g, \dots, w_G\}$ - множество выходных сигналов (выходной алфавит);

a_t - функция переходов, реализующая отображение множества $\delta_{A \times Z}$ в A ($a_s = (a_m, z_f), a_s \in A$);

w_t - функция выходов, реализующая отображение множества $\delta_{A \times Z}$ на W ($w_g = (a_m, z_f)$);

a_1 - начальное состояние автомата.

Автомат называется конечным, если конечны множества A, Z, W . В дальнейшем будут рассматриваться только конечные автоматы и термин "конечный" почти всегда будет опускаться. Автомат называется полностью определенным, если $\delta_A = \delta_W = A \times Z$. Иными словами, у полностью определенного автомата области определения функции и совпадают со множеством $A \times Z$ - множеством всевозможных: пар вида (a_m, z_f) . У частичного автомата функции или определены не для всех пар $(a_m, z_f) \in A \times Z$.

Понятие состояния в определении автомата введено в связи с тем, что: возникает необходимость в описании поведения систем, выходы которых зависят не только от состояния входов в данный момент времени, но от некоторой предыстории, т. е. от сигналов, которые поступали на входы системы ранее. Состояния как раз и соответствуют некоторой памяти о прошлом позволяя устранить время как явную переменную и выразить выходные сигналы, как функцию состояний и входов в данный момент времени.

Абстрактный автомат (рис. 1) имеет один входной и один выходной канал. В каждый момент $t = 0, 1, 2, \dots$ дискретного времени автомат находится в определенном состоянии $a(t)$ из множества A состояний автомата, причем в начальный момент $t=0$ он всегда находится в начальном состоянии $a(0) = a_1$. В момент t , будучи в состоянии $a(t)$, автомат способен воспринять на входном канале сигнал $z(t) \in Z$ и выдать на выходном канале сигнал $w(t) = (a(t), z(t))$, переходя в состояние $a(t+1) = (a(t), z(t))$; $a(t) \in A, w(t) \in W$. Смысл понятия абстрактного автомата состоит в том, что он реализует некоторое отображение множества слов входного алфавита Z во множество слов выходного алфавита W . Другими словами, если на вход автомата, установленного в начальное состояние a_1 , подавать буква за буквой некоторую последовательность букв входного алфавита $z(0), z(1), z(2), \dots$ - входное слово, то на выходе автомата будут последовательно появляться буквы выходного алфавита $w(0), w(1), w(2), \dots$ - выходное слово. Относя к каждому входному слову соответствующее ему выходное слово, мы получим отображение, индуцированное абстрактным автоматом.

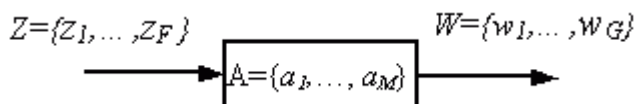


Рис. 1

Автоматы Мили и Мура

На практике наибольшее распространение получили два типа конечных автоматов: автомат Мили и автомат Мура. Поведение *автомата Мили* описывается с помощью следующих

$$\begin{aligned} a_{t+1} &= (z_t, a_t); \\ w_t &= (z_t, a_t); \end{aligned} \quad \text{где } t = 0, 1, 2, \dots \quad (1)$$

здесь функция переходов a_{t+1} определяет следующее внутреннее состояние автомата (состояние перехода, а функция выходов w_t - формируемый выходной набор. Характерным для автомата Мили является то, что выходной набор w_t зависит как от входного набора z_t , так и от внутреннего состояния a_t .

В *автомате Мура* функция выходов зависит только от состояния a_t и не зависит от входного набора, поэтому поведение автомата Мура описывается следующими уравнениями:

$$\begin{aligned} a_{t+1} &= (z_t, a_t); \\ w_t &= (a_t); \end{aligned} \quad \text{где } t = 0, 1, 2, \dots \quad (2)$$

При построении конечных автоматов функции переходов и выходов реализуются с помощью комбинационных схем CL_Φ и CL_Ψ , соответственно, а память автомата реализуется в виде регистра RG, где в каждый момент автоматного времени t хранится код внутреннего состояния a_t . Структурные схемы автоматов Мили и Мура показаны на рис. 1а и 1б, соответственно. Особенностью структуры автомата Мура является то, что комбинационная схема CL_Ψ , реализующая выходные функции, не имеет непосредственной связи со входами схемы.

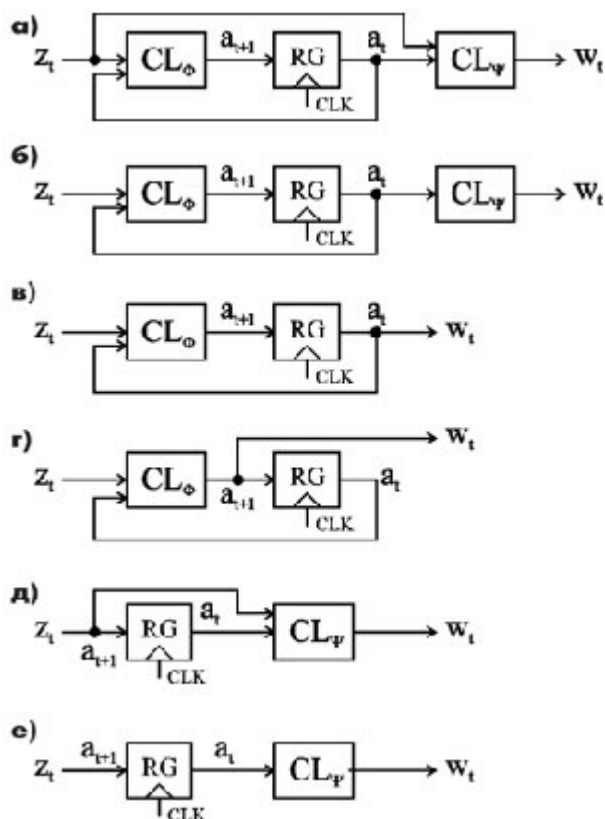


Рис. 2. Основные структуры конечных автоматов:

- | | |
|--------------------|--------------------|
| а - Мили класса А; | б - Мура класса В; |
| в - Мура класса С; | г - Мили класса D; |
| д - Мили класса Е; | е - Мура класса F; |

Методы задания автоматов

Чтобы задать конечный автомат S , необходимо описать все элементы множества $S = \{A, Z, W, a_t, w_t, a_1\}$, т. е. входной и выходной алфавиты и алфавит состояний, а также функции переходов и выходов. Среди множества состояний необходимо выделить состояние a_1 , в котором автомат находится в момент $t = 0$. Существует несколько способов задания работы автомата, но наиболее часто используются табличный, графический и матричный.

Табличный метод

Описание работы автомата Мили таблицами переходов и выходов иллюстрируется таблицами Табл. 1 и Табл. 2. Строки этих таблиц соответствуют входным сигналам, а столбцы - состояниям, причем крайний левый столбец состояний обозначен начальным состоянием a_1 .

Табл. 1
Общий вид таблицы переходов автомата Мили

	a_1	...	a_m
Z_1	$\delta(a_1, z_1)$...	$\delta(a_m, z_1)$
...
Z_F	$\delta(a_1, z_F)$...	$\delta(a_m, z_F)$

Табл. 2
Общий вид таблицы выходов автомата Мили

	a_1	...	a_m
Z_1	$\lambda(a_1, z_1)$...	$\lambda(a_m, z_1)$
...
Z_F	$\lambda(a_1, z_F)$...	$\lambda(a_m, z_F)$

На пересечении столбца a_m и строки z_f в таблице переходов ставится состояние $a_s = (a_m, z_f)$, в которое автомат переходит из состояния a_m под действием сигнала z_f , а в таблице выходов-соответствующий этому переходу выходной сигнал $w_g = (a_m, z_f)$. Пример табличного способа задания полностью определенного автомата Мили S_1 с тремя состояниями, двумя входными и двумя выходными сигналами приведен в таблицах Табл. 3 и Табл. 4.

Табл. 3
Таблица переходов автомата Мили S_1

	a_1	a_2	a_3
Z_1	a_3	a_1	a_1
Z_2	a_1	a_3	a_2

Табл. 4.
Таблица выходов автомата Мили S_1

	a_1	a_2	a_3
Z_1	w_1	w_1	w_2
Z_2	w_1	w_2	w_1

Для частичных автоматов, у которых функции δ или λ определены не для всех пар (a_m, z_f) $A \times Z$, на месте неопределенных состояний и выходных сигналов ставится прочерк (частичный автомат S_2 задан таблицами Табл. 5 и Табл. 6).

Табл. 5
Таблица переходов частичного автомата Мили S_2

	a_1	a_2	a_3	a_4
Z_1	a_2	a_3	a_4	-
Z_2	a_3	-	a_2	a_2

Табл. 6
Таблица выходов частичного автомата Мили S_2

	a_1	a_2	a_3	a_4
Z_1	w_1	w_3	w_3	-
Z_2	w_2	-	w_1	w_2

Так как в автомате Мура выходной сигнал зависит, только от состояния, автомат Мура задается одной отмеченной таблицей переходов (Табл. 7), в которой каждому ее столбцу

приписан, кроме состояния a_m , еще и выходной сигнал $w_g = (a_m)$, соответствующий этому состоянию. Пример табличного описания автомата Мура S_3 иллюстрируется таблицей Табл. 8.

Табл. 7
Общий вид отмеченной таблицы автомата Мура S_2

	$\lambda(a_1)$...	$\lambda(a_M)$
	a_1	...	a_M
Z_1	$\delta(a_1, z_1)$...	$\delta(a_M, z_1)$
...
Z_F	$\delta(a_1, z_F)$...	$\delta(a_M, z_F)$

Табл. 8
Отмеченная таблица переходов автомата Мура S_2

	w_1	w_1	w_3	w_2	w_3
	a_1	a_2	a_3	a_4	a_5
Z_1	a_2	a_5	a_5	a_3	a_3
Z_2	a_4	a_2	a_2	a_1	a_1

Графический метод

Граф автомата - ориентированный связный граф, вершины которого соответствуют состояниям, а дуги - переходам между ними.

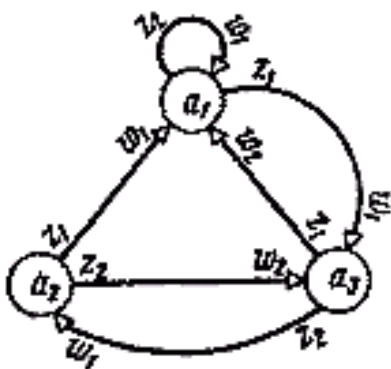


Рис. 3 Граф автомата Мили S_1

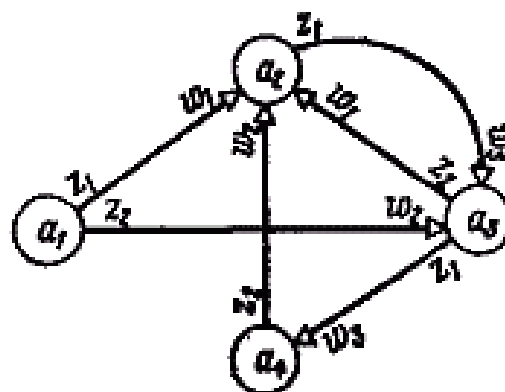


Рис. 4 Граф автомата Мили S_2

Две вершины графа автомата a_m и a_s (исходное состояние и состояние перехода) соединяются дугой, направленной от a_m к a_s , если в автомате имеется переход из a_m в a_s , т. е. если $a_s = (a_m, z_f)$ при некотором $z_f \in Z$. Дуге (a_m, a_s) графа автомата приписывается входной сигнал z_f и выходной сигнал $w_g = (a_m, z_f)$, если он определен, и ставится прочерк в противном случае. Если переход автомата из состояния a_m в состояние a_s происходит под действием нескольких входных сигналов, то дуге $(a_m - a_s)$ приписываются все эти входные и соответствующие выходные сигналы. При описании автомата Мура в виде графа выходной сигнал $w_g = (a_m)$ записывается внутри вершины или рядом с ней. На рис. 3, 4 и 5 приведены заданные ранее таблицами графы автоматов S_1, S_2, S_3 .

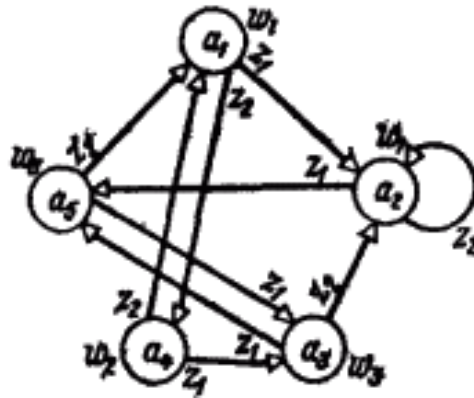


Рис. 5. Граф автомата Мили S_3

В данной работе рассматриваются только детерминированные автоматы, у которых выполнено условие однозначности переходов: автомат, находящийся в некотором состоянии, под действием любого входного сигнала не может перейти более чем в одно состояние. Применительно к графическому способу задания автомата это означает, что в графе автомата из любой вершины не могут выходить две и более дуги, отмеченные одним и тем же входным сигналом. Аналогично этому в матрице соединений автомата в каждой строке любой входной сигнал не должен встречаться более одного раза.

Табл. 9
Отмеченная таблица переходов асинхронного автомата Мура S_4

	w_1	w_3	w_2
	a_1	a_2	a_3
z_1	a_2	a_2	a_2
z_2	a_3	a_2	a_2
z_3	a_1	a_1	a_1

В заключение данного параграфа определим синхронные и асинхронные автоматы. Состояние a_s автомата S называется устойчивым состоянием, если для любого входа $z_f \in Z$, имеет место $(a_m, z_f) = a_s$.

Автомат S называется асинхронным, если каждое его состояние $a_s \in A$ устойчиво. Автомат S называется синхронным, если он не является асинхронным. Необходимо заметить, что построенные на практике автоматы - всегда асинхронные и устойчивость их состояний всегда обеспечивается тем или иным способом, например введением сигналов синхронизации.

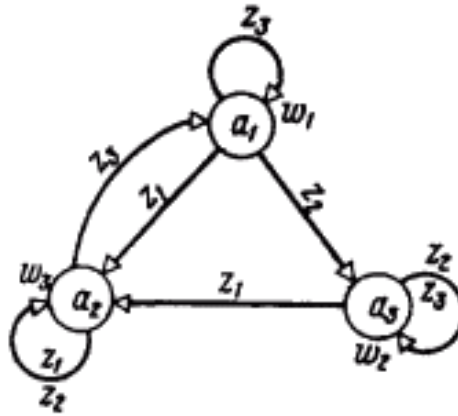


Рис. 6. Граф асинхронного автомата Мура S4

Однако на уровне абстрактной теории, когда автомат есть лишь математическая модель, которая не отражает многих конкретных особенностей его возможной реализации, часто оказывается более удобным оперировать с синхронными автоматами, что мы и будем делать на протяжении всей данной главы. Пример асинхронного автомата Мура S4 приведен в Табл. 9 и на Рис. 6. Нетрудно видеть, что все его состояния устойчивы.

Очевидно, что если в таблице переходов асинхронного автомата некоторое состояние a_s стоит на пересечении строки z_f и столбца a_m ($m < s$), то это состояние a_s обязательно должно встретиться в этой же строке в столбце a_s . В графе асинхронного автомата, если в некоторое состояние имеются переходы из других состояний под действием каких-то сигналов, то в вершине a_s , должна быть петля, отмеченная символами тех же входных сигналов. Анализ таблиц 3, 5 и 8 или рис. 3 - 5 показывает, что автоматы S1, S2 и S3 являются синхронными.

Связь между моделями Мили и Мура

Абстрактный автомат работает как преобразователь слов входного алфавита в слова в выходном алфавите. Остановимся на этом вопросе более подробно, взяв в качестве примера автомат Мили S1 на рис. 3 (или табл. 3, 4).

Пусть на вход этого автомата, установленного в начальное состояние, поступает входное слово = $z_1z_1z_2z_1z_2z_2$. Так как $(a_1, z_1) = a_3$, а $(a_1, z_1) = w_1$, то под действием первой буквы слова - входного сигнала z_1 автомат перейдет в состояние a_3 и на выходе его появится сигнал w_1 .

Далее, $(a_3, z_1) = a_1$, $(a_3, z_1) = w_2$ и потому при приходе второго сигнала z_1 автомат окажется в состоянии a_1 , а на выходе его появится сигнал w_2 .

Проследив непосредственно по графу или таблицам переходов и выходов дальнейшее поведение автомата, опишем его тремя строчками, первая из которых соответствует входному слову вторая - последовательности состояний, которые проходит автомат под действием букв слова, третья - выходному слову W, которое появляется на выходе автомата:

z_1	z_1	z_2	z_1	z_2	z_2
a_3	a_1	a_1	a_3	a_2	a_3
w_1	w_2	w_1	w_1	w_1	w_2

Лекция 10

Учет значений параметров при моделировании

Моделирование заключается не только в формировании математической модели, но и в проведении с нею серии экспериментов, обработка результатов которых и является конечной целью математического моделирования. Именно по результатам судят, насколько удачным и прогрессивным является предложенное решение, насколько оно конкурентоспособно, позволит ли решить поставленные задачи. Понятно, что чем больше получено результатов, тем более обоснованно будет принятое решение. С другой стороны, это потребует проведения большего количества экспериментов, что выразится в большем времени, занимаемым моделированием.

Особенностью компьютерного моделирования является гибкая и простая форма задания параметров модели для каждого этапа моделирования, а также достаточно слабая зависимость времени моделирования от конкретных значений параметров. Последнее объясняется тем, что математическая модель почти всегда в компьютере воплощается в виде того или иного программного продукта, а время выполнения программы в большей степени зависит от ее алгоритма, чем от значений переменных. Для простоты в дальнейших рассуждениях будем считать время одного "прогона" программы для одного какого либо набора параметров модели постоянной величиной в рамках данной математической модели.

С учетом данного допущения становится очевидным, что время моделирования $T_{\text{мод}}$ определяется:

$$T_{\text{мод}} = Q \cdot T$$

где T – время одного запуска программы с математической моделью;

Q – количество вариантов параметров, для которых нужно выполнить моделирование.

Считаем, что время T определяется быстродействием компьютера, качеством написания и сложностью программы, особенностями языков программирования и т.д. Эти характеристики мы в данном разделе рассматривать не будем, поскольку они могут быть оговорены и оптимизированы уже на этапе моделирования. Целью нашего исследования является значение Q .

Пусть у модели имеется m параметров, каждый из которых оказывает влияние на получаемый при моделировании результат. Обозначим эти параметры p_1, p_2, \dots, p_m .

Теперь разберемся со значениями этих параметров. Принципы математического моделирования с помощью компьютерной программы подразумевают, что для каждого параметра определяется дискретный ряд значений, которые поочередно должен принимать данный параметр при моделировании. Очевидно, что на длительность моделирования влияют не конкретные значения данного параметра, а их количество. Обозначим для каждого из m параметров количество этих значений как q_1, q_2, \dots, q_m . Тогда Q может быть определено по формуле:

$$Q = q_1 \cdot q_2 \cdot q_3 \cdot \dots \cdot q_m$$

или

$$Q = \sum_{i=1}^m q_i$$

Отсюда становится очевидным: чтобы ускорить моделирование нужно критически пересмотреть параметры математической модели, убрав те, которые не оказывают на результат существенного влияния, а у оставшихся оптимизировать количество значений путем их уменьшения. Конечно, и первое, и второе не должно осуществляться за счет ухудшения качества математической модели, поэтому часть параметров с рядом своих значений будут оставлены при любых условиях. Теперь рассмотрим некоторые приемы уменьшения количества параметров без ухудшения качества модели.

Приемы уменьшения количества параметров.

Основная идея этого преобразования заключается в том, чтобы исключить такие сочетания параметров, которые бы дублировали уже смоделированные варианты. Например, если при моделировании электронной схемы некая моделируемая величина пропорциональна отношению параметров, выраженному в виде дроби

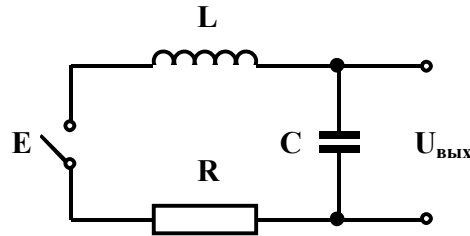
$$\frac{p_i}{p_j}$$

$$p_j$$

то, по видимому бессмысленно рассматривать как самостоятельные варианты случаи, когда и параметр p_i и параметр p_j одновременно увеличиваются или уменьшаются в одинаковое число раз, так как значение дроби при этом остается неизменным. Аналогичным образом для каждого математического преобразования можно найти такие "дублирующие" пары значений входящих в него параметров.

Другой прием заключается в замене группы из нескольких исходных параметров новым, который интегрированно отражает влияние группы этих параметров. Рассмотрим такого рода преобразования на примере.

Пусть объектом математического моделирования является схема:



Уравнение, описывающее изменение напряжения на конденсаторе C в RLC последовательном контуре при включении внешнего источника напряжения E (переходной процесс) при нулевых начальных условиях выглядит следующим образом:

$$\frac{\partial^2 U_{\text{ВЫХ}}}{\partial t^2} + \frac{R}{L} \cdot \frac{\partial U_{\text{ВЫХ}}}{\partial t} + \frac{1}{LC} U_{\text{ВЫХ}} = \frac{1}{LC} E$$

Вывод формулы выполняем через представление комплексных сопротивлений R , C и L в операторном виде и применение закона Ома для последовательно соединенных элементов:

$$U_{\text{ВЫХ}} = i_c \cdot Z_c$$

$$i_c = \frac{E}{Z_L + Z_C + Z_R}$$

$$Z_L = pL \quad Z_C = \frac{1}{pC} \quad Z_R = R$$

$$U_{\text{ВЫХ}} = \frac{1}{pC} \cdot \frac{E}{R + pL + \frac{1}{pC}} = \frac{E}{p^2 LC + pRC + 1}$$

Теперь введем безразмерную величину времени τ , зависящую только от параметров контура:

$$t = RC\tau$$

тогда $\tau = \frac{t}{RC} \quad \partial\tau = \frac{\partial t}{RC} \quad \partial\tau^2 = \frac{\partial t^2}{R^2 C^2}$

и первоначальная формула преобразуется к виду:

$$\frac{1}{R^2 C^2} \cdot \frac{\partial^2 U_{\text{ВЫХ}}}{\partial \tau^2} + \frac{R}{L} \cdot \frac{1}{RC} \cdot \frac{\partial U_{\text{ВЫХ}}}{\partial \tau} + \frac{1}{LC} U_{\text{ВЫХ}} = \frac{1}{LC} E$$

или

$$\frac{\partial^2 U_{\text{ВЫХ}}}{\partial \tau^2} + \frac{R^2 C}{L} \cdot \frac{\partial U_{\text{ВЫХ}}}{\partial \tau} + \frac{R^2 C}{L} U_{\text{ВЫХ}} = \frac{R^2 C}{L} E$$

Теперь еще введем дополнительно параметр $\mu = \frac{R^2 C}{L}$ и окончательно получим:

$$\frac{\partial^2 U_{\text{ВЫХ}}}{\partial \tau^2} + \mu \left(\frac{\partial U_{\text{ВЫХ}}}{\partial \tau} + U_{\text{ВЫХ}} \right) = \mu E$$

Таким образом, вместо шести первоначальных параметров (E , R , L , C , t , $U_{\text{ВЫХ}}$) осталось четыре (E , μ , τ , $U_{\text{ВЫХ}}$), что позволило уменьшить количество вариантов параметров, а значит – время моделирования.

Зависимость результата от выбора математической модели

В случае, когда для одного и того же процесса имеется несколько математических моделей. Выбор модели может оказать существенное влияние на точность получения результата.

Рассмотрим это на примере определения суммы ряда вида:

$$S = 1 + e^{-a} + e^{-2a} + \dots + e^{-(N-1)a}$$

$$\text{где } a = 0,01 \\ N = 100.$$

При этом абсолютная погрешность, с которой в компьютере вычисляется функция e^{-x} , составляет $\Delta = 10^{-5}$.

Рассмотрим два пути нахождения суммы и, соответственно, две математические модели:

- 1) путем обычного суммирования;
- 2) использование формулы для суммы ряда, представляющего убывающую геометрическую прогрессию.

В первом случае общая абсолютная погрешность вычисления суммы S_1 будет определяться как сумма абсолютных погрешностей расчета каждого слагаемого ряда. Тогда

$$\Delta_{S1} = N\Delta = 10^{-3}$$

Во втором случае расчеты будут несколько сложнее. Формула для вычисления суммы будет иметь вид:

$$S_2 = \frac{1 - e^{-Na}}{1 - e^{-a}} \approx \frac{0,63}{0,01} = 63$$

Здесь для вычисления e^{-a} применяем известное соотношение: $e^{-x} \approx 1 - x$ для $x \ll 1$.

Абсолютную погрешность Δ_{S2} для дроби непосредственно не вычислить, можно вычислить лишь относительную погрешность δ_{S2} , а затем пересчитать ее в абсолютную по формуле:

$$\Delta_{S2} = \delta_{S2} S_2$$

В свою очередь, δ_{S2} находим как сумму относительных погрешностей вычисления числителя дроби $\delta_{\text{числ}}$ и ее знаменателя $\delta_{\text{знам}}$:

$$\delta_{S2} = \delta_{\text{числ}} + \delta_{\text{знам}}$$

Сначала $\delta_{\text{знам}}$ вычисляем по формуле:

$$\delta_{\text{знам}} = \frac{\Delta_{\text{знам}}}{1 - e^{-a}} \approx \frac{\Delta}{a} = 10^{-3}$$

Затем, в свою очередь, $\delta_{\text{числ}}$ вычисляем по формуле:

$$\delta_{\text{числ}} = \frac{\Delta_{\text{числ}}}{1 - e^{-Na}} \approx \frac{\Delta}{1 - e^{-1}} \approx 1,6 \cdot 10^{-5}$$

Учитывая, что $\delta_{\text{знам}} \gg \delta_{\text{числ}}$ последним можно пренебречь, тогда $\delta_{S2} \approx \delta_{\text{знам}}$

Теперь вычисляем Δ_{S2} :

$$\Delta_{S2} = 10^{-3} \cdot 63 = 63 \cdot 10^{-3},$$

Что в 63 раза больше, чем Δ_{S1}

Таким, образом, продемонстрировано, что использование при расчетах различных моделей может привести к существенно разным погрешностям в результатах вычислений.