

ПРЕДИСЛОВИЕ

Выпуск конкурентоспособной продукции, имеющей высокие потребительские характеристики, а также оказание информационных услуг, требует интенсификации процессов разработки новой техники, повышения качества проектов, разработки и организации производства изделий в короткие сроки.

В этом случае важное значение приобретают сроки и качество выполнения проектно-конструкторских работ, что обеспечивается применением новой технологии проектирования, основанной на использовании современных компьютерных методов моделирования.

Необходимость использования метода моделирования определяется тем, что многие объекты и проблемы, связанные с ними непосредственно исследовать или вовсе невозможно, или это исследование требует слишком много времени и средств.

Моделирование позволяет не только существенно сократить и облегчить натуральный эксперимент, который часто дорог, сложен, связан с аварийными состояниями, но и позволяет повысить достоверность теоретических исследований [1].

Моделирование представляет собой процесс замещения объекта исследования некоторой его моделью и проведение исследований на модели с целью получения необходимой информации об объекте. Теория замещения одних объектов (оригиналов) другими объектами (моделями) и исследования свойств объектов на их моделях называется *теорией моделирования* [1], [2], [3], [4]. Модель – это физический или абстрактный образ моделируемого объекта, удобный для проведения исследований и позволяющий адекватно отображать интересующие исследователя физические свойства и характеристики объекта. Удобство проведения исследований может определяться различными факторами: легкостью и доступностью получения информации, сокращением сроков и уменьшением материальных затрат на исследование и др.

Различают предметное и абстрактное моделирование. При предметном моделировании строят физическую модель, которая соответствующим образом отображает основные физические свойства и характеристики моделируемого объекта. При этом модель может иметь иную физическую природу в сравнении с моделируемым объектом (например, электронная модель механической или гидравлической системы). Если модель и объект одной и той же физической природы, то моделирование называют физическим.

Физическое моделирование широко применялось до недавнего времени при создании сложных технических объектов. Обычно изготавливался макетный или опытный образец технического объекта, проводились испытания, в процессе которых определялись его выходные параметры и характеристики, оценивались надежность функционирования и степень выпол-

нения технических требований, предъявляемых к объекту. Если вариант технической разработки оказывался неудачным, все повторялось сначала, т. е. осуществлялось повторное проектирование, изготовление опытного образца, испытания и т. д. Физическое моделирование сложных технических систем сопряжено с большими временными и материальными затратами.

Абстрактное моделирование связано с построением абстрактной модели. Такая модель представляет собой математические соотношения, графы, схемы, диаграммы и т. п. Наиболее мощным и универсальным методом абстрактного моделирования является компьютерное моделирование. Оно широко используется как в научных исследованиях, так и при проектировании.

Компьютерное моделирование позволяет посредством математических символов и зависимостей составить описание функционирования технического объекта в окружающей внешней среде, определить выходные параметры и характеристики, получить оценку показателей эффективности и качества, осуществить поиск оптимальной структуры и параметров объекта. Применение компьютерного моделирования при проектировании в большинстве случаев позволяет отказаться от физического моделирования, значительно сократить объемы испытаний и доводочных работ, обеспечить создание технических объектов с высокими показателями эффективности и качества. Одним из основных компонентов системы проектирования в этом случае становится компьютерная модель.

Формализация процесса проектирования на основе компьютерного моделирования позволяет его автоматизировать. Основными компонентами системы автоматизированного проектирования являются инженеринговые средства, включающие в себя компьютерные средства: для расчета режимов работы информационно-управляющих систем (ИУС) и модулей управления (процессорных, интерфейсных, информационных, технологических и др.); компьютерные модели объектов проектирования и их элементов; методы и алгоритмы выполнения проектных операций и процедур.

Полностью формализовать и автоматизировать процесс проектирования практически невозможно. При разработке концепции ИУС, формировании технического задания, выборе технического решения, синтезе структуры, принятии решений и на других этапах разработчик действует согласно своему опыту и интуиции.

Для разработки эффективной технологии автоматизированного проектирования необходимо детальное представление обо всех этапах и стадиях создания объекта с тем, чтобы осуществить их формализацию и математическое описание.

Высокий технический уровень изделия достигается в значительной мере на этапе функционального проектирования, на котором определяются основные параметры объекта. При недостаточной проработке проекта за-

траты на обеспечение качества, обусловленные необходимостью последующей доводки конструкции, достигают 10–20 % от полной стоимости продукции [5]. При этом 50–70 % общих причин дефектов продукции связано с ошибками в проектно-конструкторских решениях, 20–30 % – с недостатками технологических процессов, 5–15 % возникают по вине рабочих [5]. Поэтому главная задача разработчика состоит в том, чтобы выявить и устранить потенциальные источники дефектов еще на стадии проектирования.

Процедуры функционального проектирования, как правило, почти полностью поддаются формализации, что позволяет рассчитывать (определять) и выбирать оптимальные параметры и структуры технического объекта. При этом используются математические модели создаваемых объектов, модели оценки и принятия решений, которые в виде соответствующих алгоритмов реализуются при проектировании.

При решении задач синтеза структуры, моделировании процессов функционирования объектов с переменной структурой возникает необходимость постоянного изменения математической модели. Поэтому большое внимание уделяется методам автоматизированного формирования математических моделей, создания базы унифицированных компьютерных моделей элементов ИУС.

На различных этапах и стадиях проектирования сложной технической системы используются различные математические модели. На ранних стадиях – обычно модели простые, но чем подробнее проработка проекта, тем сложнее нужна модель. Для целей моделирования ИУС применяется различный математический аппарат, теория информационно-управляющих систем, законы физики и др.

Поскольку все проектные работы носят оптимизационный характер, то процесс автоматизированного проектирования (АПр) является много итерационным, а многомерность и многокритериальность также осложняют решение задачи. На заключительных этапах АПр часто приходится использовать вероятностные модели, с тем, чтобы исследовать процессы функционирования технической системы в условиях, максимально приближенных к реальным.

Математические модели ИУС позволяют анализировать процессы их функционирования, получать оценки выходных параметров различных предлагаемых вариантов технических решений и сравнивать их между собой. Но конечной целью проектирования является получение наилучшего технического решения из числа возможных альтернатив. Это достигается в процессе решения задачи синтеза, которая направлена на определение структуры и оптимальных параметров объекта.

Аналитический подход основывается на построении формульных зависимостей, связывающих параметры и элементы системы. Такой подход долгое время и был собственно математическим подходом. Однако при рассмотрении сложных систем строгие математические зависимости весь-

ма сложны, требуется большое количество измерений для получения требуемых значений параметров.

Анализ характеристик процессов функционирования сложных систем с помощью только аналитических методов исследования наталкивается на значительные трудности, приводящие к необходимости существенного упрощения моделей либо на этапе их построения, либо в процессе работы с моделью, что снижает достоверность результатов.

Имитационный (статистический) подход в моделировании базируется на использовании предельной теоремы Чебышева при вероятностном представлении параметров системы. На основе предварительного изучения моделируемой системы достаточно просто определяются виды и значения законов распределения случайных величин параметров. В рамках имитационного подхода используются аналитические зависимости между параметрами элементов системы, однако эти зависимости имеют более обобщенный, упрощенный характер. Они значительно проще, нежели зависимости в рамках аналитического подхода.

Математическое моделирование систем, в том числе и информационных, имеет целью оптимизацию структуры систем, выбор наиболее оптимальных режимов функционирования систем, определение требуемых характеристик аппаратного оборудования и программного обеспечения.

Математическое моделирование технологических процессов, в том числе и информационных, имеет основными целями нахождение оптимальных или приемлемых характеристик самого объекта, нахождение оптимальных режимов обработки, обучение персонала, обеспечение определенных функций управления.

Процесс моделирования включает три элемента: субъект (исследователь), объект исследования, модель и состоит из следующих основных этапов.

1. Конструирование или поиск в реальном мире подходящей модели исследуемого объекта.

2. Замена объекта исследования моделью и проведение "модельных" экспериментов.

3. Перенос полученных в ходе проведения исследования на модели знаний на объект моделирования. При этом знания о модели должны быть скорректированы с учетом тех свойств объекта, которые не нашли отражения или были изменены при построении модели.

4. Практическая проверка получаемых с помощью моделей знаний о реальном объекте.

1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ МОДЕЛИРОВАНИЯ

1.1. Основные понятия теории моделирования

Целью создания моделей является изучение, описание, проектирование или оптимизация некоторого объекта или процесса. Например, в общественной деятельности модели используют для проведения анализа при принятии решений.

Модель – это такой материальный или мысленно представляемый объект, который в процессе исследования замещает объект оригинал так, что его непосредственное изучение дает новые знания об объекте-оригинале.

Моделирование – это замещение одного объекта другим с целью изучения важнейших свойств замещаемого объекта. Замещаемый объект называется *оригиналом*, замещающий – *моделью*.

Моделирование тесно связано с такими гносеологическими категориями, как: абстракция, формализация, идеализация, аналогия, гипотеза и др.

В математической теории модель определяется как результат отображения одной абстрактной математической структуры на другую, также абстрактную, либо как результат интерпретации первой модели в терминах и образах второй.

Главная особенность моделирования в том, что это метод опосредованного познания с помощью объектов-заместителей. Модель выступает как своеобразный инструмент для познания, который исследователь ставит между собой и объектом и с помощью которого изучает интересующий его объект.

Моделирование представляет собой процесс построения, изучения и применения моделей. Процесс моделирования включает три элемента: субъект (исследователь), объект исследования, модель и состоит из следующих основных этапов:

1. конструирование или поиск в реальном мире подходящей модели исследуемого объекта;
2. замена объекта исследования моделью и проведение «модельных» экспериментов;
3. «перенос знаний с модели на оригинал». При этом знания о модели должны быть скорректированы с учетом тех свойств объекта-оригинала, которые не нашли отражения или были изменены при построении модели;
4. практическая проверка получаемых с помощью моделей знаний о реальном объекте.

Моделирование тесно связано с такими гносеологическими категориями, как: абстракция, формализация, идеализация, аналогия, гипотеза и др. Процесс моделирования неизбежно включает и построение абстраций, и умозаключения по аналогии, и конструирование научных гипотез.

Абстракция служит для представления существенных свойств объекта без детального учета его внутренней организации и конкретной реализации. Модель отражает только те стороны явлений, которые существенны в

данном исследовании, и позволяет абстрагироваться от второстепенных факторов. Элементарная форма абстракции – типизация объектов.

Формализация – отображение объекта или явления в знаковой форме какого-либо искусственного языка (например, математики) с целью исследования объектов и свойств через формальное исследование соответствующих знаков. Термин «формальный» означает, что объекты и операции над ними рассматриваются чисто формально, без какой бы то ни было содержательной интерпретации символов.

Идеализация – это мысленное конструирование объектов, которые в реальности не существуют (например, идеальный газ, идеальный импульс, неограниченная память и др.). В результате идеализации реальные объекты лишаются некоторых присущих им свойств и наделяются гипотетическими свойствами.

Аналогия – прием, используемый в методах познания состоящий в том, что знания о предметах и явлениях выявляются на основании изучения сходных с ними в каком-либо отношении (по структуре, составу, принципу действия, используемым средствам, характеристикам) объектов.

Однако, делая выводы на основании аналогии нужно знать, насколько эти выводы достоверны. Модель никогда полностью не тождественна оригиналу, между ними всегда есть определенные различия. Поэтому выводы на основании аналогии всегда имеют не абсолютно достоверный, а предположительный характер.

Гипотеза – предположение, основывающееся на догадке, наблюдении, опыте и требующее подтверждения или доказательства. В процессе познания каждая гипотеза подвергается проверке. С накоплением новых фактов одна гипотеза может быть заменена другой.

При этом часто старая гипотеза не отбрасывается полностью, а только исправляется и уточняется. В результате исправленная и уточненная гипотеза превращается в теорию или закон.

Моделирование – циклический процесс. Это означает, что за первым четырехэтапным циклом может последовать второй, третий и т. д. При этом знания об исследуемом объекте расширяются и уточняются, а исходная модель постепенно совершенствуется.

Наука моделирования состоит в разделении процесса моделирования (системы, модели) на этапы (подсистемы, подмодели), детальном изучении каждого этапа, взаимоотношений, связей, отношений между ними и затем эффективного описания их с максимально возможной степенью формализации и адекватности. В случае нарушения этих правил получаем не модель системы, а модель "собственных и неполных знаний".

Моделирование (в значении "метод", "модельный эксперимент") рассматривается как особая форма эксперимента, эксперимента не над самим оригиналом (это называется простым или обычным экспериментом), а над копией (заместителем) оригинала. Здесь важен изоморфизм систем (ориги-

нальной и модельной) – изоморфизм, как самой копии, так и знаний, с помощью которых она была предложена.

Модели и моделирование применяются по основным направлениям:

- обучение (как моделям, моделированию, так и самим моделям);
- познание и разработка теории исследуемых систем (с помощью каких-либо моделей, моделирования, результатов моделирования);
- прогнозирование (выходных данных, ситуаций, состояний системы);
- управление (системой в целом, отдельными подсистемами системы), выработка управленческих решений и стратегий);
- автоматизация (системы или отдельных подсистем системы).

1.2. Теория подобия в моделировании систем

В основе метода моделирования лежит теория подобия, устанавливающая те условия, при которых модель действительно отражает оригинал.

Подобие явлений. Характеризующееся соответствием (например, пропорциональностью) величин, участвующих в изучаемых явлениях, происходящих в оригинале и в моделях, по степени соответствия параметров модели и оригинала может быть трех видов: *полное подобие, неполное подобие, приближенное подобие.*

Неполное подобие, в отличие от полного подобия, связано с изучением процессов только во времени или только в пространстве. Приближенное подобие реализуется при некоторых упрощающих допущениях, что, как правило, приводит к искажениям, которые должны заранее оцениваться количественно.

Все виды подобия подчиняются общим закономерностям, описываемым теоремами о подобии и некоторыми дополнительными положениями.

Помимо соблюдения условий подобия эффективность моделирования зависит от того, в какой степени модель согласована с культурной средой, в которой ей предстоит функционировать. Она должна органически вписываться в эту среду, а не представлять собой чуждый ей элемент. Еще одним аспектом согласованности является обеспеченность модели ресурсами.

Основные различия между моделью и действительностью связаны с конечностью, упрощенностью и приближенностью моделей.

Основное противоречие познавательных моделей состоит в необходимости использования для познания бесконечного мира ограниченных средств. Поэтому модель подобна оригиналу лишь в конечном числе отношений.

Познавательные возможности модели обуславливаются тем, что модель отображает лишь наиболее существенные черты объекта-оригинала. Вопрос о необходимой и достаточной мере сходства оригинала и модели и об адекватности модели требует особого анализа.

Модель, с помощью которой успешно достигается поставленная цель, называется адекватной этой цели.

Модель утрачивает свой смысл как в случае тождества с оригиналом (тогда она перестает быть моделью), так и в случае чрезмерного отличия от оригинала.

Сходство модели и действительности проверяется на практике. При моделировании, как и в других методах исследования, решающим критерием истинности является практика. Помимо безусловно истинного, в модели есть, как правило, и верное лишь при определенных условиях, и даже неправильное, т.е. не имеющее отношение к оригиналу.

1.3. Системный подход в моделировании систем

В настоящее время при анализе и синтезе сложных систем, таких, как ИУС, получил развитие *системный подход*, который отличается от классического подхода. Классический подход рассматривает систему путем перехода от частного к общему и синтезирует модель путем слияния ее компонент, разрабатываемых отдельно. В отличие от этого системный подход предполагает последовательный переход от общего к частному, когда в основе рассмотрения лежат цель и задачи, причем исследуемый объект выделяется из окружающей среды.

Системный анализ (СА) признается в настоящее время наиболее конструктивным из направлений системных исследований. Этот термин впервые появился в 1948 г. в работах корпорации RAND в связи с задачами военного управления. Получил распространение в отечественной литературе после перевода книги С. Оптнера «системный анализ деловых и промышленных проблем».

Системный анализ – обобщает методологию исследования сложных технических, природных и социальных систем [6], [7].

Система есть множество компонент, взаимодействующих друг с другом и служащих общему назначению, или цели.

Система имеет следующие основные характеристики (рис.1.1): 1. компоненты; 2. взаимодействия; 3. границу; 4. цель; 5. внешнюю среду; 6. вход; 7. выход; 8. интерфейс; 9. ограничения.

Системные характеристики:

1. Компонент есть либо неделимая часть, либо агрегат, состоящий их частей, называемый подсистемой.

2. Компоненты взаимодействуют между собой таким образом, что функционирование одного влияет на функционирование другого компонента.

3. Система имеет границу, внутри которой содержатся все компоненты, и которая устанавливает пределы системы, отделяя ее от других систем.

4. Все компоненты работают вместе, чтобы достичь цель существования системы.

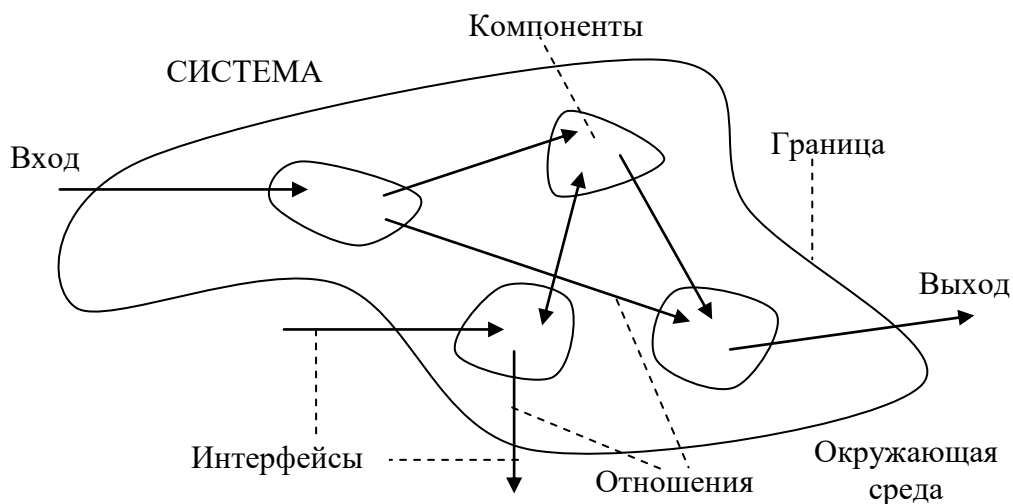


Рис. 1.1. Характеристики системы

5. Система оперирует внутри окружающей среды – всего, что находится за границей системы. Окружающая среда влияет на систему и подвергается влиянию системы.

6. Точка, в которой система взаимодействует со средой, называется интерфейсом.

7. Система имеет ограничения функционирования.

В других определениях системы понятие «цель» присутствует в неявном виде [7]: по определению Ф. Е. Темникова) «система – организованное множество (в котором цель проявляется при раскрытии понятия организованности)», далее – в виде конечного результата, системообразующего критерия, функции (В. И. Вернадский, У. Р. Гибсон, П. К. Анохин). Символически эту группу определений можно представить как: $S = \langle A, B, Z \rangle$, где A – элементы ИУС; B – параметры элементов системы; Z – цель, совокупность или структура целей.

Далее, в определение системы можно включить наблюдателя N , т.е. лицо, представляющее объект или процесс в виде системы при их исследовании или принятии решений: $S = \langle A, B, R, N \rangle$.

По определению Ю.И. Черняка [7], система есть отражение в сознании субъекта (исследователя, наблюдателя) свойства объектов и их отношений в решении задачи исследования, познания.

На основе анализа различных определений понятия «система» [4], [6], [8], [9] сформулировано определение сложных динамических систем (СДС), которые обладают следующими базовыми свойствами (системообразующими факторами) [7], [9]:

- целостность и возможность декомпозиции на элементы A (объекты, подсистемы);
- наличие стабильных связей (отношений) R между элементами A ;
- упорядоченность (организация) элементов в определенную структуру (Str);

- наделение элементов параметрами (P);
 - наличие синергетических (интегративных) свойств Q , которыми не обладают ни один из элементов системы;
 - наличие множества законов, правил и операций Z с вышеперечисленными атрибутами системы;
 - наличие цели функционирования и развития (G).
- Таким образом, система есть совокупность $Syst = \{A, Str, Q, R, Z, G\}$ (см. рис. 1.2).

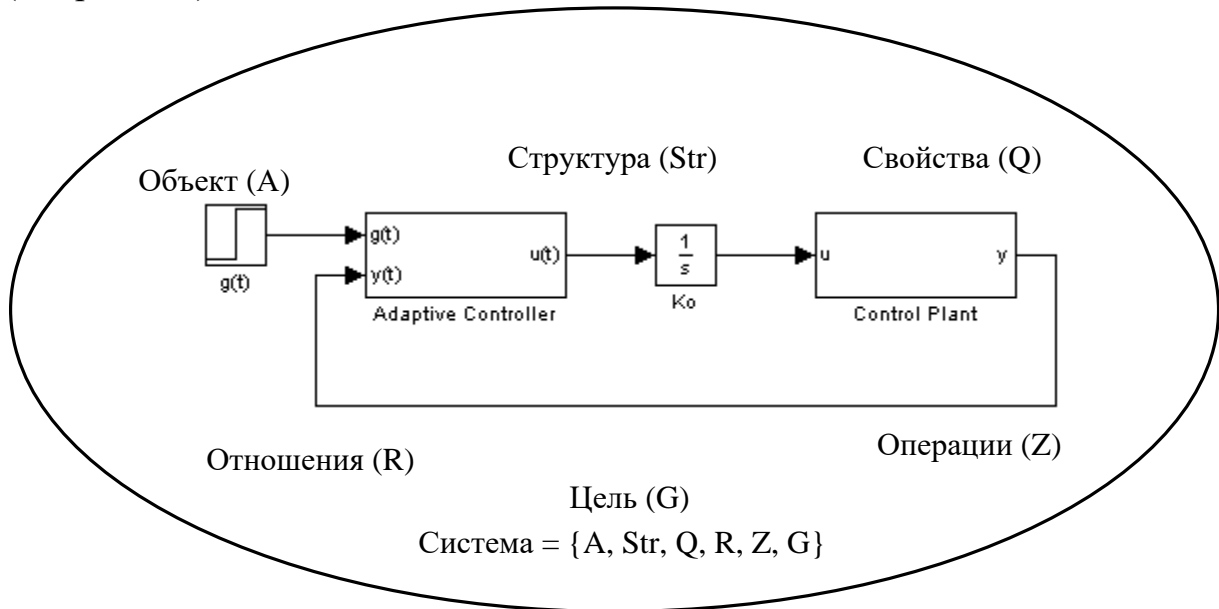


Рис. 1.2. Содержание понятия «система»

Наличие множества Z законов, правил и операций способствует созданию того формального аппарата, который позволяет на математическом (абстрактном) уровне строить из множества A элементов и множества R связей различные структуры систем, а также анализировать их и синтезировать системы с заданными свойствами.

Данное определение системы используется в дальнейшем при исследовании (анализе, моделировании) сложных управляемых систем с целью установления связи между структурой, параметрами и свойствами системы при их поведении в проблемных ситуациях.

Свойство системности является всеобщим свойством материи. Современные научные данные [7] и современные системные представления позволяют говорить о мире как о бесконечной иерархической системе систем. Причем части системы находятся в развитии, на разных стадиях развития, на разных уровнях системной иерархии и организации. Системность как всеобщее свойство материи проявляется через следующие составляющие: системность практической деятельности, системность познавательной деятельности и системность среды, окружающей человека.

При системном подходе к моделированию ИУС необходимо прежде всего четко определить цель моделирования. Поскольку невозможно полностью смоделировать реально функционирующую систему, создается модель под определенные задачи. Таким образом, применительно к вопросам моделирования цель возникает из требуемых задач моделирования, что позволяет подойти к выбору критерия и оценить, какие элементы войдут в создаваемую модель. Поэтому необходимо иметь критерий отбора отдельных элементов в создаваемую модель.

Данное определение более полно отражает содержательную сторону системы, чем известные определения, основанные на первых трех признаках: элементах, связях и их упорядоченности в единое целое. Параметризация структурных элементов позволяет конкретизировать систему, придавать ей индивидуальность, а также выделять то множество свойств, которое присуще данной системе. При этом к свойствам системы можно отнести ее способность к адаптации, к самоорганизации, к обеспечению устойчивости, к выполнению различных сложных функций (самосохранения, саморазвития и т. д.). К свойствам системы можно отнести и ее способность к формированию целей функционирования и развития и к организации их достижения.

Моделирование – метод системного анализа. Но часто в системном анализе при модельном подходе исследования может совершаться одна методическая ошибка, а именно, – построение корректных и адекватных моделей (подмоделей) подсистем системы и их логически корректная увязка не дает гарантий корректности построенной таким способом модели всей системы. Модель, построенная без учета связей системы со средой и ее поведения по отношению к этой среде, может часто лишь служить еще одним подтверждением теоремы Геделя, а точнее, ее следствия, утверждающего, что в сложной изолированной системе могут существовать истины и выводы, корректные в этой системе и некорректные вне ее.

1.3.1. Подходы к исследованию систем

Важным является определение структуры системы, совокупность связей между элементами системы, отражающих их взаимодействие (рис. 1.3). Структура системы может измениться извне, а также изнутри.

Переход системы из одного состояния в другое означает функционирование системы $S(t)$ т. е. движение в пространстве состояний Z .

Создаваемая модель M с точки зрения системного подхода тоже является системой $S' = S'(M)$ и может рассматриваться по отношению к внешней среде E .

Наиболее просты по представлению модели, в которых сохраняется прямая аналогия явления.

Применяют такие модели, в которых нет переноса аналогии, а сохраняются лишь законы и общие закономерности поведения элементов системы S .

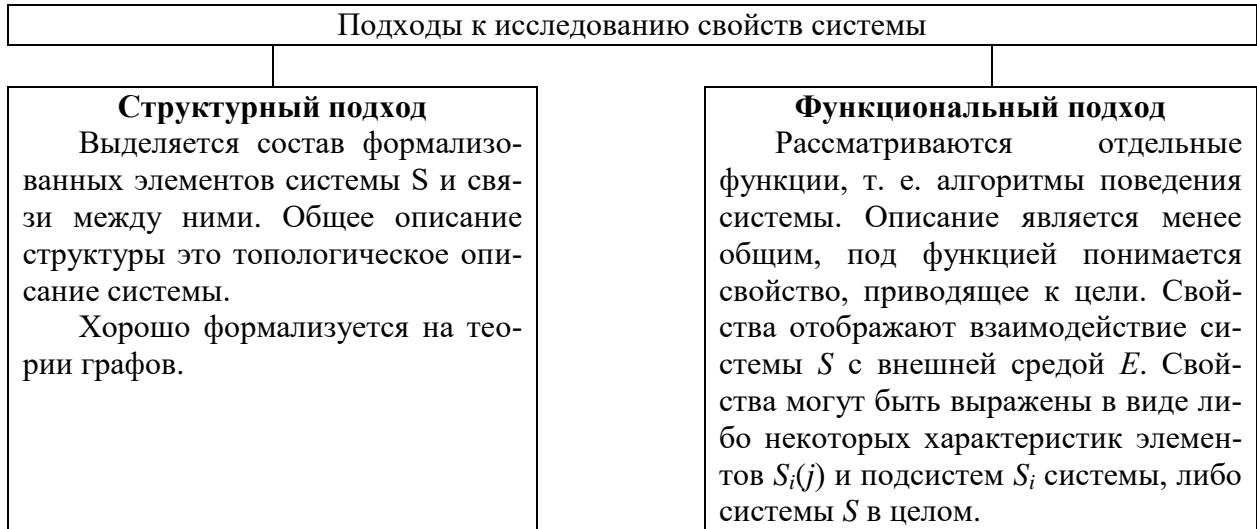


Рис. 1.3. Подходы к исследованию свойств системы

Важным для системного подхода является определение *структуры системы* – совокупности связей между элементами системы, отражающих их взаимодействие. Структура системы может изучаться извне с точки зрения состава отдельных подсистем и отношений между ними, а также изнутри, когда анализируются отдельные свойства, позволяющие системе достигать заданной цели, т. е. когда изучаются функции системы. В соответствии с этим наметился ряд подходов к исследованию структуры системы с ее свойствами, к которым следует прежде всего отнести структурный и функциональный.

При *структурном подходе* выявляются состав выделенных элементов системы S и связи между ними. Совокупность элементов и связей между ними позволяет судить о структуре системы. Последняя в зависимости от цели исследования может быть описана на разных уровнях рассмотрения. Наиболее общее описание структуры – это топологическое описание, позволяющее определить в самых общих понятиях составные части системы и хорошо формализуемое на базе теории графов.

Менее общим является функциональное описание, когда рассматриваются отдельные функции, т. е. алгоритмы поведения системы, и реализуется *функциональный подход*, оценивающий функции, которые выполняет система, причем под функцией понимается свойство, приводящее к достижению цели. Поскольку функция отображает свойство, а свойство отображает взаимодействие системы S с внешней средой E , то свойства могут быть выражены в виде либо некоторых характеристик элементов $S_i(j)$ и подсистем S_i системы, либо системы S в целом.

При наличии некоторого эталона сравнения можно ввести количественные и качественные характеристики систем. Для количественной характеристики вводятся числа, выражающие отношения между данной характеристикой и эталоном. Качественные характеристики системы находятся, например, с помощью метода экспертных оценок.

Проявление функций системы во времени $S(t)$, т. е. *функционирование системы*, означает переход системы из одного состояния в другое, т. е. движение в пространстве состояний Z . При эксплуатации системы S весьма важно качество ее функционирования, определяемое показателем эффективности и являющееся значением критерия оценки эффективности. Существуют различные подходы к выбору критериев оценки эффективности.

Система S может оцениваться либо совокупностью частных критериев, либо некоторым общим интегральным критерием.

Следует отметить, что создаваемая модель M с точки зрения системного подхода также является системой, т. е. $S' = S'(M)$, и может рассматриваться по отношению к внешней среде E . Наиболее просты по представлению модели, в которых сохраняется прямая аналогия явления. Применяют также модели, в которых нет прямой аналогии, а сохраняются лишь законы и общие закономерности поведения элементов системы S . Правильное понимание взаимосвязей как внутри самой модели M , так и взаимодействия ее с внешней средой E в значительной степени определяется тем, на каком уровне находится наблюдатель.

Когда проводятся исследования по разработке имитационной модели реальной системы, невозможно моделировать сразу всю систему в целом. Вначале моделируются подсистемы, которые имеют отношение к исследуемой проблеме. Они включают части системы на различных уровнях детализации.

Модель симуляции, в основном, используется для изучения реальной системы, которая пока не существует. Интересны измерения количественных характеристик функционирования системы для изучения возможных значений входных параметров. Такие количественные измерения функционирования очень полезны для процессов принятия управленческих решений. Основные шаги включают выполнение упражнений симуляции. Все соответствующие переменные изучаемой системы организованы в две группы. Те, которые рассматриваются как данные, которыми нельзя манипулировать (неуправляемые переменные) и те, которыми можно манипулировать так, чтобы прийти к решению (управляемые переменные). Различие между управляемыми и неуправляемыми переменными зависит в основном от точки зрения на изучение системы.

Простой подход к изучению взаимосвязей между отдельными частями модели предусматривает рассмотрение их как отражение связей между отдельными подсистемами объекта. Такой классический подход может быть использован при создании достаточно простых моделей. Процесс синтеза

модели M на основе классического (индуктивного) подхода представлен на рис. 1.4, а [4].

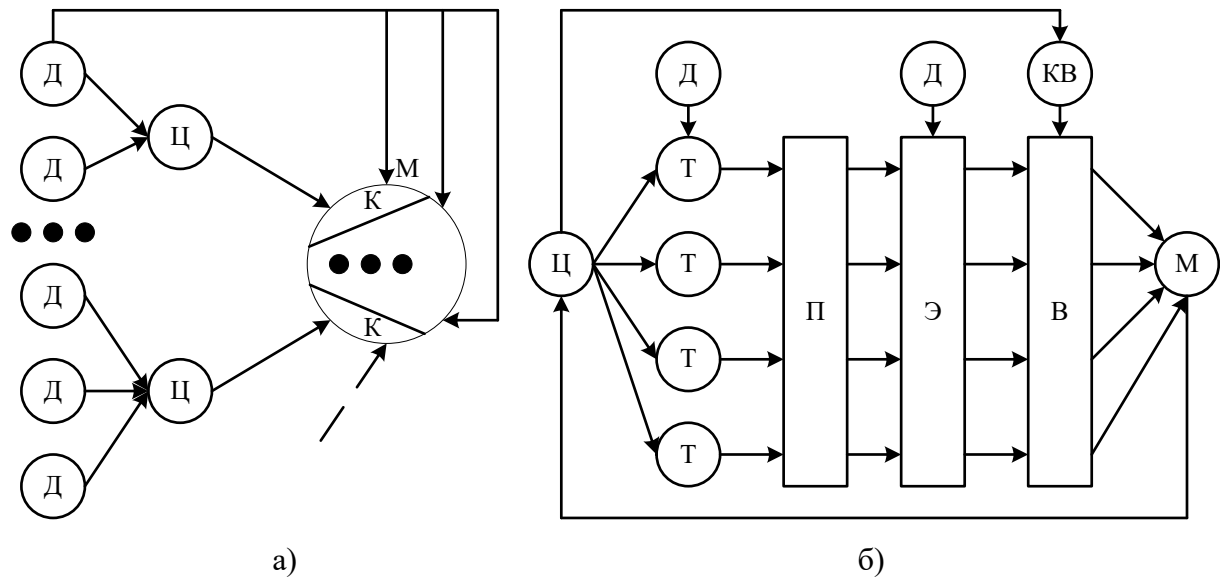


Рис. 1.4. Процесс синтеза модели на основе классического (а) и системного (б) подходов

Реальный объект, подлежащий моделированию, разбивается на отдельные подсистемы, т. е. выбираются исходные данные D для моделирования и ставятся цели C , отображающие отдельные стороны процесса моделирования. По отдельной совокупности исходных данных D ставится цель моделирования отдельной стороны функционирования системы, на базе этой цели формируется некоторая компонента K будущей модели. Совокупность компонент объединяется в модель M .

Таким образом, разработка модели M на базе классического подхода означает суммирование отдельных компонент в единую модель, причем каждая из компонент решает свои собственные задачи и изолирована от других частей модели. Поэтому классический подход может быть использован для реализации сравнительно простых моделей, в которых возможно разделение и взаимно независимое рассмотрение отдельных сторон функционирования реального объекта. Для модели сложного объекта такая разобщенность решаемых задач недопустима, так как приводит к значительным затратам ресурсов при реализации модели на базе конкретных программно-технических средств. Можно отметить две отличительные стороны классического подхода: наблюдается движение от частного к общему, создаваемая модель (система) образуется путем суммирования отдельных ее компонент и не учитывается возникновение нового системного эффекта.

С усложнением объектов моделирования возникла необходимость наблюдения их с более высокого уровня. В этом случае наблюдатель (разработчик) рассматривает данную систему S как некоторую подсистему какой-то метасистемы, т. е. системы более высокого ранга, и вынужден перейти на позиции нового системного подхода, который позволит ему по-

строить не только исследуемую систему, решающую совокупность задач, но и создавать систему, являющуюся составной частью метасистемы.

Системный подход получил применение в системотехнике в связи с необходимостью исследования больших реальных систем, когда сказалась недостаточность, а иногда ошибочность принятия каких-либо частных решений. На возникновение системного подхода повлияли увеличивающееся количество исходных данных при разработке, необходимость учета сложных стохастических связей в системе и воздействий внешней среды E . Все это заставило исследователей изучать сложный объект не изолированно, а во взаимодействии с внешней средой, а также в совокупности с другими системами некоторой метасистемы.

Системный подход позволяет решить проблему построения сложной системы с учетом всех факторов и возможностей, пропорциональных их значимости, на всех этапах исследования системы S и построения модели M . Системный подход означает, что каждая система S является интегрированным целым даже тогда, когда она состоит из отдельных разобщенных подсистем. Таким образом, в основе системного подхода лежит рассмотрение системы как интегрированного целого, причем это рассмотрение при разработке начинается с главного – формулировки цели функционирования. Процесс синтеза модели M на базе системного подхода условно представлен на рис. 1.4, б. На основе исходных данных D , которые известны из анализа внешней системы, тех ограничений, которые накладываются на систему сверху либо исходя из возможностей ее реализации, и на основе цели функционирования формулируются исходные требования T к модели системы S . На базе этих требований формируются ориентировочно некоторые подсистемы $П$, элементы \mathcal{E} и осуществляется наиболее сложный этап синтеза – выбор B составляющих системы, для чего используются специальные критерии выбора $KВ$.

При моделировании необходимо обеспечить максимальную эффективность модели системы. Эффективность обычно определяется как некоторая разность между какими-то показателями ценности результатов, полученных в итоге эксплуатации модели, и теми затратами, которые были вложены в ее разработку и создание.

Стадии разработки моделей. На базе системного подхода может быть предложена и некоторая последовательность разработки моделей, когда выделяют две основные стадии проектирования: макропроектирование и микропроектирование.

На стадии макропроектирования на основе данных о реальной системе S и внешней среде E строится модель внешней среды, выявляются ресурсы и ограничения для построения модели системы, выбирается модель системы и критерии, позволяющие оценить адекватность модели M реальной системы S . Построив модель системы и модель внешней среды, на основе критерия эффективности функционирования системы в процессе модели-

рования выбирают оптимальную стратегию управления, что позволяет реализовать возможности модели по воспроизведению отдельных сторон функционирования реальной системы S .

Стадия микропроектирования в значительной степени зависит от конкретного типа выбранной модели. В случае имитационной модели необходимо обеспечить создание информационного, математического, технического и программного обеспечений систем моделирования. На этой стадии можно установить основные характеристики созданной модели, оценить время работы с ней: затраты ресурсов для получения заданного качества соответствия модели процессу функционирования системы S .

Независимо от типа используемой модели M при ее построении необходимо руководствоваться рядом принципов системного подхода: 1) пропорционально-последовательное продвижение по этапам и направлениям создания модели; 2) согласование информационных, ресурсных, надежных и других характеристик; 3) правильное соотношение отдельных уровней иерархии в системе моделирования; 4) целостность отдельных обособленных стадий построения модели.

Модель M должна отвечать заданной цели ее создания, поэтому отдельные части должны компоноваться взаимно, исходя из единой системной задачи. Цель может быть сформулирована качественно, тогда она будет обладать большей содержательностью и длительное время может отображать объективные возможности данной системы моделирования. При количественной формулировке цели возникает целевая функция, которая точно отображает наиболее существенные факторы, влияющие на достижение цели.

Построение модели относится к числу системных задач, при решении которых синтезируют решения на базе огромного числа исходных данных, на основе предложений больших коллективов специалистов. Использование системного подхода в этих условиях позволяет не только построить модель реального объекта, но и на базе этой модели выбрать необходимое количество управляющей информации в реальной системе, оценить показатели ее функционирования и тем самым на базе моделирования найти наиболее эффективный вариант построения и выгодный режим функционирования реальной системы S .

1.4. Использование моделирования при исследовании и проектировании сложных систем

Одна из проблем современной науки и техники – разработка и внедрение в практику проектирования современных методов исследования характеристик сложных информационно-управляющих и информационно - вычислительных систем различных уровней (например, автоматизированных систем научных исследований и комплексных испытаний, АСУ технологическими процессами, вычислительных систем, комплексов и сетей, инфор-

мационных систем, цифровых сетей интегрального обслуживания и т. д.). При проектировании сложных систем и их подсистем возникают многочисленные задачи, требующие оценки количественных и качественных закономерностей процессов функционирования таких систем, проведения структурного алгоритмического и параметрического их синтеза.

♦ **Особенности разработки систем.** Рассматриваемые в данном пособии ИУС, автоматизированные системы обработки информации и управления, относятся к классу больших систем, этапы проектирования, внедрения, эксплуатации и эволюции которых в настоящее время невозможны без использования различных видов моделирования. При моделировании таких систем необходимо учитывать: 1) сложность структуры и стохастичность связей между элементами; 2) неоднозначность алгоритмов поведения при различных условиях; 3) большое количество параметров и переменных; 4) неполноту и недетерминированность исходной информации; 5) разнообразие и вероятностный характер воздействий внешней среды и т. д. Необходимо разработать методики и алгоритмы их моделирования, позволяющие: 1) в соответствующей форме представить процессы функционирования ИУС; 2) описание протекания этих процессов с помощью математических моделей; 3) получение результатов экспериментов с моделями по оценке характеристики исследуемых объектов. Особенностью ИУС является то, что в их состав входит большое многообразие подсистем. В этом случае применение метода моделирования преследует конкретные цели, а эффективность метода зависит от того, насколько грамотно разработчик использует возможности моделирования.

Независимо от разбиения конкретной ИУС на подсистемы при проектировании каждой из них необходимо выполнить внешнее проектирование (макропроектирование) и внутреннее проектирование (микропроектирование). Так как на этих стадиях разработчик преследует различные цели, то и используемые при этом методы и средства моделирования могут существенно отличаться.

На стадии макропроектирования должна быть разработана обобщенная модель процесса функционирования ИУС, позволяющая разработчику получить ответы на вопросы об эффективности различных стратегий управления объектом при его взаимодействии с внешней средой. Стадию внешнего проектирования можно разбить на анализ и синтез. При анализе изучают объект управления, строят модель воздействий внешней среды, определяют критерии оценки эффективности, имеющиеся ресурсы, необходимые ограничения. Конечная цель стадии анализа – построение модели объекта управления для оценки его характеристик. При синтезе на этапе внешнего проектирования решаются задачи выбора стратегии управления на основе модели объекта моделирования, т. е. сложной системы.

На стадии микропроектирования разрабатывают модели с целью создания эффективных подсистем. Причем используемые методы и средства

моделирования зависят от того, какие конкретно обеспечивающие подсистемы разрабатываются: информационные, математические, технические, программные и т. д.

♦ **Особенности использования моделей.** Выбор метода моделирования и необходимая детализация моделей существенно зависят от этапа разработки сложной системы. На этапах обследования объекта управления, например промышленного предприятия, и разработки технического задания на ИУС модели в основном носят описательный характер и преследуют цель наиболее полно представить в компактной форме информацию об объекте, необходимую разработчику системы.

На этапах разработки технического и рабочего проектов систем, модели отдельных подсистем детализируются, и моделирование служит для решения конкретных задач проектирования, т. е. выбора оптимального по определенному критерию при заданных ограничениях варианта из множества допустимых. Поэтому в основном на этих этапах проектирования сложных систем используются модели для целей синтеза.

Целевое назначение моделирования на этапе внедрения и эксплуатации сложных ИУС – это исследование возможных ситуаций для принятия обоснованных и перспективных решений по рабочему функционированию таких систем. Моделирование (имитацию) также широко применяют при обучении и тренировке персонала автоматизированных систем управления, вычислительных комплексов и сетей, информационных систем в различных сферах. В этом случае моделирование носит характер деловых игр. Модель, реализуемая обычно на персональном компьютере (ПК), воспроизводит поведение управляемого объекта и внешней среды, а люди в определенные моменты времени принимают решения по управлению объектом.

1.5. Общая характеристика проблемы моделирования систем

С развитием системных исследований, с расширением экспериментальных методов изучения реальных явлений и процессов все большее значение приобретают абстрактные методы. При создании ИУС важное значение имеют математические методы анализа и синтеза, целый ряд открытий базируется на чисто теоретических изысканиях. Однако было бы неправильно забывать о том, что основным критерием любой теории является практика. Схема основы научного подхода к процессу разработки математических моделей показана на рис. 1.5.

Экспериментальные исследования систем. Одновременно с развитием теоретических методов анализа и синтеза совершенствуются и методы экспериментального изучения реальных объектов, появляются новые средства исследования. Однако эксперимент был и остается одним из основных и существенных инструментов познания. Подобие и моделирование позволяют по новому описать реальный процесс и упростить экспериментальное его изучение. Совершенствуется и само понятие моделирования. Если

раньше моделирование означало реальный физический эксперимент либо построение макета, имитирующего реальный процесс, то в настоящее время появились новые виды моделирования, в основе которых лежит постановка не только физических, но также и математических экспериментов.

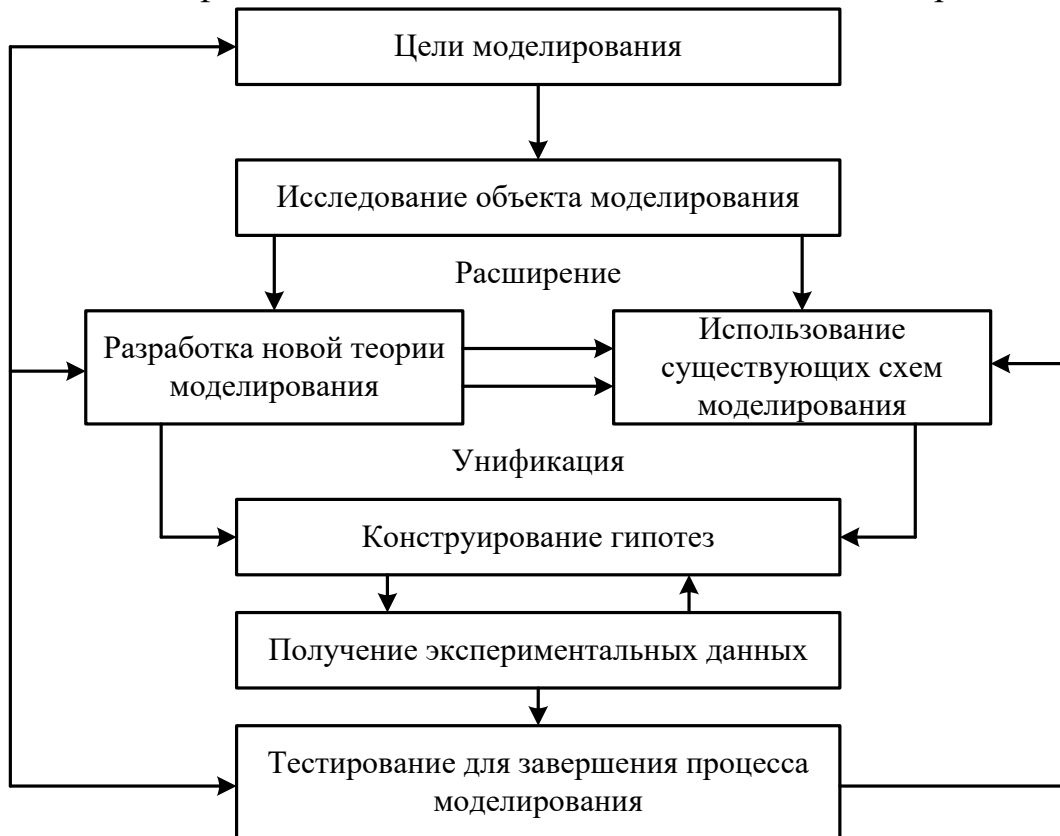


Рис. 1.5. Основы научного подхода

Познание реальной действительности является длительным и сложным процессом. Определение качества функционирования большой системы, выбор оптимальной структуры и алгоритмов поведения, построение системы S в соответствии с поставленной перед нею целью – основная проблема при проектировании современных систем, поэтому моделирование можно рассматривать как один из методов, используемых при проектировании и исследовании больших систем.

Моделирование базируется на некоторой аналогии реального и мысленного эксперимента. *Аналогия* - основа для объяснения изучаемого явления, однако критерием истины может служить только практика, только опыт. Хотя современные научные гипотезы могут создаваться чисто теоретическим путем, но, по сути, базируются на широких практических знаниях. Для объяснения реальных процессов выдвигаются гипотезы, для подтверждения которых ставится эксперимент либо проводятся такие теоретические рассуждения, которые логически подтверждают их правильность. В широком смысле под *экспериментом* можно понимать некоторую проце-

дуру организации и наблюдения каких-то явлений, которые осуществляют в условиях, близких к естественным, либо имитируют их.

Различают *пассивный эксперимент*, когда исследователь наблюдает протекающий процесс, и *активный*, когда наблюдатель вмешивается и организует протекание процесса. В последнее время распространен активный эксперимент, поскольку именно на его основе удастся выявить критические ситуации, получить наиболее интересные закономерности, обеспечить возможность повторения эксперимента в различных точках и т. д.

В основе любого вида моделирования лежит некоторая *модель*, имеющая соответствие, базирующееся на некотором общем качестве, которое характеризует реальный объект. Объективно реальный объект обладает некоторой формальной структурой, поэтому для любой модели характерно наличие некоторой структуры, соответствующей формальной структуре реального объекта, либо изучаемой стороне этого объекта.

В основе моделирования лежат *информационные процессы*, поскольку само создание модели M базируется на информации о реальном объекте. В процессе реализации модели получается информация о данном объекте, одновременно в процессе эксперимента с моделью вводится управляющая информация, существенное место занимает обработка полученных результатов, т. е. информация лежит в основе всего процесса моделирования [3], [10].

Характеристики моделей систем. В качестве объекта моделирования выступают сложные организационно-технические системы, которые можно отнести к классу больших систем. Более того, по своему содержанию и созданная модель M также становится системой $S(M)$ и тоже может быть отнесена к классу больших систем, для которых характерно следующее [10], [11].

1. Цель функционирования, которая определяет степень целенаправленности поведения модели M . В этом случае модели могут быть разделены на:

- одноцелевые, предназначенные для решения одной задачи;
- многоцелевые, позволяющие разрешить или рассмотреть ряд сторон функционирования реального объекта.

2. Сложность, которую, учитывая, что модель M является совокупностью отдельных элементов и связей между ними, можно оценить по общему числу элементов в системе и связей между ними. По разнообразию элементов можно выделить ряд уровней иерархии, отдельные функциональные подсистемы в модели M , ряд входов и выходов и т. д., т. е. понятие сложности может быть идентифицировано по целому ряду признаков.

3. Целостность, указывающая на то, что создаваемая модель M является одной целостной системой $S(M)$, включает в себя большое количество составных частей (элементов), находящихся в сложной взаимосвязи друг с другом.

4. Неопределенность, которая проявляется в системе: по состоянию системы, возможности достижения поставленной цели, методам решения задач, достоверности исходной информации и т. д. Основной характеристикой неопределенности служит такая мера информации, как энтропия, позволяющая в ряде случаев оценить количество управляющей информации, необходимой для достижения заданного состояния системы. При моделировании основная цель – получение требуемого соответствия модели реальному объекту и в этом смысле количество управляющей информации в модели можно также оценить с помощью энтропии и найти то предельное минимальное количество, которое необходимо для получения требуемого результата с заданной достоверностью. Таким образом, понятие неопределенности, характеризующее большую систему, применимо к модели M и является одним из ее основных признаков [11].

5. Поведенческая страта, которая позволяет оценить эффективность достижения системой поставленной цели. В зависимости от наличия случайных воздействий можно различать детерминированные и стохастические системы, по своему поведению – непрерывные и дискретные и т. д. Поведенческая страта рассмотрения системы S позволяет применительно к модели M оценить эффективность построенной модели, а также точность и достоверность полученных при этом результатов. Очевидно, что поведение модели M не обязательно совпадает с поведением реального объекта, причем часто моделирование может быть реализовано на базе иного материального носителя [12].

6. Адаптивность, которая является свойством высокоорганизованной системы. Благодаря адаптивности удается приспособиться к различным внешним возмущающим факторам в широком диапазоне изменения воздействий внешней среды. Применительно к модели существенна возможность ее адаптации в широком спектре возмущающих воздействий, а также изучение поведения модели в изменяющихся условиях, близких к реальным. Надо отметить, что существенным может оказаться вопрос устойчивости модели к различным возмущающим воздействиям. Поскольку модель M – сложная система, весьма важны вопросы, связанные с ее существованием, т. е. вопросы живучести, надежности и т. д. [13], [14].

7. Организационная структура системы моделирования, которая во многом зависит от сложности модели и степени совершенства средств моделирования. Одним из последних достижений в области моделирования можно считать возможность использования имитационных моделей для проведения машинных экспериментов. Необходимы оптимальная организационная структура комплекса технических средств, информационного, математического и программного обеспечений системы моделирования $S'(M)$, оптимальная организация процесса моделирования, поскольку следует обращать особое внимание на время моделирования и точность получаемых результатов.

8. Управляемость модели, вытекающая из необходимости обеспечивать управление со стороны экспериментаторов для получения возможности рассмотрения протекания процесса в различных условиях, имитирующих реальные. В этом смысле наличие многих управляемых параметров и переменных модели в реализованной системе моделирования дает возможность поставить широкий эксперимент и получить обширный спектр результатов [15], [16]. Управляемость системы тесно связана и со степенью автоматизации моделирования. В настоящее время получили применение системы моделирования, отличающиеся высокой степенью автоматизации процесса моделирования, когда наряду с программными средствами управления машинным моделированием используется возможность мультимедийного общения исследователя с процессом моделирования.

9. Возможность развития модели, которая, исходя из современного уровня науки и техники, позволяет создавать мощные системы моделирования $S(M)$ для исследования многих сторон функционирования реального объекта. Однако нельзя при создании системы моделирования ограничиваться только задачами сегодняшнего дня. Необходимо предусматривать возможность развития системы моделирования как по горизонтали в смысле расширения спектра изучаемых функций, так и по вертикали в смысле расширения числа подсистем, т. е. созданная система моделирования должна позволять применять новые современные методы и средства. Естественно, что интеллектуальная система моделирования может функционировать только совместно с коллективом людей, поэтому к ней предъявляют эргономические требования [13], [14], [16].

Цели моделирования систем. Одним из наиболее важных аспектов построения систем моделирования является проблема цели. Любую модель строят в зависимости от цели, которую ставит перед ней исследователь, поэтому одна из основных проблем при моделировании – это проблема целевого назначения. Подобие процесса, протекающего в модели M , реальному процессу является не целью, а условием правильного функционирования модели, и поэтому в качестве цели должна быть поставлена задача изучения какой-либо стороны функционирования объекта.

Для упрощения модели M цели делят на подцели и создают более эффективные виды моделей в зависимости от полученных подцелей моделирования. Можно указать целый ряд примеров целей моделирования в области сложных систем.

Если цель моделирования ясна, то возникает следующая проблема, а именно проблема построения модели M . Построение модели оказывается возможным, если имеется информация или выдвинуты гипотезы относительно структуры, алгоритмов и параметров исследуемого объекта. На основании их изучения осуществляется идентификация объекта. В настоящее время широко применяют различные способы оценки параметров: по мето-

ду наименьших квадратов, по методу максимального правдоподобия, байесовские, марковские оценки [17], [18], [19], [20].

Если модель M построена, то следующей проблемой можно считать проблему работы с ней, т. е. реализацию модели, основные задачи которой – минимизация времени получения конечных результатов и обеспечение их достоверности.

Для правильно построенной модели M характерным является то, что она выявляет лишь те закономерности, которые нужны исследователю, и не рассматривает свойства системы S , не существенные для данного исследования. Следует отметить, что оригинал и модель должны быть одновременно сходны по одним признакам и различны по другим, что позволяет выделить наиболее важные изучаемые свойства. В этом смысле модель выступает как некоторый «заместитель» оригинала, обеспечивающий фиксацию и изучение лишь некоторых свойств реального объекта.

При моделировании существует большая группа научно-технических проблем, к основным из которых можно отнести следующие: идентификацию реальных объектов, выбор вида моделей, построение моделей и их машинную реализацию, взаимодействие исследователя с моделью в ходе машинного эксперимента, проверку правильности полученных в ходе моделирования результатов, выявление основных закономерностей, исследованных в процессе моделирования. В зависимости от объекта моделирования и вида используемой модели эти проблемы могут иметь разную значимость.

В одних случаях наиболее сложной оказывается идентификация, в других – проблема построения формальной структуры объекта. Возможны трудности и при реализации модели, особенно в случае имитационного моделирования больших систем. При этом следует подчеркнуть роль исследователя в процессе моделирования. Постановка задачи, построение содержательной модели реального объекта во многом представляют собой творческий процесс и базируются на опыте исследователя и эвристике.

В этом смысле нет формальных путей выбора оптимального вида модели. Часто отсутствуют формальные методы, позволяющие достаточно точно описать реальный процесс. Поэтому выбор той или иной аналогии, выбор того или иного математического аппарата моделирования полностью основывается на имеющемся опыте исследователя и ошибка исследователя может привести к ошибочным результатам моделирования [3], [4], [10].

Если в ходе моделирования существенное место занимает компьютерный эксперимент, то здесь весьма важна и надежность используемых инструментальных средств (ПК и программных средств), поскольку сбои и отказы программно-технических средств могут приводить к искаженным значениям выходных данных, отображающих протекание процесса. В случае проведения полунатурных экспериментов необходимы специальная аппаратура, специально разработанное математическое и информационное

обеспечение, которые позволяют реализовать диагностику средств моделирования, чтобы отсеять те ошибки в выходной информации, которые вызваны неисправностями функционирующей аппаратуры. В ходе проведения исследования на ПК могут иметь место и ошибочные действия человека-оператора. В этих условиях серьезные задачи стоят в области эргономического обеспечения процесса моделирования.

Всем моделям присуще наличие некоторой структуры (статической или динамической, материальной или идеальной), которая подобна структуре объекта – оригинала. В процессе работы модель выступает в роли относительно самостоятельного квазиобъекта, позволяющего получить при исследовании некоторые знания о самом объекте. Если результаты такого исследования (моделирования) подтверждаются и могут служить основой для прогнозирования в исследуемых объектах, то говорят, что модель адекватна объекту. При этом адекватность модели зависит от цели моделирования и принятых критериев.

1.6. Виды моделей

По сфере применения модели разделяются на: *экономические, социальные, биологические, производственно-технологические, научно-исследовательские* и др.

В каждой области знаний существует своя классификация моделей. Ввиду обширности этого вопроса рассмотрим лишь некоторые общие признаки, по которым эту классификацию можно производить.

С точки зрения целей создания модели делятся на *познавательные и прагматические*.

Познавательные модели создаются на основании изучения и для изучения уже существующих объектов. Они являются формой организации и представления знаний, средством соединения новых знаний с имеющимися. В процессе их создания познавательные модели приближаются к реальности, которую они отображают.

Прагматические модели служат для организации практических действий, на основе сформулированной ранее «идеальной» модели. В этом случае выявляемые расхождения между реальностью и моделью устраняются путем «подгонки» реальности под модель.

Таким образом, имеется два основных отличия рассмотренных выше типов моделей.

Во-первых, познавательные модели строятся на основании отображаемых объектов, в то время как построение прагматических моделей предшествует созданию реальных объектов.

Во-вторых, познавательные модели отображают существующее, а прагматические – не существующее (а лишь желаемое).

По уровню, "глубине" моделирования модели бывают:

– эмпирические - на основе эмпирических фактов, зависимостей;

- теоретические - на основе математических описаний;
- смешанные, полуэмпирические - на основе эмпирических зависимостей и математических описаний.

В зависимости от используемых средств модели делятся на абстрактные или идеальные и материальные, т.е. реальные или вещественные (рис. 1.6).

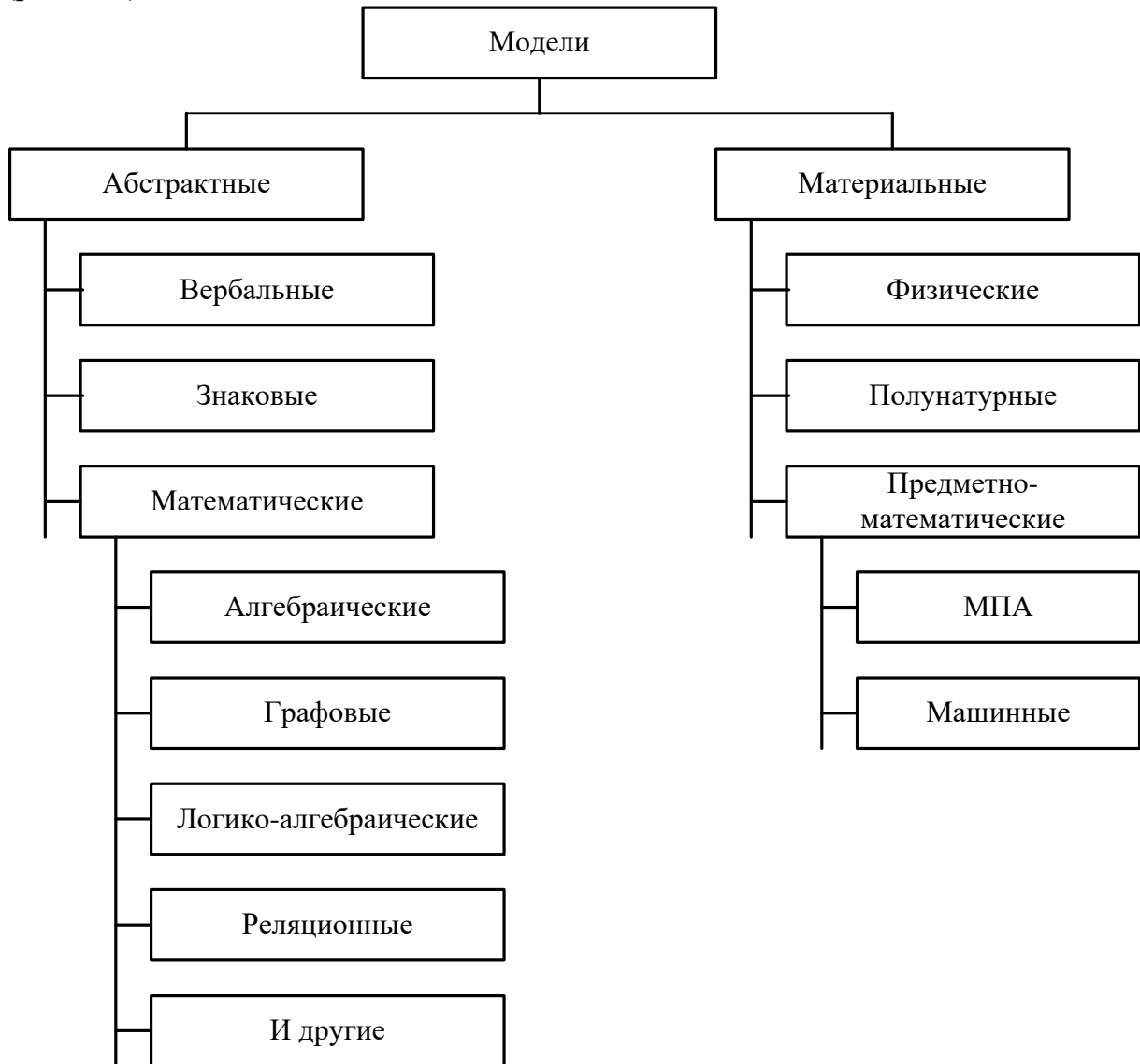


Рис. 1.6. Классификация моделей

Абстрактные модели являются идеальными конструкциями, построенными средствами мышления, сознания. Они бывают двух типов: **мысленные, умозрительные или интуитивные** модели, с одной стороны, и **логико-математические**, с другой.

Класс абстрактных моделей образуют **вербальные** (словесные), **знаковые** и **математические** модели.

Вербальные модели используют **естественные** и **специальные** языки.

Примерами моделей, основанных на использовании **естественных языков**, являются различные инструкции, планы действий и т.д.

Для построения таких моделей не требуется специальной подготовки, но точность таких моделей не велика. На практике их приблизительность, расплывчатость преодолевается с помощью «правильного» понимания (интерпретации).

Дифференциация наук потребовала создания **специализированных языков**, более четких и точных, чем естественный язык.

Применение специальных языков требует дополнительных затрат на их разработку и изучение, но они многократно окупаются за счет преимуществ использования компактных, ясных и точных моделей.

Знаковые модели изучаются в семиотике (науке о знаках). В семиотике выделены три основные группы или аспекта отношений:

синтаксис (греч. «построение», «порядок») – отношение между знаками;

семантика (греч. «обозначение») – отношение между знаками и тем, что они обозначают;

прагматика (греч. «дело», «действие») – отношение между знаками и результатом их использования.

Знаковые модели особенно широко применяются в системах искусственного интеллекта (СИИ).

Математическая модель есть описание какого-либо класса ИУС или другой сложной технической системы, выраженное с помощью математической символики.

Класс математических моделей очень широк, для их изучения необходимо знание соответствующих разделов математики, поэтому ограничимся лишь перечислением некоторых из них.

К математическим моделям относятся: *алгебраические, логико-алгебраические, графовые.*

Материальные модели представляют собой некоторые материальные объекты, отражающие в той или иной мере свойства моделируемых объектов моделирования. По способу отражения этих свойств различают *физические* и *предметно-математические* модели.

Физические модели имеют такую же физическую природу, как оригинал и отличаются в основном размерами (модель локальной сети, модель ПК и др.).

К недостаткам физических моделей относится относительно высокая трудоемкость их изготовления, сложность изменения параметров моделей при переходе от одного варианта к другому.

Предметно-математические модели основаны на совпадении математических выражений, которыми описываются процессы в оригинале и в модели. Для построения этих моделей могут использоваться *методы прямой аналогии* и *машинное моделирование.*

Метод прямой аналогии (МПА) основан на одинаковости математического описания многих электрических, механических, тепловых, гидродинамических и других явлений (например, закон Ома для электрической цепи и закон Фурье для исследования спектрального состава информационного сигнала).

Математический изоморфизм позволяет одни физические системы исследовать с помощью других. Для этого, как правило, используются *электрические модели*.

Преимущество электрических моделей заключается в удобстве изменения параметров, удобстве регистрации характеристик электрических цепей. Однако не для каждой физической задачи существует подходящая электрическая аналогия.

МПА имитирует физическую систему по ее элементам таким образом, что каждому физическому элементу природы соответствует определенный эквивалент в модели. В МПА, как и в физических моделях, происходит расчленение исследуемой системы на составляющие ее физические элементы.

Метод машинного моделирования использует другой подход. Его применение связано с расчленением математического описания исследуемой системы на отдельные математические операции. При этом могут использоваться различные классы вычислительных машин: *аналоговые (АВМ), цифровые (ЦВМ) и аналого-цифровые (АЦВМ)*.

Метод полунатурного моделирования (ПНМ) состоит в том, что на определенном этапе исследования одна часть системы (например, объект управления) замещается моделью, а другая часть (устройство управления) остается натурной. В процессе разработки системы модели отдельных устройств могут заменяться реальными блоками.

Кибернетические модели. Известно, что первые успехи кибернетики связаны с развитием систем автоматического регулирования. В это время сформировались многие фундаментальные понятия, такие, как: *обратная связь, структурно-графовые модели систем, частотные методы расчета и проектирования систем обработки информации*.

Большое влияние на кибернетику оказала *статистическая физика*, вероятностные методы которой были перенесены и развиты в *теории массового обслуживания и надежности, теории игр* и в *теории распознавания образов*. Эти модели образуют класс вероятностных моделей.

Модели дискретной математики содержат модели в виде *графов, автоматов, лингвистических и логических структур*.

По степени универсальности кибернетические модели условно делятся на три типа:

1. модели, не зависящие от предметной области (ПО);
2. модели ориентированные на ПО;
3. модели, настраиваемые на ПО.

Модели не зависящие от ПО так называемые, автономные или синтаксические модели, являются наиболее универсальными. Примерами таких моделей могут служить модели массового обслуживания.

Модели систем массового обслуживания (СМО) являются одной из разновидностей вероятностных моделей и характеризуются случайным законом изменения входных воздействий и параметров системы. Они зависят от типа этого закона и структуры системы, но никак не связаны с природой моделируемого объекта (например, управление пассажиропотоками, обслуживание клиентов, передача пакетов по каналам связи).

Модели ориентированные на ПО применяются в тех случаях, когда реальные системы обладают только им присущей индивидуальностью. В этом случае необходимо, чтобы в модели присутствовала ПО. Кроме того, это связано еще с требованием семантического (логического или смыслового) и прагматического (целевого) управления. Задачи семантической и прагматической обработки информации не могут быть решены без ориентации на предметную область.

Примером такой модели может служить ситуационная модель управления сложной системой (типа аэропорт Пулково). При ситуационном управлении в результате сбора, накопления и формализации знаний об объекте управления и способах управления им вырабатываются правила типа «ситуация – решение по управлению».

Настраиваемые модели занимают промежуточное положение между независимыми и ориентированными на ПО моделями.

Примерами таких моделей могут служить *оболочки экспертных систем*. С помощью таких оболочек можно решать определенные задачи (диагностики, мониторинга, проектирования и т.д.) в различных предметных областях. Перед их применением с помощью специалистов-экспертов остается заполнить базу знаний.

База знаний – семантическая модель, предназначенная для представления в ПК знаний накопленных людьми в определенной предметной области.

Имитационное моделирование используется в тех случаях, когда аналитические методы оказываются малопригодными из-за большой размерности и сложности решаемых задач. Проблема заключается в том, что многочисленные допущения, которые вынужден делать исследователь в этом случае для упрощения задачи, делают аналитическую модель неадекватной реальной системе. Имитационная модель позволяет более полно учитывать особенности реальной системы и в то же время обладает достаточной точностью.

Процесс создания имитационной модели, как правило, включает три стадии:

1. составление вербальной модели системы с помощью естественного и/или специальных языков;

2. составление формализованной модели на основании словесного описания;

3. реализация модели на ПК с помощью специальных программных средств и/или языков моделирования.

Модели систем искусственного интеллекта (СИИ) возникли в связи с желанием и необходимостью повысить «интеллектуальность» разрабатываемых систем управления, наделить машины способностью принимать решения аналогично тому, как это делает человек.

В лабиринтной модели переход от постановки задачи к решению проходит через лабиринт возможных альтернативных путей. Но не всякая творческая задача сводится к такой схеме, кроме того, возможностей может быть слишком много и их не в состоянии за приемлемое время перебрать даже самая быстрая ПК. В этом случае кроме (или вместо) процедуры целенаправленного поиска требуется, использовать другие процедуры.

Ассоциативная модель мышления в качестве основной процедуры использует ассоциативный поиск и ассоциативное рассуждение. Ассоциативное рассуждение позволяет переносить приемы, уже использованные ранее, на текущую ситуацию. Ассоциативные модели в целом не получили широкого распространения, но успешно используются в задачах распознавания образов, в классификационных задачах, для самообучения ПК.

Модели представления знаний (рис. 1.7). Первый подход, называемый **эмпирическим**, основан на изучении принципов организации человеческой памяти и моделировании механизмов решения задач человеком. На основе этого подхода в настоящее время разработаны и получили наибольшую известность следующие модели:

1. **продукционные модели** – модель основанная на правилах, позволяет представить знание в виде предложений типа: «ЕСЛИ условие, ТО действие». Продукционная модель обладает тем недостатком, что при накоплении достаточно большого числа (порядка нескольких сотен) продукций они начинают противоречить друг другу;

2. **сетевые модели (или семантические сети)** – в инженерии знаний под ней подразумевается граф, отображающий смысл целостного образа. Узлы графа соответствуют понятиям и объектам, а дуги – отношениям между объектами. Обладает тем недостатком, что однозначного определения семантической сети в настоящее время отсутствует;

3. **фреймовая модель** – основывается на таком понятии как фрейм (англ. frame – рамка, каркас). Фрейм – структура данных для представления некоторого концептуального объекта. Информация, относящаяся к фрейму, содержится в составляющих его слотах. Слоты могут быть терминальными либо являться сами фреймами, т.о. образуя целую иерархическую сеть.

Условно в группу эмпирического подхода можно включить **нейронные сети и генетические алгоритмы**, относящиеся к бионическому (основано на предположении о том, что если в искусственной системе воспроизвести

структуры и процессы человеческого мозга, то и результаты решения задач такой системой будут подобны результатам, получаемым человеком) направлению искусственного интеллекта. Особенностью моделей этого типа является широкое использование эвристик, что в каждом случае требует доказательства правильности получаемых решений.

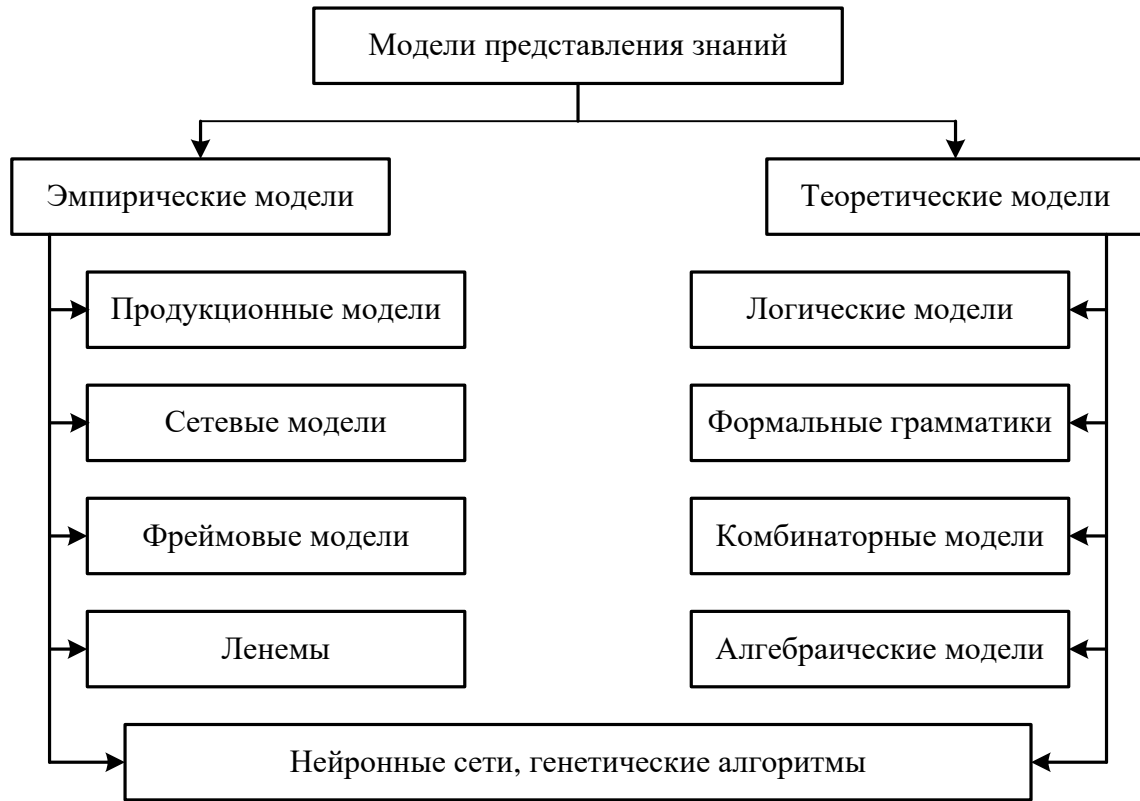


Рис. 1.7. Модели представления знаний

Второй подход можно определить как **теоретически обоснованный**, гарантирующий правильность решений. Он в основном представлен моделями, основанными на формальной логике (исчисление высказываний, исчисление предикатов), формальных грамматиках, комбинаторными моделями, в частности моделями конечных проективных геометрий, теории графов, тензорными и алгебраическими моделями. В рамках этого подхода до настоящего времени удавалось решать только сравнительно простые задачи из узкой предметной области.

Процедурные знания описывают последовательность действий при решении задач (алгоритмы, программы, инструкции и т.д.). **Декларативные знания** имеют в основном повествовательный характер (например, «знание информатики - необходимое условие успешного изучения специальных дисциплин»).

На первых этапах автоматизации обработки данных главную роль играли процедурные знания, доминировали языки программирования процедурного типа. В настоящее время по мере повышения уровня интеллекту-

альности ПК все большую роль при обработке информации играют декларативные знания.

К языкам процедурного типа относятся такие, ставшие традиционными языки, как: Basic, Pascal, Си++.

Языки декларативного типа пока еще не получили столь широкого распространения. В зависимости от используемого математического аппарата они подразделяются на: *функциональные, логические, реляционные*. Рассмотрим некоторые примеры.

Модели формального представления знаний в ПК относятся к одному из трех типов: *логические, сетевые, продукционные*.

Каждая модель представления знаний имеет свои преимущества и недостатки, поэтому разработаны **комбинированные языки представления знаний** (в них чаще всего используются фреймовые и продукционные модели).

1.7. Классификация видов моделирования систем

В качестве одного из признаков классификации видов моделирования можно выбрать степень полноты модели и разделить модели в соответствии с этим признаком на полные, неполные и приближенные. В основе полного моделирования лежит полное подобие, которое проявляется как во времени, так и в пространстве. Для неполного моделирования характерно неполное подобие модели изучаемому объекту. В основе приближенного моделирования лежит приближенное подобие, при котором некоторые стороны функционирования реального объекта не моделируются совсем. Классификация видов моделирования систем S приведена на рис. 1.8 [4].

В зависимости от характера изучаемых процессов в системе S все виды моделирования могут быть разделены на детерминированные и стохастические, статические и динамические, дискретные, непрерывные и дискретно-непрерывные.

Детерминированное моделирование отображает детерминированные процессы, т. е. процессы, в которых предполагается отсутствие всяких случайных воздействий.

Стохастическое моделирование отображает вероятностные процессы и события. В этом случае анализируется ряд реализаций случайного процесса и оцениваются средние характеристики, т. е. набор однородных реализаций.

Статическое моделирование служит для описания поведения объекта в какой-либо момент времени, а **динамическое моделирование** отражает поведение объекта во времени.

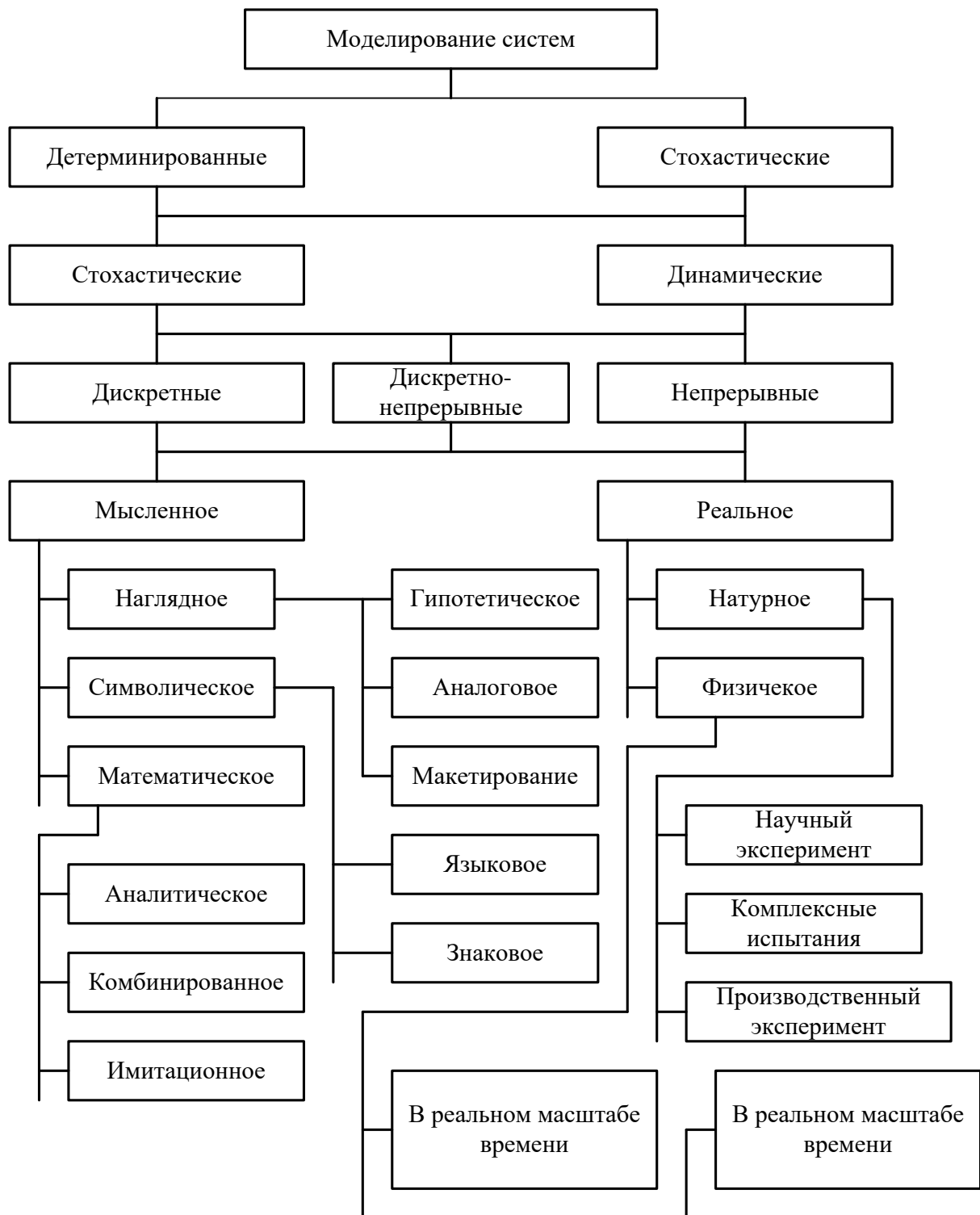


Рис. 1.8. Классификация видов моделирования систем

Дискретное моделирование служит для описания процессов, которые предполагаются дискретными, соответственно непрерывное моделирование позволяет отразить непрерывные процессы в системах.

Дискретно-непрерывное моделирование используется для случаев, когда хотят выделить наличие как дискретных, так и непрерывных процессов.

В зависимости от формы представления объекта (системы можно выделить мысленное и реальное моделирование.

Мысленное моделирование часто является единственным способом моделирования объектов, которые либо практически нереализуемы в заданном интервале времени, либо существуют вне условий, возможных для их физического создания. Мысленное моделирование может быть реализовано в виде наглядного, символического и математического.

При **наглядном моделировании** на базе представлений человека о реальных объектах создаются различные наглядные модели, отображающие явления и процессы, протекающие в объекте. В основе **гипотетического моделирования** исследователем закладывается некоторая гипотеза о закономерностях протекания процесса в реальном объекте, которая отражает уровень знаний исследователя объекте и базируется на причинно-следственных связях между входом и выходом изучаемого объекта. Гипотетическое моделирование используется, когда знаний об объекте недостаточно для построения формальных моделей.

Аналоговое моделирование основывается на применении аналогий различных уровней. Наивысшим уровнем является полная аналогия, имеющая место только для достаточно простых объектов. С усложнением объекта используют аналогии последующих уровней, когда аналоговая модель отображает несколько либо только одну сторону функционирования объекта.

Существенное место при мысленном наглядном моделировании занимает **макетирование**. Мысленный макет может применяться в случаях, когда протекающие в реальном объекте процессы не поддаются физическому моделированию, либо может предшествовать проведению других видов моделирования. В основе построения мысленных макетов также лежат аналогии, однако обычно базирующиеся на причинно-следственных связях между явлениями и процессами в объекте. Если ввести условное обозначение отдельных понятий, т. е. знаки, а также определенные операции между этими знаками, то можно реализовать **знаковое моделирование** и с помощью знаков отображать набор понятий – составлять отдельные цепочки из слов и предложений. Используя операции объединения, пересечения и дополнения теории множеств, можно в отдельных символах дать описание какого-то реального объекта.

В основе **языкового моделирования** лежит некоторый тезаурус. Последний образуется из набора входящих понятий, причем этот набор должен быть фиксированным. Следует отметить, что между тезаурусом и обычным словарем имеются принципиальные различия. Тезаурус – словарь, который очищен от неоднозначности, т. е. в нем каждому слову может соответствовать лишь единственное понятие, хотя в обычном словаре одному слову могут соответствовать несколько понятий.

Символическое моделирование представляет собой искусственный процесс создания логического объекта, который замещает реальный и выражает основные свойства его отношений с помощью определенной системы знаков или символов.

Математическое моделирование. Для исследования характеристик процесса функционирования любой системы математическими методами, включая и машинные, должна быть проведена формализация этого процесса, т. е. построена математическая модель.

Под *математическим моделированием* будем понимать процесс установления соответствия данному реальному объекту некоторого математического объекта, называемого математической моделью, и исследование этой модели, позволяющее получать характеристики рассматриваемого реального объекта. Вид математической модели зависит как от природы реального объекта, так и задач исследования объекта и требуемой достоверности и точности решения этой задачи. Любая математическая модель, как и всякая другая, описывает реальный объект лишь с некоторой степенью приближения к действительности. Математическое моделирование для исследования характеристик процесса функционирования систем можно разделить на аналитическое, имитационное и комбинированное.

Для *аналитического моделирования* характерно то, что процессы функционирования элементов системы записываются в виде некоторых функциональных соотношений (алгебраических, интегро - дифференциальных, конечно-разностных и т. п.) или логических условий. Аналитическая модель может быть исследована следующими методами: а) аналитическим, когда стремятся получить в общем виде явные зависимости для искомых характеристик; б) численным, когда, не умея решать уравнений в общем виде, стремятся получить числовые результаты при конкретных начальных данных; в) качественным, когда, не имея решения в явном виде, можно найти некоторые свойства решения (например, оценить устойчивость решения).

Наиболее полное исследование процесса функционирования системы можно провести, если известны явные зависимости, связывающие искомые характеристики с начальными условиями, параметрами и переменными системы. Однако такие зависимости удается получить только для сравнительно простых систем. При усложнении систем исследование их аналитическим методом наталкивается на значительные трудности, которые часто бывают непреодолимыми. Поэтому, желая использовать аналитический метод, в этом случае идут на существенное упрощение первоначальной модели, чтобы иметь возможность изучить хотя бы общие свойства системы. Такое исследование на упрощенной модели аналитическим методом помогает получить ориентировочные результаты для определения более точных оценок другими методами. Численный метод позволяет исследовать по сравнению с аналитическим методом более широкий класс систем, но при

этом полученные решения носят частный характер. Численный метод особенно эффективен при использовании ПК.

В отдельных случаях исследования системы могут удовлетворить и те выводы, которые можно сделать при использовании качественного метода анализа математической модели. Такие качественные методы широко используются, например, в теории автоматического управления для оценки эффективности различных вариантов систем управления.

В настоящее время распространены методы машинной реализации исследования характеристик процесса функционирования больших систем. Для реализации математической модели на ПК необходимо построить соответствующий моделирующий алгоритм.

При *имитационном моделировании* реализующий модель алгоритм воспроизводит процесс функционирования ИУС во времени, причем имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени, что позволяет по исходным данным получить сведения о состояниях процесса в определенные моменты времени, дающие возможность оценить характеристики ИУС.

Основным преимуществом имитационного моделирования по сравнению с аналитическим является возможность решения более сложных задач. Имитационные модели позволяют достаточно просто учитывать такие факторы, как наличие дискретных и непрерывных элементов, нелинейные характеристики элементов системы, многочисленные случайные воздействия и др., которые часто создают трудности при аналитических исследованиях. В настоящее время имитационное моделирование – наиболее эффективный метод исследования больших систем, а часто и единственный практически доступный метод получения информации о поведении системы, особенно на этапе ее проектирования.

Когда результаты, полученные при воспроизведении на имитационной модели процесса функционирования ИУС, являются реализациями случайных величин и функций, тогда для нахождения характеристик процесса требуется его многократное воспроизведение с последующей статистической обработкой информации и целесообразно в качестве метода машинной реализации имитационной модели использовать метод статистического моделирования. Первоначально был разработан метод статистических испытаний, представляющий собой численный метод, который применялся для моделирования случайных величин и функций, вероятностные характеристики которых совпадали с решениями аналитических задач (такая процедура получила название метода Монте-Карло). Затем этот прием стали применять и для машинной имитации с целью исследования характеристик процессов функционирования систем, подверженных случайным воздействиям, т. е. появился метод статистического моделирования. Таким образом, методом статистического моделирования будем в дальнейшем назы-

вать метод машинной реализации имитационной модели, а методом статистических испытаний (Монте-Карло) – численный метод решения аналитической задачи.

Метод имитационного моделирования позволяет решать задачи анализа больших ИУС, включая задачи оценки: вариантов структуры системы, эффективности различных алгоритмов управления системой, влияния изменения различных параметров системы. Имитационное моделирование может быть положено также в основу структурного, алгоритмического и параметрического синтеза больших систем, когда требуется создать систему, с заданными характеристиками при определенных ограничениях, которая является оптимальной по некоторым критериям оценки эффективности.

При решении задач машинного синтеза систем на основе их имитационных моделей помимо разработки моделирующих алгоритмов для анализа фиксированной системы необходимо также разработать алгоритмы поиска оптимального варианта системы. Далее в методологии машинного моделирования будем различать два основных раздела: статику и динамику, — основным содержанием которых являются соответственно вопросы анализа и синтеза систем, заданных моделирующими алгоритмами.

Комбинированное (аналитико-имитационное) *моделирование* при анализе и синтезе систем позволяет объединить достоинства аналитического и имитационного моделирования. При построении комбинированных моделей проводится предварительная декомпозиция процесса функционирования объекта на составляющие подпроцессы и для тех из них, где это возможно, используются аналитические модели, а для остальных подпроцессов строятся имитационные модели. Такой комбинированный подход позволяет охватить качественно новые классы систем, которые не могут быть исследованы с использованием только аналитического и имитационного моделирования в отдельности.

Другие виды моделирования. При *реальном моделировании* используется возможность исследования различных характеристик либо на реальном объекте целиком, либо на его части. Такие исследования могут проводиться как на объектах, работающих в нормальных режимах, так и при организации специальных режимов для оценки интересующих исследователя характеристик (при других значениях переменных и параметров, в другом масштабе времени и т. д.). Реальное моделирование является наиболее адекватным, но при этом его возможности с учетом особенностей реальных объектов ограничены. Например, проведение реального моделирования АСУ предприятием потребует, во-первых, создания такой АСУ, а во-вторых, проведения экспериментов с управляемым объектом, т. е. предприятием, что в большинстве случаев невозможно. Рассмотрим разновидности реального моделирования.

Натурным моделированием называют проведение исследования на реальном объекте с последующей обработкой результатов эксперимента на

основе теории подобия. При функционировании объекта в соответствии с поставленной целью удастся выявить закономерности протекания реального процесса. Надо отметить, что такие разновидности натурального эксперимента, как *производственный эксперимент* и *комплексные испытания*, обладают высокой степенью достоверности.

С развитием техники и проникновением в глубь процессов, протекающих в реальных системах, возрастает техническая оснащенность современного *научного эксперимента*. Он характеризуется широким использованием средств автоматизации проведения, применением весьма разнообразных средств обработки информации, возможностью вмешательства человека в процесс проведения эксперимента, и в соответствии с этим появилось новое научное направление — автоматизация научных экспериментов.

Отличие эксперимента от реального протекания процесса заключается в том, что в нем могут появиться отдельные критические ситуации и определяться границы устойчивости процесса. В ходе эксперимента вводятся новые факторы и возмущающие воздействия в процессе функционирования объекта. Одна из разновидностей эксперимента – комплексные испытания, которые также можно отнести к натурному моделированию, когда вследствие повторения испытаний изделий выявляются общие закономерности о надежности этих изделий, о характеристиках качества и т. д. В этом случае моделирование осуществляется путем обработки и обобщения сведений, проходящих в группе однородных явлений. Наряду со специально организованными испытаниями возможна реализация натурального моделирования путем обобщения опыта, накопленного в ходе производственного процесса, т. е. можно говорить о производственном эксперименте. Здесь на базе теории подобия обрабатывают статистический материал по производственному процессу и получают его обобщенные характеристики.

Другим видом реального моделирования является *физическое*, отличающееся от натурального тем, что исследование проводится на установках, которые сохраняют природу явлений и обладают физическим подобием. В процессе физического моделирования задаются некоторые характеристики внешней среды и исследуется поведение либо реального объекта, либо его модели при заданных или создаваемых искусственно воздействиях внешней среды. Физическое моделирование может протекать в *реальном и нереальном* (псевдореальном) *масштабах времени*, а также может рассматриваться без учета времени. В последнем случае изучению подлежат так называемые «замороженные» процессы, которые фиксируются в некоторый момент времени. Наибольшая сложность и интерес с точки зрения верности получаемых результатов представляет физическое моделирование в реальном масштабе времени.

С точки зрения математического описания объекта и в зависимости от его характера модели можно разделить на модели аналоговые (непрерывные), цифровые (дискретные) и аналого-цифровые (комбинированные).

Под *аналоговой* моделью понимается модель, которая описывается уравнениями, связывающими непрерывные величины. Под *цифровой* понимают модель, которая описывается уравнениями, связывающими дискретные величины, представленные в цифровом виде. Под *аналого-цифровой* понимается модель, которая может быть описана уравнениями, связывающими непрерывные и дискретные величины.

Особое место в моделировании занимает кибернетическое моделирование, в котором отсутствует непосредственное подобие физических процессов, происходящих в моделях, реальным процессам. В этом случае стремятся отобразить лишь некоторую функцию и рассматривают реальный объект как «черный ящик», имеющий ряд входов и выходов, и моделируют некоторые связи между выходами и входами. Чаще всего при использовании кибернетических моделей проводят анализ поведенческой стороны объекта при различных воздействиях внешней среды. Таким образом, в основе кибернетических моделей лежит отражение некоторых информационных процессов управления, что позволяет оценить поведение реального объекта. Для построения имитационной модели в этом случае необходимо выделить исследуемую функцию реального объекта, попытаться формализовать эту функцию в виде некоторых операторов связи между входом и выходом и воспроизвести на имитационной модели данную функцию, причем на базе совершенно иных математических соотношений и, естественно, иной физической реализации процесса.

В любом случае моделирование должно отвечать следующим требованиям: 1) модели должны быть адекватны соответствующим системам или технологическим задачам; 2) должна обеспечиваться необходимая точность; 3) должно обеспечиваться удобство работы пользователя - специалиста по технологии или по обработке информации (управлению): а) понятный интерфейс управления моделированием; б) достаточная скорость работы; в) наглядность результатов; г) приемлемая стоимость разработки и использования средств моделирования.

1.8. Основные свойства модели системы

Основные свойства любой модели:

- целенаправленность – модель всегда отображает некоторую систему, т.е. имеет цель;
- конечность – модель отображает оригинал лишь в конечном числе его отношений и, кроме того, ресурсы моделирования конечны;
- упрощенность – модель отображает только существенные стороны объекта и, кроме того, должна быть проста для исследования или воспроизведения;
- приближительность – действительность отображается моделью грубо или приблизительно;

- адекватность – модель должна успешно описывать моделируемую систему;
- наглядность, обозримость основных ее свойств и отношений;
- доступность и технологичность для исследования или воспроизведения;
- информативность – модель должна содержать достаточную информацию о системе (в рамках гипотез, принятых при построении модели) и должна давать возможность получить новую информацию;
- сохранение информации, содержащейся в оригинале (с точностью рассматриваемых при построении модели гипотез);
- полнота – в модели должны быть учтены все основные связи и отношения, необходимые для обеспечения цели моделирования;
- устойчивость – модель должна описывать и обеспечивать устойчивое поведение системы, если даже она вначале является неустойчивой;
- целостность – модель реализует некоторую систему (т.е. целое);
- замкнутость – модель учитывает и отображает замкнутую систему необходимых основных гипотез, связей и отношений;
- адаптивность – модель может быть приспособлена к различным входным параметрам, воздействиям окружения;
- управляемость (имитационность) – модель должна иметь хотя бы один параметр, изменениями которого можно имитировать поведение моделируемой системы в различных условиях;
- эволюционируемость – возможность развития моделей (предыдущего уровня).

1.9. Жизненный цикл моделируемой системы

Жизненный цикл моделируемой системы:

- сбор информации об объекте, выдвижение гипотез, предмодельный анализ;
- проектирование структуры и состава моделей (подмоделей);
- построение спецификаций модели, разработка и отладка отдельных подмоделей, сборка модели в целом, идентификация (если это нужно) параметров моделей;
- исследование модели - выбор метода исследования и разработка алгоритма (программы) моделирования;
- исследование адекватности, устойчивости, чувствительности модели;
- оценка средств моделирования (затраченных ресурсов);
- интерпретация, анализ результатов моделирования и установление некоторых причинно-следственных связей в исследуемой системе;
- генерация отчетов и проектных (народно-хозяйственных) решений;

– уточнение, модификация модели, если это необходимо, и возврат к исследуемой системе с новыми знаниями, полученными с помощью модели и моделирования.

Вопросы для самопроверки

1. Что такое модель системы?
2. Что такое метод подобия?
3. Как определяется понятие "моделирование"?
4. В чем заключается метод системного подхода к моделированию?
5. Что называется гипотезой и аналогией в исследовании систем?
6. Как определяется понятие "формализация"?
7. Как определяется понятие "абстракция"?
8. Чем отличается использование метода моделирования при внешнем и внутреннем проектировании систем?
9. В чем заключается главная особенность моделирования?
10. Назовите основные подходы к компьютерному исследованию ТС?
11. Назовите общие проблемы моделирования?
12. Назовите виды моделей?
13. Приведите классификацию видов моделирования?
14. Назовите основные свойства модели ТС?
15. В чем заключается жизненный цикл моделируемой системы?

2. МАТЕМАТИЧЕСКИЕ СХЕМЫ МОДЕЛИРОВАНИЯ СИСТЕМ

2.1. Математические схемы описания информационно-управляющих систем

Исходной информацией при построении математических моделей процессов функционирования систем служат данные о назначении и условиях работы исследуемой (проектируемой) системы S . Эта информация определяет основную цель моделирования, требования к математическим моделям, уровень абстрагирования, выбор математической схемы моделирования.

Понятие *математическая схема* позволяет рассматривать математику как средство перехода от словесного описания к формализованному представлению процесса её функционирования в виде некоторой математической модели.

При пользовании математической схемой в первую очередь исследователя системы должен интересовать вопрос об адекватности отображения в виде конкретных схем реальных процессов в исследуемой системе, а не возможность получения ответа (результата решения) на конкретный вопрос исследования.

Например, представление процесса функционирования ИУС коллективного пользования в виде сети схем массового обслуживания даёт возможность хорошо описать процессы, происходящие в системе, но при

сложных законах входящих потоков и потоков обслуживания не даёт возможности получения результатов в явном виде.

Математическую схему можно определить как звено при переходе от содержательного к формализованному описанию процесса функционирования системы с учётом воздействия внешней среды. Т.е. имеет место цепочка: описательная модель – математическая схема – имитационная модель. Каждая конкретная система S характеризуется набором свойств, под которыми понимаются величины, отображающие поведение моделируемого объекта (реальной системы) и учитываются условия её функционирования во взаимодействии с внешней средой (системой) E .

При построении математической модели системы S необходимо решить вопрос о её полноте. Полнота моделирования регулируется, в основном, выбором границ "Система S – среда E ". Также должна быть решена задача упрощения математической модели, которая помогает выделить основные свойства системы, отбросив второстепенные в плане цели моделирования.

Математическая модель объекта моделирования, т.е. системы S можно представить в виде множества величин, описывающих процесс функционирования реальной системы и образующих в общем случае следующие подмножества:

- совокупность X – входных воздействий на S $x_i \in X, i = 1 \dots n_x$;
- совокупность воздействий внешней среды $v_l \in V, l = 1 \dots n_v$;
- совокупность внутренних (собственных) параметров системы $h_k \in H, k = 1 \dots n_h$;
- совокупность выходных характеристик системы $y_j \in Y, j = 1 \dots n_y$.

В перечисленных множествах можно выделить управляемые и неуправляемые величины. В общем случае X, V, H, Y не пересекаемые множества, содержат как детерминированные так и стохастические составляющие. Входные воздействия E и внутренние параметры S являются независимыми (экзогенными) переменными, $\vec{O}(t); \vec{V}(t); \vec{H}(t)$. Выходные характеристики – зависимые переменные (эндогенные) $\vec{Y}(t)$. Процесс функционирования S описывается оператором F_S :

$$\vec{Y}(t) = F_S(\vec{X}, \vec{V}, \vec{H}, t) \quad (2.1)$$

где $\vec{Y}(t)$ – выходная траектория; F_S – закон функционирования S . F_S может быть функция, функционал, логические условия, алгоритм, таблица или словесное описание правил.

Алгоритм функционирования A_S – метод получения выходных характеристик $\vec{Y}(t)$ с учётом входных воздействий $\vec{O}(t); \vec{V}(t); \vec{H}(t)$. Очевидно один и тот же F_S может быть реализован различными способами, т.е. с помощью множества различных A_S .

Соотношение (2.1) является математическим описанием поведения объекта S моделирования во времени t , т.е. отражает его динамические свойства. Уравнение 2.1 описывает динамическую модель системы S . Для статических условий ММ есть отображения X, V, H в Y , т.е.

$$\vec{Y}(t) = f(\vec{X}, \vec{V}, \vec{H}) \quad (2.2)$$

Соотношения (2.1), (2.2) могут быть заданы формулами, таблицами и т.д.

Также соотношения в ряде случаев могут быть получены через свойства системы в конкретные моменты времени, называемые состояниями.

Состояния системы S характеризуются векторами:

$$\vec{Z}^I(z_1^I, \dots, z_k^I) \text{ и } \vec{Z}^{II}(z_1^{II}, \dots, z_k^{II}), \text{ где } z_1^I = z_1(t^I) \dots z_k^I = z_k(t^I) \text{ в момент } t^I \in (t_0, T)$$

$$z_1^{II} = z_1(t^{II}), \dots, z_k^{II} = z_k(t^{II}) \text{ в момент } t^{II} \in (t_0, T) \text{ и т.д. } k = 1 \dots n_Z. Z_1(t), Z_2(t) \dots$$

$Z_k(t)$ – это координаты точки в k -мерном фазовом пространстве. Каждой реализации процесса будет соответствовать некоторая фазовая траектория.

Совокупность всех возможных значений состояний $\{\vec{Z}\}$ называется пространством состояний объекта моделирования Z , причём $z_k \in Z$.

Состояние системы S в интервале времени $t_0 < t \leq T$ полностью определяется начальными условиями $\vec{Z}^0 = (z_1^0, \dots, z_k^0)$, где $z_1^0 = z_1(t_0) \dots$ входными $\vec{X}(t)$, внутренними параметрами $\vec{H}(t)$ и воздействиями внешней среды $\vec{V}(t)$, которые имели место за промежуток времени $t^* - t_0$ с помощью 2-х векторных уравнений:

$$\vec{Z}(t) = \Phi(\vec{Z}^0, \vec{X}, \vec{V}, \vec{h}, t); \quad (2.3)$$

$$\vec{Y}(t) = F(\vec{Z}, t). \quad (2.4)$$

иначе:

$$\vec{Y}(t) = F(\Phi(\vec{Z}^0, \vec{X}, \vec{V}, \vec{h}, t)). \quad (2.5)$$

Время в модели S может рассматриваться на интервале моделирования (t_0, T) как непрерывное, так и дискретное, т.е. квантованное на отрезке длиной Δt .

Таким образом под математической моделью объекта понимаем конечное множество переменных $\{\vec{X}, \vec{Z}, \vec{h}\}$ вместе с математическими связями между ними и характеристиками \vec{Y} .

Моделирование называется детерминированным, если операторы F, Φ детерминированные, т.е. для конкретного входа выход детерминированный, а также если каждому входному набору параметров соответствует вполне определенный и однозначно определяемый набор выходных параметров; в противном случае - модель недетерминированная, стохастическая (вероятностная).

В практике моделирование объектов в области системного анализа на первичных этапах исследования рациональнее использовать типовые мате-

математические схемы: дифференциальные уравнения, конечные и вероятностные автоматы, СМО, сети Петри и т. д.

Не обладающие такой степенью общности, как модели (2.3), (2.4), типовые математические схемы имеют преимущество простоты и наглядности, но при существенном сужении возможности применения.

В качестве детерминированных моделей, когда при исследовании случайный факт не учитывается, для представления систем, функционирующих в непрерывном времени, используются дифференциальные, интегральные и другие уравнения, а для представления систем, функционирующих в дискретном времени – конечные автоматы и конечно разностные схемы.

В начале стохастических моделей (при учёте случайного фактора) для представления систем с дискретным временем используются вероятностные автоматы, а для представления систем с непрерывным временем – системы массового обслуживания (СМО). Большое практическое значение при исследовании сложных индивидуальных управленческих систем, к которым относятся АСУ, имеют так называемые агрегативные модели.

Агрегативные модели (системы) позволяют описать широкий круг объектов исследования с отображением системного характера этих объектов. Именно при агрегативном описании сложный объект расчленяется на конечное число частей (подсистем), сохраняя при этом связи, обеспечивая взаимодействие частей.

Таким образом, при построении математических моделей процессов функционирования систем можно выделить следующие основные подходы: непрерывно-детерминированный (например, дифференциальные уравнения); дискретно-детерминированный (конечные автоматы); дискретно-стохастический (вероятностные автоматы); непрерывно-стохастический (системы массового обслуживания); обобщенный, или универсальный (агрегативные системы) (рис. 2.1). Подробно о математических схемах моделирования написано в [4].

2.2. Непрерывно-детерминированные модели (*D* – схемы)

Математические схемы такого вида отражают *динамику* процессов, протекающих во времени в системе. Поэтому они называются *D – схемы*. Частным случаем динамических систем являются *системы автоматического управления*.

Линейная автоматическая система описывается линейным дифференциальным уравнением вида

$$a_0 \frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_n y = b_0 \frac{d^m x}{dt^m} + b_1 \frac{d^{m-1} x}{dt^{m-1}} + \dots + b_m x \quad (2.6)$$

где $x(t)$ – задающее воздействие или входная переменная системы; $y(t)$ – состояние системы или выходная переменная; a_i, b_i – коэффициенты; t – время.

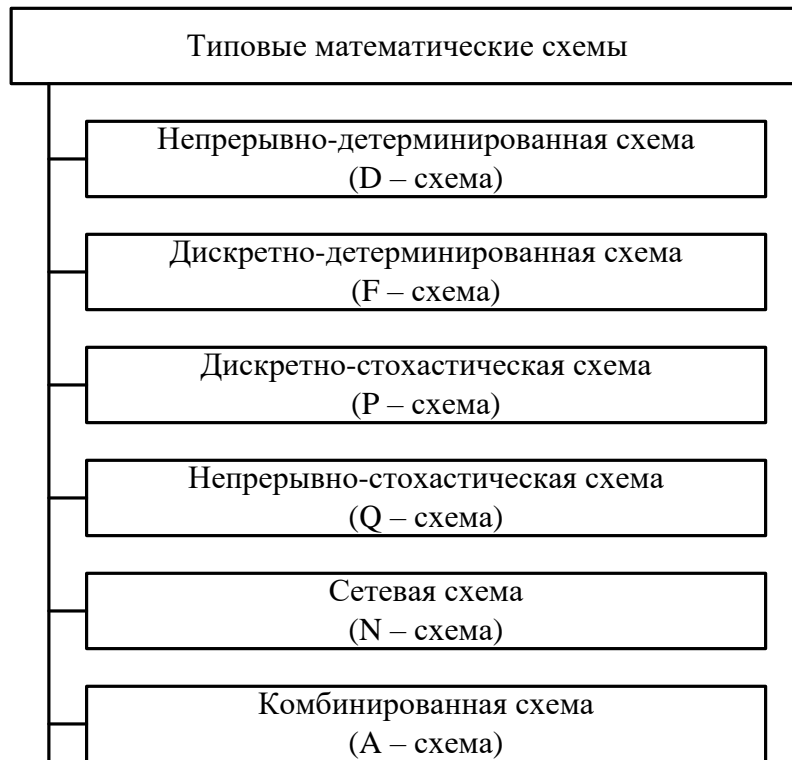


Рис. 2.1. Типовые математические схемы моделирования

На рис.2.2 представлена укрупненная функциональная схема системы автоматического управления, где $h'(t) = x(t) - y(t)$ – сигнал ошибки; $h''(t)$ – управляющее воздействие; $f(t)$ – возмущающее воздействие.

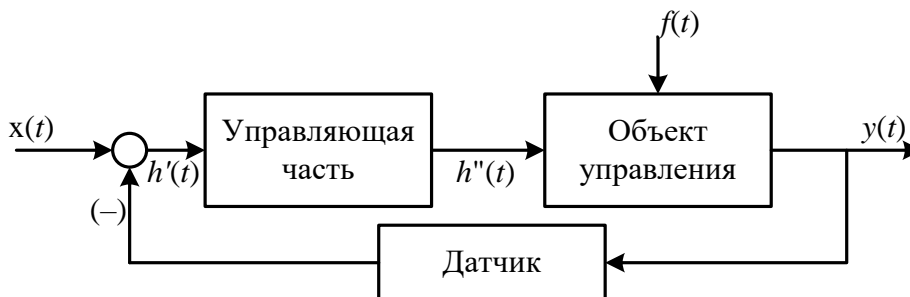


Рис. 2.2. Структура системы автоматического управления

Данная система основана на принципе отрицательной обратной связи, так как для приведения выходной переменной $y(t)$ к ее заданному значению используется информация об отклонении между ними. По ней можно разработать структурную схему и математическую модель в виде передаточной функции или в виде дифференциального уравнения (2.6), в котором

для простоты предполагается, что точки приложения возмущающих воздействий совпадают с входом системы.

Непрерывно - детерминированные схемы (D – схемы) выполняются также и на аналоговых вычислительных машинах (АВМ).

2.3. Дискретно–детерминированные модели (F – схемы)

Основным видом дискретно-детерминированных моделей является *конечный автомат*.

Конечным автоматом называют дискретный преобразователь информации, способный под воздействием входных сигналов переходить из одного состояния в другое и формировать сигналы на выходе. Это автомат *с памятью*. Для организации памяти в описание автомата вводят автоматное время и понятие *состояние автомата*.

Понятие «*состояние*» автомата означает, что выходной сигнал автомата зависит не только от входных сигналов в данный момент времени, но и учитывает входные сигналы, поступающие ранее. Это позволяет устранить время как явную переменную и выразить выходные сигналы как функцию состояний и входных сигналов.

Всякий переход автомата из одного состояния в другое возможен не ранее, чем через дискретный интервал времени. Причем сам переход считается, происходит мгновенно, то есть не учитывают переходные процессы в реальных схемах.

Существует два способа введения автоматного времени по которому автоматы делятся на *синхронные* и *асинхронные*.

В *синхронных* автоматах моменты времени, в которых фиксируются изменения состояний автомата, задаются специальным устройством – генератором синхросигналов. Причем сигналы поступают через равные интервалы времени – Δt . Частота тактового генератора выбирается такой, чтобы любой элемент автомата успел закончить свою работу до появления очередного импульса.

В *асинхронном* автомате моменты перехода автомата из одного состояния в другое заранее не определены и зависят от конкретных событий. В таких автоматах интервал дискретности является переменным.

Также существуют *детерминированные* и *вероятностные* автоматы.

В *детерминированных* автоматах поведение и структура автомата в каждый момент времени однозначно определены текущей входной информацией и состоянием автомата.

В *вероятностных* автоматах они зависят от случайного выбора.

Абстрактно конечный автомат можно представить как математическую схему (F – схему), которая характеризуется шестью видами переменных и функций:

1. конечное множество $x(t)$ входных сигналов (входной алфавит);
2. конечное множество $y(t)$ выходных сигналов (выходной алфавит);

3. конечное множество $z(t)$ внутренних состояний (алфавит состояний);

4. начальное состояние автомата z_0 , $z_0 \in Z$;

5. функция переходов $\phi(z, x)$ автомата из одного состояния в другое;

6. функция выходов $\Psi(z, x)$ автомата.

Абстрактный конечный автомат имеет один вход и один выход. В каждый дискретный момент времени $t = 0, 1, 2, \dots$ F -автомат находится в определенном состоянии $z(t)$ из множества Z – состояний автомата, причем в начальный момент времени $t = 0$ он всегда находится в начальном состоянии $z(0) = z_0$. В момент t , будучи в состоянии $z(t)$, автомат способен воспринять на входном канале сигнал $x(t) \in X$ и выдать на выходном канале сигнал $y(t) = \Psi[z(t), x(t)]$, переходя в состояние $z(t+1) = \phi[z(t), x(t)]$, где $z(t) \in Z, x(t) \in X$.

Абстрактный конечный автомат реализует некоторое отображение множества слов входного алфавита X на множество слов выходного алфавита Y , то есть, если на вход конечного автомата, установленного в начальное состояние z_0 , подавать в некоторой последовательности буквы входного алфавита $x(0), x(1), x(2), \dots$, которые составляют входное слово, то на выходе автомата будут последовательно появляться буквы выходного алфавита $y(0), y(1), y(2), \dots$ образуя выходное слово.

Следовательно, работа конечного автомата происходит по следующей схеме: на каждом t -ом такте на вход автомата, находящегося в состоянии $z(t)$, подается некоторый сигнал $x(t)$, на который автомат реагирует переходом на $(t+1)$ -ом такте в новое состояние $z(t+1)$ и выдачей некоторого выходного сигнала.

В зависимости от способа определения выходного сигнала синхронные абстрактные конечные автоматы подразделяются на два типа:

1. F – автомат первого рода, также называется **автомат Мили**:

$$\begin{cases} z(t+1) = \phi [z(t), x(t)], & t = 0, 1, 2, \dots; \\ y(t) = \Psi [z(t), x(t)], & t = 0, 1, 2, \dots; \end{cases}$$

2. F – автомат второго рода:

$$\begin{cases} z(t+1) = \phi [z(t), x(t)], & t = 0, 1, 2, \dots; \\ y(t) = \Psi [z(t), x(t-1)], & t = 1, 2, 3, \dots; \end{cases}$$

Автомат второго рода, для которого

$$y(t) = \Psi[z(t)], \quad t = 0, 1, 2, \dots$$

называется **автомат Мура** – функция выходов не зависит от входной переменной $x(t)$.

Чтобы задать конечный F – автомат, необходимо описать все элементы множества $F = \langle z, x, y, \phi, \Psi, z_0 \rangle$.

Существует несколько способов задания работы F – автоматов среди которых наибольшее применение нашли табличный, графический и матричный.

2.4. Дискретно – непрерывные модели

Процессы в линейных импульсных и цифровых системах автоматического управления описываются дискретно – разностными уравнениями вида:

$$b_k \Delta^k y(n) + b_{k-1} \Delta^{k-1} y(n) + \dots + b_0 y(n) = x(n), \quad (2.7)$$

где $x(n)$ – решетчатая функция входного сигнала; $y(n)$ – решетчатая функция выходного сигнала, которая определяется решением уравнения (2.7); b_k – постоянные коэффициенты; $\Delta^k y(n)$ – разность k – го порядка; $t = nT$, где nT – n -ый момент времени, T – период дискретности (в выражении (2.7) он условно принят за единицу).

Уравнение (2.7) можно представить в другом виде:

$$a_k y(n-k) + a_{k-1} y(n-k+1) + \dots + a_0 y(n) = x(n) \quad (2.8)$$

Уравнение (2.8) представляет собой рекуррентное соотношение, которое позволяет вычислить любой $(i+1)$ -й член последовательности по значениям предыдущих её членов $i, i-1, \dots$ и значению $x(i+1)$.

Основным математическим аппаратом моделирования цифровых автоматических систем является Z – преобразование, которое базируется на дискретном преобразовании Лапласа. Для этого необходимо найти импульсную передаточную функцию системы, задаться входной переменной и, варьируя параметрами системы, можно найти лучший вариант проектируемой системы.

2.5. Дискретно – стохастические модели (P – схемы)

К дискретно – стохастической модели относится *вероятностный автомат*. В общем, виде вероятностный автомат является дискретным по тактным преобразователем информации с памятью, функционирование которого в каждом такте зависит только от состояния памяти в нем и может быть описано статистически. Поведение автомата зависит от случайного выбора.

Применение схем вероятностных автоматов имеет важное значение для проектирования дискретных систем, в которых проявляется статистически закономерное случайное поведение.

Для P – автомата вводится аналогичное математическое понятие, как и для F – автомата. Рассмотрим множество G , элементами которого являются всевозможные пары (x_i, z_s) , где x_i и z_s элементы входного подмножества X и подмножества состояний Z соответственно. Если существуют две такие функции Φ и Ψ , что с их помощью осуществляется отображение $G \rightarrow Z$ и

$G \rightarrow Y$, то говорят, что $F = \langle Z, X, Y, \phi, \Psi \rangle$ определяет автомат детерминированного типа.

Функция переходов вероятностного автомата определяет не одно конкретное состояние, а распределение вероятностей на множестве состояний (автомат со случайными переходами). Функция выходов также есть распределение вероятностей на множестве выходных сигналов (автомат со случайными выходами).

Для описания вероятностного автомата введем в рассмотрение более общую математическую схему. Пусть Φ – множество всевозможных пар вида (z_k, y_j) , где y_j – элемент выходного подмножества Y . Далее потребуем чтобы любой элемент множества G индуцировал на множестве Φ некоторый закон распределения следующего вида:

$$\begin{array}{ccccccc} \text{элементы из } \Phi & \dots & (z_1, y_1) & (z_1, y_2) & \dots & (z_k, y_{j-1}) & (z_k, y_j) \\ & & & & & & \\ (x_i, z_s) & \dots & b_{11} & b_{12} & \dots & b_{k(j-1)} & b_{kj} \end{array}$$

где b_{kj} – вероятности перехода автомата в состояние z_k и появления на выходе сигнала y_j , если он был в состоянии z_s и на его вход в этот момент времени поступал сигнал x_i .

Число таких распределений, представленных в виде таблиц равно числу элементов множества G . Если обозначить это множество таблиц через B , то тогда четверку элементов $P = \langle Z, X, Y, B \rangle$ называют **вероятностным автоматом** (P – автоматом). При этом $\sum_{k=1}^k \sum_{j=1}^j b_{kj} = 1$.

Частным случаем P – автомата, задаваемого как $P = \langle Z, X, Y, B \rangle$ являются автоматы, у которых либо переход в новое состояние, либо выходной сигнал определяются детерминировано (Z – *детерминированный вероятностный автомат*, Y – *детерминированный вероятностный автомат* соответственно).

Очевидно, что с точки зрения математического аппарата задание Y – детерминированного P – автомата эквивалентно заданию некоторой марковской цепи с конечным множеством состояний. В связи с этим аппарат марковских цепей является основным при использовании P – схем для аналитических расчетов. Подобные P – автоматы используют генераторы марковских последовательностей при построении процессов функционирования систем или воздействий внешней среды.

Марковские последовательности, согласно теореме Маркова, – это последовательность случайных величин, для которой справедливо выражение

$$\lim_{N \rightarrow \infty} \left(\frac{1}{N^2} \right) D \left[\sum_{i=1}^N x_i \right] = 0,$$

где N – количество независимых испытаний; D – дисперсия.

Такие P – автоматы (P – схемы) могут быть использованы для оценки различных характеристик исследуемых систем как для аналитических моделей, так и для имитационных моделей с использованием методов статистического моделирования.

Y – детерминированный P – автомат можно задать двумя таблицами: переходов (таблица 2.1) и выходов (таблица 2.2).

Таблица 2.1

z_k	z_k				
	z_1	z_2	...	z_{k-1}	z_k
z_1	P_{11}	P_{12}	...	$P_{1(k-1)}$	P_{1k}
z_2	P_{21}	P_{22}	...	$P_{2(k-1)}$	P_{2k}
...
z_k	P_{k1}	P_{k2}	...	$P_{k(k-1)}$	P_{kk}

Таблица 2.2

$Z...$	z_1	z_2	...	z_{k-1}	z_k
$Y...$	y_{i1}	y_{i2}	...	$y_{i(k-1)}$	y_{ik}

где P_{ij} – вероятность перехода P – автомата из состояния z_i в состояние z_j , при этом $\sum_{j=1}^k P_{ij} = 1$.

Таблицу 2.1 можно представить в виде квадратной матрицы размерности $k \times k$. Такую таблицу будем называть *матрицей переходных вероятностей* или просто *матрицей переходов P – автомата*, которую можно представить в компактной форме:

$$P_p = \begin{vmatrix} P_{11} & P_{12} & \dots & P_{1k} \\ P_{21} & P_{22} & \dots & P_{2k} \\ \cdot & \cdot & \cdot & \cdot \\ P_{k1} & P_{k2} & \dots & P_{kk} \end{vmatrix}$$

Для описания Y – детерминированного P –автомата необходимо задать начальное распределение вероятностей вида:

$$\begin{matrix} Z... & z_1 & z_2 & \dots & z_{k-1} & z_k \\ D... & d_1 & d_2 & \dots & d_{k-1} & d_k \end{matrix}$$

где d_k – вероятность того, что в начале работы P – автомат находится в состоянии z_k , при этом $\sum_{k=1}^K d_k = 1$.

И так, до начала работы P – автомат находится в состоянии z_0 и в начальный (нулевой) такт времени меняет состояние в соответствии с распределением D . После этого смена состояний автомата определяется матрицей переходов P . С учетом z_0 размерность матрицы P_p следует увеличить

до $(k + 1) \times (k + 1)$, при этом первая строка матрицы будет $(d_0, d_1, d_2, \dots, d_k)$, а первый столбец будет нулевым.

Пример [4]. Y - детерминированный P - автомат задан таблицей переходов:

Таблица 2.3

$$P_p = \begin{vmatrix} 0 & 0,50 & 0 & 0 & 0,50 \\ 0 & 0 & 0 & 1,00 & 0 \\ 0 & 0 & 0,75 & 0 & 0,25 \\ 0 & 0 & 0,40 & 0 & 0,60 \\ 0 & 1,00 & 0 & 0 & 0 \end{vmatrix}$$

и таблицей выходов

Таблица 2.4

Z	z_0	z_1	z_2	z_3	z_4
Y	0	0	1	1	0

С учетом таблицы 2.3 граф переходов вероятностного автомата представлен на рис. 2.3.

Требуется оценить суммарные финальные вероятности пребывания этого автомата в состоянии z_2 и z_3 , т.е. когда на выходе автомата появляются единицы.

При аналитическом подходе можно использовать известные соотношения из теории марковских цепей и получить систему уравнений для определения финальных вероятностей. Причем начальное состояние можно не учитывать в виду того, что начальное распределение не оказывает влияние на значения финальных вероятностей.

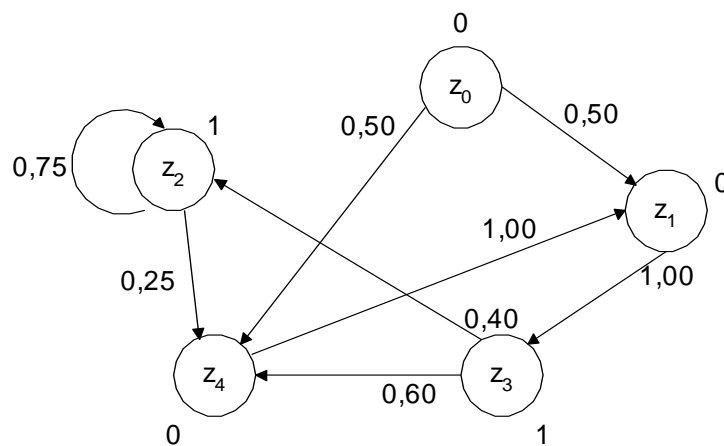


Рис. 2.3. Граф переходов

Тогда таблица 2.3 примет вид:

$$\vec{C} = \vec{C} \begin{vmatrix} 0 & 0 & 1,00 & 0 \\ 0 & 0,75 & 0 & 0,25 \\ 0 & 0,40 & 0 & 0,60 \\ 1,00 & 0 & 0 & 0 \end{vmatrix};$$

где $\vec{C} = (c_k) = (c_1, c_2, c_3, c_4)$ – финальная вероятность пребывания Y – детерминированного P – автомата в состоянии z_k .

В результате получаем систему уравнений:

$$\begin{cases} c_1 = c_4 \\ c_2 = 0,75c_2 + 0,40c_3 \\ c_3 = c_1 \\ c_4 = 0,25c_2 + 0,60c_3 \end{cases} \quad (2.9)$$

К данной системе следует добавить условие нормировки:

$$c_1 + c_2 + c_3 + c_4 = 1 \quad (2.10)$$

Теперь решая систему уравнений (2.9) совместно с (2.10), получаем:

$$c_1 = \frac{5}{23}; c_2 = \frac{8}{23}; c_3 = \frac{5}{23}; c_4 = \frac{5}{23}.$$

Таким образом, при бесконечной работе заданного автомата на его выходе будет формироваться двоичная последовательность с вероятностью появления единицы, равной: $c_2 + c_3 = \frac{13}{23} = 0,5652$.

Кроме аналитических моделей в виде P – схем можно применять и имитационные модели, реализуемые, например, методом статистического моделирования.

2.6. Непрерывно–стохастические модели (Q – схемы)

Такие модели рассмотрим на примере использования в качестве типовых математических схем систем массового обслуживания, которые называют Q – *схемами*. Такие Q – схемы применяются при формализации процессов функционирования систем, которые по своей сути являются процессами *обслуживания*.

К *процессам обслуживания* можно отнести: потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ПК от удаленных терминалов сети ПК. Характерным признаком для функционирования таких систем или сетей является случайное появление заявок на обслуживание. Причем в любом элементарном акте обслуживания можно выделить две основные составляющие: ожидание обслуживания и, собственно, сам процесс обслуживания заявки. Представим это в виде некоторого i -го прибора обслуживания $П_i$ (рис. 2.4), состоящего из накопителя заявок

H_i , в котором может находиться одновременно λ_i заявок; K_i – канал обслуживания заявок.

На каждый элемент прибора Π_i поступают потоки событий, в накопитель H_i поток заявок λ_i , на канал K_i – поток обслуживания I_i .

Потоки событий могут быть *однородными*, если он характеризуется только последовательностью поступления этих событий ($0 \leq t_1 \leq t_2 \leq \dots \leq t_n$), или *неоднородными*, если он характеризуется набором признаков события, например таким набором признаков: источник заявок, наличие приоритета, возможность обслуживания тем или иным типом канала и т.п. [4].

Обычно при моделировании различных систем применительно к каналу K_i можно считать, что поток заявок λ_i на входе K_i образует подмножество неуправляемых переменных, а поток обслуживания I_i – образует подмножество управляемых переменных.

Те заявки, которые по различным причинам не обслуживаются каналом K_i , образуют выходной поток Y_i .

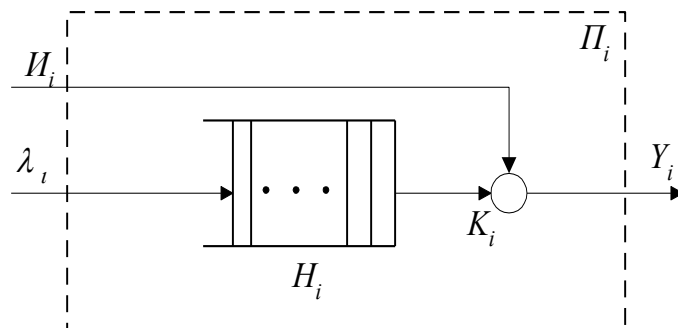


Рис. 2.4. Прибор обслуживания

Пример. Построение информационно-вычислительных сетей коллективного пользования. Данная модель строится с точки зрения обработки информации и принятия решений по её обработке, хранению и передачи [4].

Блок-схема информационной сети показана на рис. 2.5, где выделены уровни базовой (магистральной) сети, реализующей обмен информацией между центрами коллективного пользования абонентской (терминальной) сети, обеспечивающей обмен информацией между пользователями и сетью ПК.

Основными элементами сети являются узлы коммутации (центры) потоков, осуществляющие все основные операции по управлению сетью, включая коммутацию и маршрутизацию потоков сообщений (пакетов); концентраторы, обеспечивающие сопряжение входных низкоскоростных каналов связи с выходным высокоскоростным каналом; терминалы, выпол-

няющие функции организации доступа пользователя к ресурсам сети и функции по локальной обработке информации; каналы связи, реализующие обмен информацией между узлами сети (узлами коммутации, концентраторами, терминалами) с требуемым качеством.

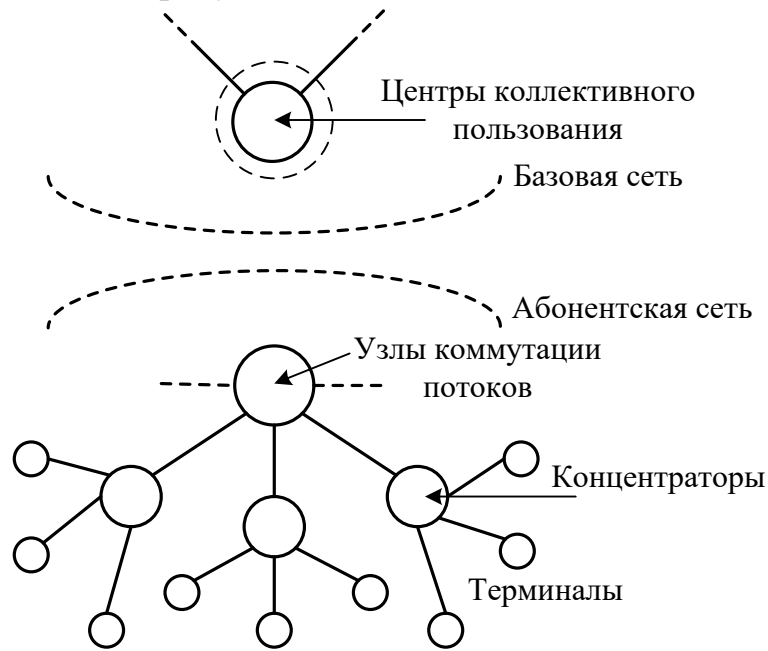


Рис. 2.5. Структурная схема информационной сети

На рис. 2.6 изображена структурная схема информационно-вычислительной сети (ИВС) на уровне взаимодействия терминальной и базовой сети.

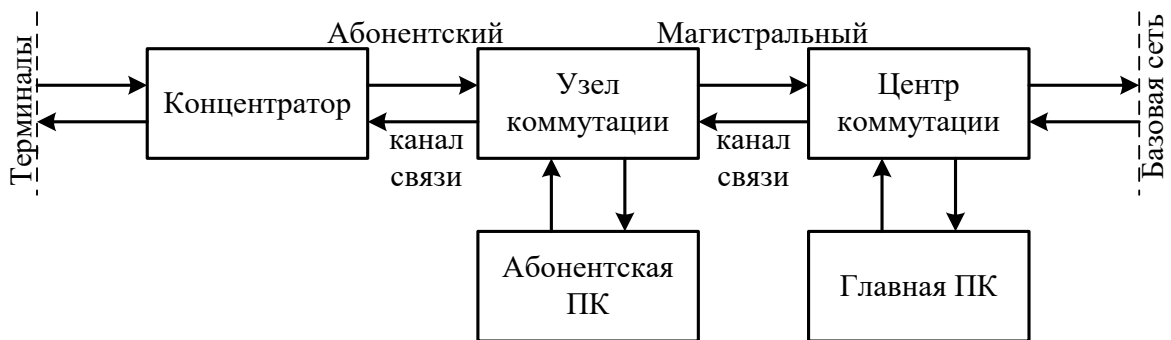


Рис. 2.6. Структура взаимодействия терминальной и базовой сети

Информация поступает с терминалов пользователей в виде сообщений длиной q бит с интенсивностью λ сообщений в секунду. Абонентская ПК подключена к узлу коммутации.

Она обрабатывает поступающую информацию от концентраторов с производительностью h бит/сек. Мультиплексные каналы ПК обслуживают по k – терминалов каждый, передавая данные со скоростью V бит/сек.

Главная ПК имеет суммарную производительность H бит/с при наличии n -мультиплексных каналов. В том случае, если производительность абонентской ПК недостаточна, то она через узлы коммутации и магистральный канал связи с пропускной способностью C бит/с подключается посредством центра коммутации к главной ПК. Предполагается, что процессы коммутации выполняются мгновенно.

Укрупненная блок-схема моделирующего алгоритма приведена [4].

В процессе проектирования информационно-вычислительной сети необходимо оценить среднее время обработки информации T_0 и вероятность отказа в выполнении работ $P_{от}$ в двух случаях:

- в случае работы только с абонентской ПК (автономный режим);
- в случае подключения к одной из ПК сети коллективного пользования.

Процесс функционирования данного фрагмента ИВС может быть представлен в виде Q - схемы, имеющей два параллельных канала обслуживания, а также связи, управляющей блокировкой. Структура такой Q - схемы представлена на рисунке 2.7 (где I – источник; $\hat{E}1, \hat{E}2$ – каналы обслуживания).

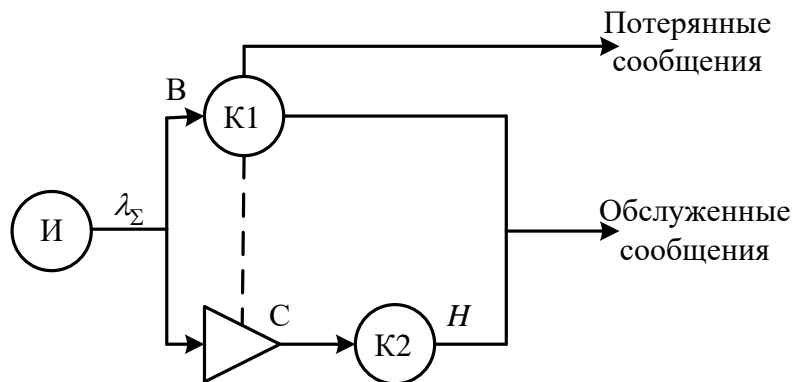


Рис. 2.7. Q - схема.

В качестве выходного потока источника I рассматривается суммарный поток сообщений от терминалов, т.е. поток на выходе концентратора.

Выходной поток Q – схемы составляет обслуживание сообщения при работе каналов $\hat{E}1, \hat{E}2$ и потерянные сообщения при отключении (блокировке) канала $\hat{E}2$. В такой постановке решения этой задачи аналитическим методом в явном виде с использованием теории массового обслуживания не представляется возможным из-за стохастического характера работы механизма блокировки. Поэтому для оценки интересующих характеристик можно воспользоваться методом имитационного моделирования.

Рассмотрим моделирующий алгоритм процесса функционирования фрагмента ИВС. Для этого запишем:

– *эндогенные переменные* (зависимые переменные) *от внутренних свойств* сети: T_0 – среднее время обслуживания сообщений; $P_{от}$ – вероятность отказа в обслуживании;

– *экзогенные переменные от внешних свойств* сети: $\lambda_\Sigma = knq\lambda$ – интенсивность входного потока сообщений; h – производительность абонентской ПК; H – суммарная производительность главных ПК сети; B – пропускная способность селекторных каналов ПК; C – пропускная способность магистрального канала связи.

Вид *уравнения модели* зависит от режима работы схемы:

а) при заблокированном канале

$$T_0 = 2B + q/h,$$

$$P_{\text{от}} = (\lambda_\Sigma - h)/h \text{ при } \lambda_\Sigma > h;$$

б) при работе каналов \hat{E}_1, \hat{E}_2

$$T_0 = 2q/B + q/h + h(2q/C + q/H) \frac{1}{\lambda_\Sigma - L},$$

$$P_{\text{от}} = 0,$$

где L – величина, связанная с длиной сообщения.

К Q – схемам относятся непрерывно-стохастические модели. В качестве таких типовых схем могут выступать системы массового обслуживания.

2.7. Стохастические минимаксные модели

Эти модели можно отнести к оптимальным стохастическим моделям [28].

Во многих случаях при построении модели не все условия заранее известны. Эффективность нахождения модели здесь будет зависеть от трех факторов: 1) заданных условий x_1, x_2, \dots, x_n ; 2) неизвестных условий y_1, y_2, \dots, y_k ; 3) зависящих от нас факторов u_1, u_2, \dots, u_m , которые необходимо найти.

Показатель эффективности решения такой задачи имеет вид:

$$W = W(x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_k; u_1, u_2, \dots, u_m)$$

Наличие неизвестных факторов y_i переводит задачу оптимизации в задачу о выборе решения в условиях неопределенности. Задача становится чрезвычайно сложной.

Особенно задача осложняется для случаев, когда величины y_i не обладают статистической устойчивостью, то есть неизвестные факторы y_i нельзя изучить с помощью статистических методов. Их законы распределения либо не могут быть получены, либо вовсе не существуют.

В этих случаях рассматриваются комбинации всевозможных значений $Y: \alpha = (y_1^*, y_2^*, \dots, y_k^*)$ таким образом, чтобы получить как «наилучшее», так и «наихудшее» сочетания значений переменных y_i .

Тогда в качестве критерия оптимизации рассматривается:

$$W = W(x, \alpha, u)$$

$$x = x(x_1, x_2, \dots, x_n), \alpha = \alpha(\alpha_1, \alpha_2, \dots, \alpha_k)$$

$$u = u(u_1, u_2, \dots, u_m)$$

2.8. Особенности моделирования инфокоммуникационных систем

Проведение моделирования инфокоммуникационных систем (ИНФС) включает: 1. разработку структуры модели ИНФС; 2. построение сетевой модели; 3. построение функциональных моделей элементов ИНФС; 4. анализ структурно-функциональной модели ИНФС, заключающийся в определении вероятностно-временных характеристик (ВВХ) системы в зависимости от параметров отдельных функциональных моделей и организации связей между ними.

При моделировании ИНФС можно выделить три группы задач, решаемых на этапе разработки структуры модели ИНФС.

Первая группа задач связана с разработкой структурной схемы модели ИНФС с учетом ее развития, динамики потоков и живучести, выбором производительности и алгоритмов функционирования ее элементов.

Вторая группа задач связана с реализацией технологии взаимодействия, которая учитывает механизм обмена информацией (система протоколов).

Третья группа задач связана с задачами управления взаимодействием на уровне транспортной сети. При решении этих задач необходимо разработать адекватные модели управления коммутацией, маршрутизацией, управлением потоками, реконfigurацией сети.

При моделировании ИНФС в качестве критериев используются: 1. ВВХ доставки сообщения; 2. стоимость сети; 3. структурная надежность либо живучесть сети. Причем стоимость сети выступает часто в виде ограничения. Названные задачи моделирования ИНФС связаны между собой. Комплексное моделирование, учитывающее решение всех задач является сложной задачей. На практике сложились два подхода: 1. когда решаются отдельные задачи моделирования и затем объединяются в общей процедуре моделирования; 2. Производится декомпозиция (вертикальная и /или горизонтальная) сложной ИНФС.

На первом этапе моделирования ИНФС эвристическим методом определяется исходная структура модели ИНФС. Далее проводится анализ структуры модели ИНФС с привлечением имитационного моделирования. Определяются основные характеристики ИНФС (в том числе надежность и временные). Определяется область допустимых значений нагрузки,

при которых обеспечивается требуемое время передачи информации. Сложность моделирования заключается в том, что на различных участках сети должны использоваться различные модели транспортировки информации (пакетов). Это обуславливает необходимость оценки ВВХ доставки между всеми корреспондирующими парами узлов при заданных моделях структуры сети, моделей алгоритмов маршрутизации и т. д.

Значения исходных параметров модели ИНФС так же определяется с учетом вида трафика с тем, чтобы обеспечить низкие вероятности потерь пакетов данных и речи и незначительный разброс задержек пакетов при передаче видеoinформации.

Практически любая стратегия маршрутизации представляет собой совокупность маршрутных таблиц (МТ), которые распределяются по узлам коммутации и указывают, как в зависимости от конечного адресата должен быть распределен по выходным линиям трафик, поступающий в данный узел. В качестве такого средства можно использовать имитационную модель установления соединения.

Концепция моделирования установления соединения. Процесс установления соединения представляет собой посылку и прохождение вызова от источника к адресату по путям, составляющим направление обмена между источником и адресатом. Время установления соединения жестко ограничивается, как и вероятность превышения этого времени. Ограничивается также вероятность не установления соединения за допустимое время.

При имитации доставки вызова каждый прогон модели в случае доставки вызова до адресата дает реализации случайных величин, по значениям которых подсчитывается время доставки вызова.

Входными данными для моделирования установления соединения являются: 1) структурная и потоковая метрики; 2) гамаки кратчайших путей; 3) полученные на их основе значения вероятностей полной занятости транзитов (ветвей) и ограничения, при которых должно выполняться соединение; 4) максимально допустимое число транзитов в соединении; 5) допустимое время установления соединения (либо доставки вызова); 6) времена продвижения вызова через транзит и подключения коммутационного поля; 7) число рестартов (возвратов вызова для передачи его по обходным путям, время выполнения рестарта).

Выходными данными модели являются: 1) статистики, позволяющие оценить вероятность установления соединения (доставки вызова адресату) за время, не превышающее допустимого; 2) средние и среднеквадратические значения, характеризующие процесс установления соединения.

Факт доставки вызова устанавливается с помощью имитации процесса доставки. При этом ввиду возможности - рестартов - возвращения вызова на предыдущие узлы - число транзитов, пройденных вызовом при его доставке к адресату, оказывается случайным. Поэтому для моделирования установления соединения применяются стохастические методы.

При моделировании рассмотренной задачи может быть применен метод расслоения и метод генерации, предложенный Б. И. Плаксом.

При построении полной имитационной модели сети в целях оценки ее ВВХ возникает задача снижения размерности модели.

Формально уменьшение размерности может быть сведено к выделению на структуре сети некоторого множества классов объектов с последующей их заменой соответствующими моделями. Актуально при этом обеспечить однотипность моделей для представителей различных кластеров. Важно также, чтобы эти модели с достаточной полнотой отражали специфику процессов, определяющих ВВХ сети.

Для имитации взаимодействия с другими направлениями обмена информацией, т.е. учета влияния потоков, циркулирующих по сети и являющихся транзитными в выделенном ВК, строится вероятностный эквивалент не рассматриваемой части сети в виде генератора транзитных (фоновых) потоков. Как теоретическую основу данного подхода можно рассматривать метод расслоенной выборки.

Декомпозиция сети на подмножестве путей обмена информацией. Специфика декомпозиции ИНФС заключается в том, чтобы при моделировании выделенной части воспроизвести взаимодействие процессов, протекающих во всей сети и влияющих на характеристики выделенной части. Это требование реализуется при декомпозиции ИНФС на классы направлений обмена информацией. Разбиение множества путей проводится с помощью двухэтапной классификации объектов.

Имитационная модель виртуального канала. Имитационная модель виртуального канала позволяет проводить детальное исследование процесса передачи и обработки пакетов на любом выделенном пути (маршруте сети), имитировать любые режимы функционирования ВК.

Основной задачей анализа модели ВК является определение ВВХ доставки пакетов по исследуемому пути при заданных характеристиках элементов пути, параметрах входящих потоков, алгоритмах обработки и дисциплинах обслуживших очередей.

ВК представляет собой коммутационный канал, обеспечивающий транспортировку пакетов между двумя портами сети, т. е. является некоторым маршрутом в сети.

Особенностью имитационной модели ВК является отображение потоков, циркулирующих по сети и влияющих на процесс прохождения пакетов по выделенному (моделируемому) пути.

Узлы в моделирующей программе ВК представляются тремя основными устройствами: первое обеспечивает перенос пакетов между узлами (модуль *Removpaket*); второе обеспечивает коммутацию пакетов - это СМО с одним обслуживающим прибором, одной входной очередью и более чем одной выходной очередью (модуль *УК*); третье обеспечивает передачу пакетов следующему узлу - это СМО с одной входной очередью (модуль *КС*).

Наряду с разработкой алгоритмов EVT, обеспечивающих ускорение моделирования ВК методом Монте-Карло, ускорить процесс моделирования можно и за счет рационального построения программной реализации имитационной модели ВК.

Обычно в имитационной модели ВК используется механизм системного времени, в основе которого лежит принцип особых моментов, - на очередном шаге выполняется то особое состояние, время наступления которого ближайшее к текущему моменту.

Модель состоит из программных модулей, реализующих алгоритмы обслуживания моделирования, и модулей, реализующих алгоритмы обработки имитируемых физических действий.

Вопросы для самопроверки

1. Что называется математической схемой?
2. Какое математическое описание применяется при описании непрерывно-детерминированных систем?
3. Какое математическое описание применяется при описании дискретно-детерминированных систем?
4. Какое математическое описание применяется при описании непрерывно-стохастических систем?
5. Какое математическое описание применяется при описании СМО?
6. Как описываются комбинированные модели?
7. Что является экзогенными и эндогенными переменными в модели объекта?
8. Что называется законом функционирования системы?
9. Что понимается под алгоритмом функционирования?
10. Что называется статической и динамической моделями объекта?
11. Какие типовые схемы используются при моделировании сложных систем и их элементов?
12. Каковы условия и особенности использования при разработке моделей систем различных типовых схем?

3. ФОРМАЛИЗАЦИЯ И АЛГОРИТМИЗАЦИЯ ПРОЦЕССОВ ФУНКЦИОНИРОВАНИЯ СИСТЕМ

3.1. Этапы моделирования информационно-управляющих систем

Рассмотрим основные этапы моделирования ИУС (рис. 3.1 [4]), к числу которых относятся: построение концептуальной модели системы и ее формализация; алгоритмизация модели системы и ее машинная реализация; получение и интерпретация результатов моделирования системы.

Перечислим эти под этапы: 1.1 – постановка задачи машинного моделирования системы; 1.2 – анализ задачи моделирования системы; 1.3 – определение требований к исходной информации об объекте моделирования и организация ее сбора; 1.4 – выдвижение гипотез и принятие предположений; 1.5 – определение параметров и переменных модели; 1.6 – уста-

новление основного содержания модели; 1.7 – обоснование критериев оценки эффективности системы; 1.8 – определение процедур аппроксимации; 1.9 – описание концептуальной модели системы; 1.10 – проверка достоверности концептуальной модели; 1.11 – составление технической документации по первому этапу; 2.1 – построение логической схемы модели; 2.2 – получение математических соотношений; 2.3 – проверка достоверности модели системы; 2.4 – выбор инструментальных средств для моделирования; 2.5 – составление плана выполнения работ по программированию; 2.6 – спецификация и построение схемы программы; 2.7 – верификация и проверка достоверности схемы программы; 2.8 – проведение программирования модели; 2.9 – проверка достоверности программы; 2.10 – составление технической документации по второму этапу; 3.1 – планирование машинного эксперимента с моделью системы; 3.2 – определение требований к вычислительным средствам; 3.3 – проведение рабочих расчетов; 3.4 – анализ результатов моделирования системы; 3.5 – представление результатов моделирования; 3.6 – интерпретация результатов моделирования; 3.7 – подведение итогов моделирования и выдача рекомендаций; 3.8 – составление технической документации по третьему этапу.

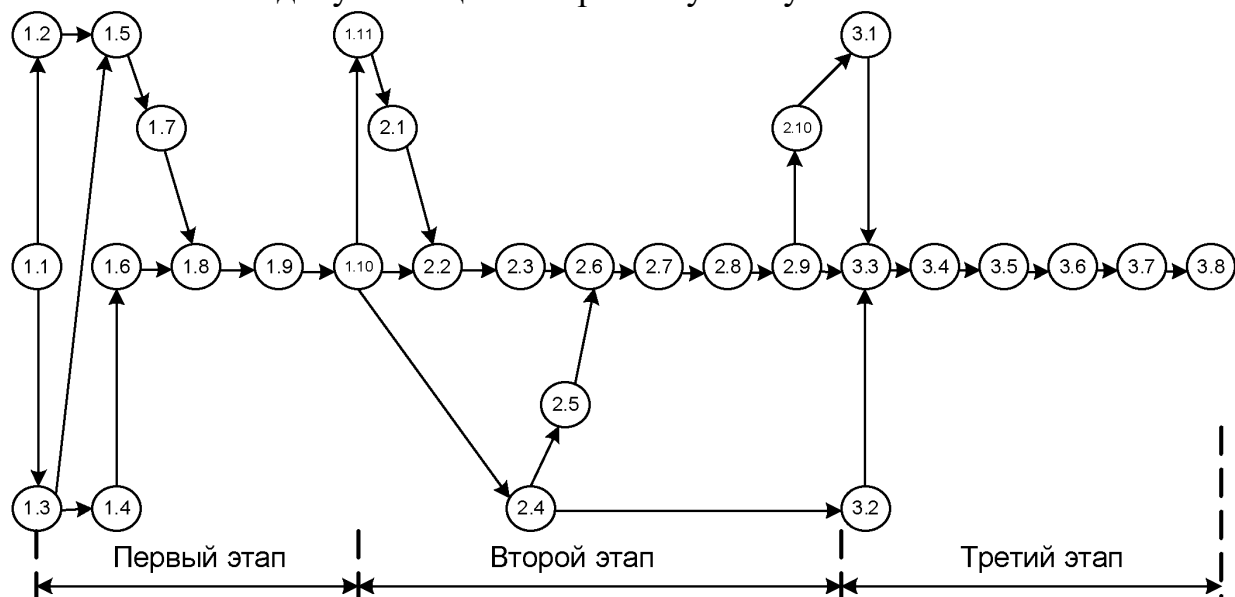


Рис. 3.1. Взаимосвязь этапов моделирования ИУС

Как видно из рис. 3.1, процесс моделирования ИУС сводится к выполнению перечисленных подэтапов, сгруппированных в виде трех этапов. На этапе построения концептуальной модели и ее формализации проводится исследование моделируемого объекта с точки зрения выделения основных составляющих процесса его функционирования, определяются необходимые аппроксимации и получается обобщенная схема модели ИУС, которая преобразуется в машинную модель на втором этапе моделирования путем последовательной алгоритмизации и программирования модели. Последний третий этап моделирования системы сводится к проведению рабочих

расчетов на ПК с использованием выбранных инструментальных средств, получению и интерпретации результатов моделирования системы с учетом воздействия внешней среды. При построении модели ИУС и ее машинной реализации при получении новой информации возможен пересмотр ранее принятых решений, т. е. процесс моделирования является итерационным. Рассмотрим содержание каждого из этапов и их под этапов [4]:

1.1. Постановка задачи машинного моделирования системы. На этом подэтапе основное внимание уделяется таким вопросам, как: а) признание существования задачи и необходимости машинного моделирования; б) выбор методики решения задачи с учетом имеющихся ресурсов; в) определение масштаба задачи и возможности разбиения ее на подзадачи.

Необходимо также ответить на вопрос о приоритетности решения различных подзадач, оценить эффективность возможных математических методов и программно-технических средств их решения.

1.2. Анализ задачи моделирования системы. На рассматриваемом втором этапе основная работа сводится именно к проведению анализа, включая: а) выбор критериев оценки эффективности процесса функционирования системы; б) определение эндогенных и экзогенных переменных модели; в) выбор возможных методов идентификации; г) выполнение предварительного анализа содержания второго этапа алгоритмизации модели системы и ее машинной реализации; д) выполнение предварительного анализа содержания третьего этапа получения и интерпретации результатов моделирования системы.

1.3. Определение требований к исходной информации об объекте моделирования и организация ее сбора. На этом подэтапе проводится: а) выбор необходимой информации о системе и внешней среде; б) подготовка априорных данных; в) анализ имеющихся экспериментальных данных; г) выбор методов и средств предварительной обработки информации о системе.

При этом необходимо помнить, что именно от качества исходной информации об объекте моделирования существенно зависят как адекватность модели, так и достоверность результатов моделирования.

1.4. Выдвижение гипотез и принятие предположений. При выдвижении гипотез и принятии предположений учитываются следующие факторы: а) объем имеющейся информации для решения задач; б) подзадачи, для которых информация недостаточна; в) ограничения на ресурсы времени для решения задачи; г) ожидаемые результаты моделирования.

1.5. Определение параметров и переменных модели. Описание каждого параметра и переменной должно даваться в следующей форме: а) определение и краткая характеристика; б) символ обозначения и единица измерения; в) диапазон изменения; г) место применения в модели.

1.6. Установление основного содержания модели. На этом подэтапе определяется основное содержание модели и выбирается метод построения

модели системы, которые разрабатываются на основе принятых гипотез и предположений. При этом учитываются следующие особенности: а) формулировка задачи моделирования системы; б) структура системы и алгоритмы ее поведения, воздействия внешней среды; в) возможные методы и средства решения задачи моделирования.

1.7. Обоснование критериев оценки эффективности системы. Для оценки качества процесса функционирования моделируемой системы необходимо выбрать некоторую совокупность критериев оценки эффективности, т. е. в математической постановке задача сводится к получению соотношения для оценки эффективности как функции параметров и переменных системы.

1.8. Определение процедур аппроксимации. Для аппроксимации реальных процессов, протекающих в системе S , обычно используются три вида процедур: а) детерминированную; б) вероятностную; в) определения средних значений.

1.9. Описание концептуальной модели системы. На этом подэтапе построения модели системы: а) описывается концептуальная модель в абстрактных терминах и понятиях; б) дается описание модели с использованием типовых математических схем; в) принимаются окончательно гипотезы и предположения; г) обосновывается выбор процедуры аппроксимации реальных процессов при построении модели.

1.10. Проверка достоверности концептуальной модели. Проверка достоверности концептуальной модели M_k должна включать: а) проверку замысла модели; б) оценку достоверности исходной информации; в) рассмотрение постановки задачи моделирования; г) анализ принятых аппроксимаций; д) исследование гипотез и предположений.

1.11. Составление технической документации по первому этапу. В конце этапа построения концептуальной модели и ее формализации составляется технический отчет по этапу, который включает в себя: а) подробную постановку задачи моделирования системы; б) анализ задачи моделирования системы; в) критерии оценки эффективности системы; г) параметры и переменные модели системы; д) гипотезы и предположения, принятые при построении модели; е) описание модели в абстрактных терминах и понятиях; ж) описание ожидаемых результатов моделирования системы.

2.1. Построение логической схемы модели. Рекомендуется строить модель по блочному принципу, т. е. в виде некоторой совокупности стандартных блоков. Построение модели систем из таких блоков обеспечивает необходимую гибкость в процессе ее эксплуатации, особенно на стадии машинной отладки. При построении блочной модели проводится разбиение процесса функционирования системы на отдельные достаточно автономные подпроцессы.

2.2. Получение математических соотношений. Схема машинной модели должна представлять собой полное отражение заложенной в модели концепции и иметь: а) описание всех блоков модели с их наименованиями; б) единую систему обозначений и нумерацию блоков; в) отражение логики модели процесса функционирования системы; г) задание математических соотношений в явном виде.

2.3. Проверка достоверности модели системы. При этом проверяются: а) возможность решения поставленной задачи; б) точность отражения замысла в логической схеме; в) полнота логической схемы модели; г) правильность используемых математических соотношений.

Только после того, как разработчик убеждается путем соответствующей проверки в правильности всех этих положений, можно считать, что имеется логическая схема модели системы, пригодная для дальнейшей работы по реализации модели на ПК.

2.4. Выбор инструментальных средств для моделирования. Вопрос о выборе ПК сводится к обеспечению следующих требований: а) наличие необходимых программных и технических средств; б) доступность выбранной ПК для разработчика модели; в) обеспечение всех этапов реализации модели; г) возможность своевременного получения результатов.

2.5. Составление плана выполнения работ по программированию. План при использовании универсальной ПК должен включать в себя: а) выбор языка (системы) программирования модели; б) указание типа ПК и необходимых для моделирования устройств; в) оценку примерного объема необходимой оперативной и внешней памяти; г) ориентировочные затраты машинного времени на моделирование; д) предполагаемые затраты времени на программирование и отладку программы на ПК.

2.6. Спецификация и построение схемы программы. Спецификация программы – формализованное представление требований, предъявляемых к программе, которые должны быть удовлетворены при ее разработке, а также описание задачи, условия и эффекта действия без указания способа его достижения. Наличие логической блок-схемы модели позволяет построить схему программы, которая должна отражать: а) разбиение модели на блоки, подблоки и т. д.; б) особенности программирования модели; в) проведение необходимых изменений; г) возможности тестирования программы; д) оценку затрат машинного времени; е) форму представления входных и выходных данных.

2.7. Верификация и проверка достоверности схемы программы. Верификация программы – доказательство того, что поведение программы соответствует спецификации на программу. Эта проверка является второй на этапе машинной реализации модели системы. При этом проводится проверка соответствия каждой операции, представленной в схеме программы, аналогичной ей операции в логической схеме модели.

2.8. Проведение программирования модели. Если имеется адекватная схема программы, то программирование представляет собой работу только для программиста без участия и помощи со стороны разработчика модели. При использовании пакетов прикладных программ моделирования проводится не посредственная генерация рабочих программ для моделирования конкретного объекта, т. е. программирование модели реализуется в автоматизированном режиме.

2.9. Проверка достоверности программы. Эта последняя проверка на этапе машинной реализации модели, которую необходимо проводить: а) обратным переводом программы в исходную схему; б) проверкой отдельных частей программы при решении различных тестовых задач; в) объединением всех частей программы и проверкой ее в целом на контрольном примере моделирования варианта системы.

2.10. Составление технической документации по второму этапу. Для завершения этапа машинной реализации модели необходимо составить техническую документацию, содержащую: а) логическую схему модели и ее описание; б) адекватную схему программы и принятые обозначения; в) полный текст программы; г) перечень входных и выходных величин с пояснениями; д) инструкцию по работе с программой; е) оценку затрат машинного времени на моделирование с указанием требуемых ресурсов ПК.

3.1. Планирование машинного эксперимента с моделью системы. Перед выполнением рабочих расчетов на ПК должен быть составлен план проведения эксперимента с указанием комбинаций переменных и параметров, для которых должно проводиться моделирование системы. Планирование машинного эксперимента призвано дать в итоге максимальный объем необходимой информации об объекте моделирования при минимальных затратах машинных ресурсов. При этом различают стратегическое и тактическое планирование машинного эксперимента.

3.2. Определение требований к вычислительным средствам. Необходимо сформулировать требования по времени использования вычислительных средств, т. е. составить график работы на одной или нескольких ПК, а также указать те внешние устройства ПК, которые потребуются при моделировании. При этом также рационально оценить, исходя из требуемых ресурсов, возможность использования для реализации конкретной модели персональной ПК или локальной вычислительной сети.

3.3. Проведение рабочих расчетов. После составления программы модели и плана проведения машинного эксперимента с моделью системы можно приступить к рабочим расчетам на ПК, которые обычно включают в себя: а) подготовку наборов исходных данных для ввода в ПК; б) проверку исходных данных, подготовленных для ввода; в) проведение расчетов на ПК; г) получение выходных данных, т. е. результатов моделирования.

Проведение машинного моделирования рационально выполнять в два этапа: контрольные, а затем рабочие расчеты.

3.4. Анализ результатов моделирования системы. Чтобы эффективно проанализировать выходные данные, полученные в результате расчетов на ПК, необходимо знать, что делать с результатами рабочих расчетов и как их интерпретировать. Эти задачи могут быть решены на основании предварительного анализа на двух первых этапах моделирования системы. Планирование машинного эксперимента с моделью позволяет вывести необходимое количество выходных данных и определить метод их анализа.

3.5. Представление результатов моделирования. Целесообразно в каждом конкретном случае выбрать наиболее подходящую форму, так как это существенно влияет на эффективность их дальнейшего употребления заказчиком. В большинстве случаев наиболее простой формой считаются таблицы, хотя графики более наглядно иллюстрируют результаты моделирования системы. При диалоговых режимах моделирования наиболее рациональными средствами оперативного отображения результатов моделирования являются средства мультимедиа технологии.

3.6. Интерпретация результатов моделирования. Получив и проанализировав результаты моделирования, их нужно интерпретировать по отношению к моделируемому объекту, т. е. системе. Основное содержание этого подэтапа - переход от информации, полученной в результате машинного эксперимента с моделью, к информации применительно к объекту моделирования, на основании которой и будут делаться выводы относительно характеристик процесса функционирования исследуемой системы.

3.7. Подведение итогов моделирования и выдача рекомендаций. При подведении итогов моделирования должны быть отмечены главные особенности, полученные в соответствии с планом эксперимента над моделью результатов, проведена проверка гипотез и предположений и сделаны выводы на основании этих результатов. Все это позволяет сформулировать рекомендации по практическому использованию результатов моделирования, на пример на этапе проектирования системы.

3.8. Составление технической документации по третьему этапу. Эта документация должна включать в себя: а) план проведения машинного эксперимента; б) наборы исходных данных для моделирования; в) результаты моделирования системы; г) анализ и оценку результатов моделирования; д) выводы по полученным результатам моделирования; е) указание путей дальнейшего совершенствования машинной модели и возможных областей ее приложения.

Полный комплект документации по моделированию конкретной ИУС на ПК должен содержать техническую документацию по каждому из трех рассмотренных этапов.

3.2. Способы описания алгоритмов

Для строгого задания различных структур данных и алгоритмов их обработки требуется иметь такую систему формальных обозначений и правил, чтобы смысл всякого используемого предписания трактовался точно и однозначно. Соответствующие системы правил называют языками описаний.

К средствам описания алгоритмов относятся следующие основные способы их представления: словесный; графический; псевдокоды; программный. На практике используются также и другой способ описания: табличный (таблицы переключений (таблицы истинности); таблицы автоматов; циклограммы работы; таблицы решений).

3.2.1. Словесный способ представления алгоритмов

Словесный способ записи алгоритмов представляет собой последовательное описание основных этапов обработки данных и задается в произвольном изложении на естественном языке.

Способ основан на использовании общепринятых средств общения между людьми и с точки зрения написания трудностей не представляет. Такой способ записи удобно использовать на начальном этапе алгоритмизации задачи. К недостаткам словесного способа записи можно отнести следующее: 1) полное подробное словесное описание алгоритма получается очень громоздким; 2) естественный язык допускает неоднозначность толкования отдельных инструкций; 3) при переходе к этапу программирования требуется дополнительная работа по формализации алгоритма, так как словесное описание может быть понятно человеку, но "непонятно" ПК. Поэтому словесный способ записи алгоритмов не имеет широкого распространения.

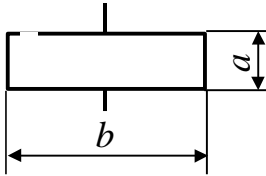
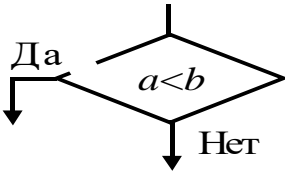
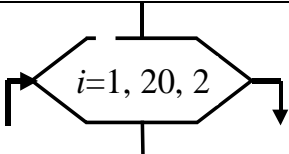
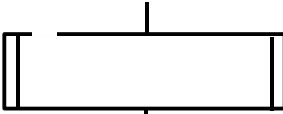

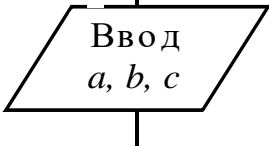

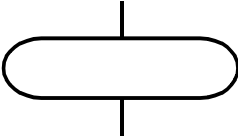
3.2.2. Графический способ представления алгоритмов

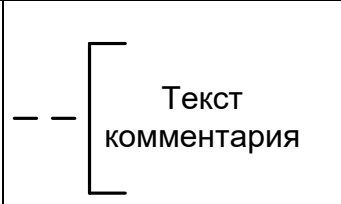
Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным. При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.

Такое графическое представление называется схемой алгоритма или блок-схемой. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т. п.) соответствует геометрическая фигура, представленная в виде блочного символа. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий. Для начертания этих схем используется набор символов, определяемых ГОСТ 19.701–90 (ИСО 5807 – 85) "Единая система программной документации". В таблице 3.1 приведены наиболее часто употребляемые символы.

Символ "Процесс" применяется для обозначения одного или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединить в один блок. Представление отдельных операций достаточно свободно.

Таблица 3.1

Название символа	Обозначение	Пояснение
Процесс		Вычислительное действие или последовательность вычислительных действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме
Документ		Вывод, печать результатов на бумаге
Ввод - вывод		Ввод – вывод данных в общем виде
Соединитель		Разрыв линий потока
Пуск, останов		Начало, конец, останов, вход и выход в подпрограммах

Комментарий		Пояснения, содержание под-программ, формулы
-------------	---	---

Например, для обозначения вычислений можно использовать математические выражения, для пересылок данных – стрелки, для других действий – пояснения на естественном языке. В зависимости от уровня детализации схемы пояснения на естественном языке могут быть более или менее подробными. Метод блок-схем так же, как и алгоритмический язык (псевдокод), независим от специфики языков программирования, поэтому в описаниях операторов не следует использовать резервированные слова и символы языков программирования, а также применять имена данных, образованные в соответствии с синтаксическими правилами этих языков.

Символ "Решение" используется для обозначения переходов управления по условию. В каждом блоке решения должны быть указаны вопрос, решение, условие или сравнение, которые он определяет.

Стрелки, выходящие из блока решения, должны быть помечены соответствующими ответами (например, ДА, НЕТ) так, чтобы были учтены все возможные ответы.

Символ "Модификация" используется для выполнения операций, меняющих команды или группы команд, изменяющих программу (например, для организации циклических конструкций).

Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и правило изменения значения параметра для каждого повторения. Блок размещается в начале циклической конструкции, для управления которой он используется, даже в том случае, если изменение параметра и проверка условий окончания цикла при реализации алгоритма производится не в начале, а в конце цикла.

Линии переходов используются для обозначения порядка выполнения действий. Для улучшения наглядности следует придерживаться стандартных правил изображения линий передач управления – сверху вниз и слева направо. Если необходимо показать передачу управления снизу вверх или справа налево, то направление следует отметить стрелкой.

Символ "Предопределенный процесс" используется для указания обращений к вспомогательным алгоритмам, выделенным автономно, в виде некоторого модуля; для обращений к библиотечным подпрограммам; для обозначения части алгоритма, независимой от основной схемы управления; для обозначения определенной части алгоритма, которая будет кодироваться вместе со всем алгоритмом, но в документации представлена отдельной схемой. Если такая часть алгоритма представляет собой итерационный процесс, то в соответствующий ей блок вызова необходимо включить опи-

сания условий окончания цикла. По мнению некоторых специалистов, использование более одной схемы для одного алгоритма затрудняет его понимание. Однако практика показывает, что удобнее всего применять схемы алгоритмов, разбитые в соответствии с уровнями абстракции.

Символ "Документ" предназначен для ввода – вывода данных, носителем которых служит бумага.

Символ "Ввод - вывод" используется для преобразования данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). Отдельным логическим устройствам ПК или отдельным функциям обмена соответствуют определенные блочные символы. В каждом из них указываются тип устройства или файла данных, тип информации, участвующий в обмене, а также вид операции обмена.

Символ "Соединитель" используется в том случае, когда схема алгоритма разделяется на автономные части, особенно если она не умещается на одном листе, или когда необходимо избежать излишних пересечений линий переходов. Применение соединителей не должно нарушать структурности при изображении схем.

Символ "Пуск - останов" используется для обозначения начала, конца, прерывания процесса обработки данных или выполнения программы.

Символ "Комментарий" позволяет включать в схемы алгоритмов пояснения к функциональным блокам. Частое использование комментариев нежелательно, так как это усложняет (загромождает) схему, делает ее менее наглядной. Однако некоторые обозначения переменных, принятые допущения или назначение отдельных алгоритмов требуют пояснительных записей.

Правила выполнения схем. Для облегчения вычерчивания и нахождения на схеме символов рекомендуется поле листа разбивать на зоны. Размеры зон устанавливаются с учетом минимальных размеров символов, изображенных на данном листе. Допускается один символ размещать в двух и более зонах, если размер символа превышает размер зоны.

Координаты зоны проставляют: по горизонтали – арабскими цифрами слева направо в верхней части листа; по вертикали – прописными буквами латинского алфавита сверху вниз в левой части листа.

Координаты зон в виде сочетания букв и цифр присваивают символам, вписанным в поля этих зон, например А1, А2, А3, В1, В2, В3 и т. д.

В пределах одной схемы, при выполнении ее от руки, допускается применять не более двух смежных размеров ряда чисел, кратных 5. Для ускорения выполнения схем от руки рекомендуется использовать бланки с контуром прямоугольника внутри каждой зоны. Контурные линии не должны воспроизводиться при изготовлении копии.

Расположение символов на схеме должно соответствовать требованиям ГОСТ 19.701–90 (ИСО 5807 – 85).

Исключение составляют обязательные символы "Линия потока",

"Канал связи", "Комментарий" и рекомендуемые символы "Межстраничный соединитель", "Транспортирование носителей", "Материальный поток".

Линии потока должны быть параллельны линиям внешней рамки схемы. Направления линии потока сверху вниз и слева направо принимают за основные и, если линии потока не имеют изломов, стрелками можно не обозначать. В остальных случаях направление линии потока обозначать стрелкой обязательно.

Расстояние между параллельными линиями потока должно быть не менее 3 мм, между остальными символами схемы – не менее 5 мм. Записи внутри символа или рядом с ним должны выполняться машинописью с одним интервалом или чертежным шрифтом. Записи внутри символа или рядом с ним должны быть краткими. Сокращения слов и аббревиатуры, за исключением установленных государственными стандартами, должны быть расшифрованы в нижней части поля схемы или в документе, к которому эта схема относится.

Для удобства детализации программы должны быть использованы символы "Процесс", "Решение", "Модификация", "Ввод – вывод" и "Пуск – останов".

Записи внутри символа должны быть представлены так, чтобы их можно было читать слева направо и сверху вниз, независимо от направления потока. В схеме символу может быть присвоен идентификатор, который должен помещаться слева над символом (например, для ссылки в других частях документации).

В схемах допускается краткая информация о символе (описание, уточнение или другие перекрестные ссылки для более полного понимания функции данной части системы). Описание символа должно помещаться справа над символом

Соотношение геометрических элементов символов. Размер a должен выбираться из ряда 10, 15, 20 мм. Допускается увеличивать размер a на число, кратное 5. Размер b равен $1,5 \cdot a$. При ручном выполнении схем алгоритмов и программ для символов, представленных в таблице 3.1, допускается устанавливать b равным $2 \cdot a$.

При выполнении условных графических обозначений автоматизированным методом размеры геометрических элементов символов округляются до значений, определяемых техническими возможностями используемых устройств.

3.2.3. Графические способы описания алгоритмов работы информационных систем (промышленных систем)

К графическим способам описания алгоритмов работы информационных систем (промышленных систем) относятся также:

– Диаграммы. Применяются для описания зависимости входных и вы-

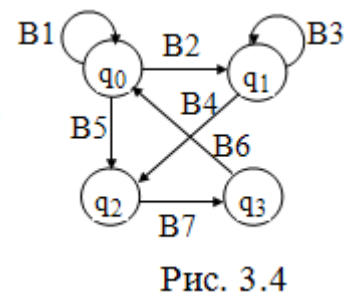
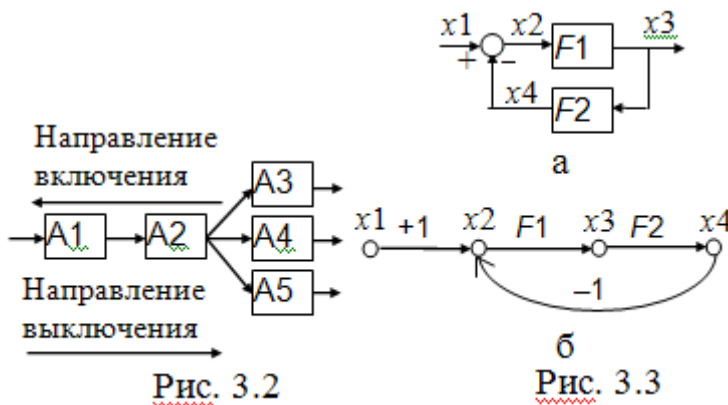
ходных переменных состояния друг от друга или от времени.

– Диаграммы последовательности включений и пошаговые диаграммы перемещений. Применяются для описания линейных (неразветвленных) дискретных процессов, например работы простых станков. На оси ординат указываются команды включения и состояния системы. На оси абсцисс – такты работы или реальный масштаб времени.

– Диаграммы работы. Применяются для описания работы контакторных схем управления.

– Схемы блокировки. Схемы блокировки (рис. 3.2) служат для изображения предусмотренной технологией взаимосвязанности процессов включения и выключения.

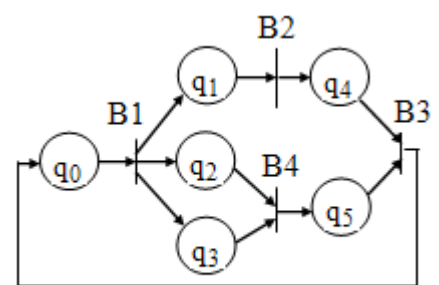
– Структурные схемы (рис. 3.3, а) и сигнальные графы (рис. 3.3, б). Применяются для изображения структуры и описания функционирования преимущественно непрерывных систем. Переход от одной формы описания к другой очень прост. Передаточные функции (передачи) F_i отдельных функциональных блоков записываются в структурной схеме внутри соответствующих прямоугольников, а в направленном графе – на его ветвях (ребрах). Переменным (сигналам) x_i в структурной схеме соответствуют линии, соединяющие блоки, а в графе – его узлы (вершины).



– Автоматные графы (рис. 3.4). Применяются для описания дискретных состояний системы и возможных переходов между ними. Узлы изображают различные возможные состояния q_i , а ветви со стрелками – переходы. Рядом с каждой ветвью записывается условие V_j , которое вызывает переход между соответствующими состояниями.

– Сети Петри (рис. 3.5). Сети Петри являются направленными графами с двумя видами узлов, а именно с узлами для изображения состояний q_i (кружки) и узлами для изображения переходов (вертикальные штрихи) между состояниями.

Переход осуществляется, если состояния,



находящиеся перед символом перехода, помечены и наступает событие, вызывающее переход. Например, имеет место переход от q_0 к q_1 , q_2 и q_3 , если имеется q_0 и выполнено условие $B1$. Сети Петри особенно удобны для изображения параллельно происходящих взаимосвязанных процессов.

– Графы последовательного выполнения программы. Пригодны для записи задач управления и для описания поведения релейных систем управления. Изображают зависящую от каких-либо условий последовательность состояний системы q_i . Положение (0 или 1) конкретных функциональных элементов ($Q1, Q2, Y1, Y2, Y3$), соответствующие некоторым характерным состояниям системы, указываются в отдельной таблице. В приведенном примере (рис. 3.6): как только $S1 = 1$, система совершает переход из состояния q_0 в q_1 ; если $S2 = 1$, то осуществляется переход в состояние q_2 , для $S2 = 0$ – в состояние q_0 и т. д.

– Схемы работы (рис. 3.7). Очень удобно использовать для описания линейно протекающих процессов и работы соответствующих систем управления.

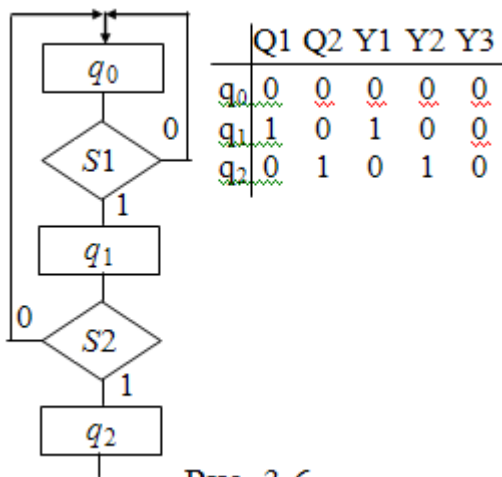


Рис. 3.6

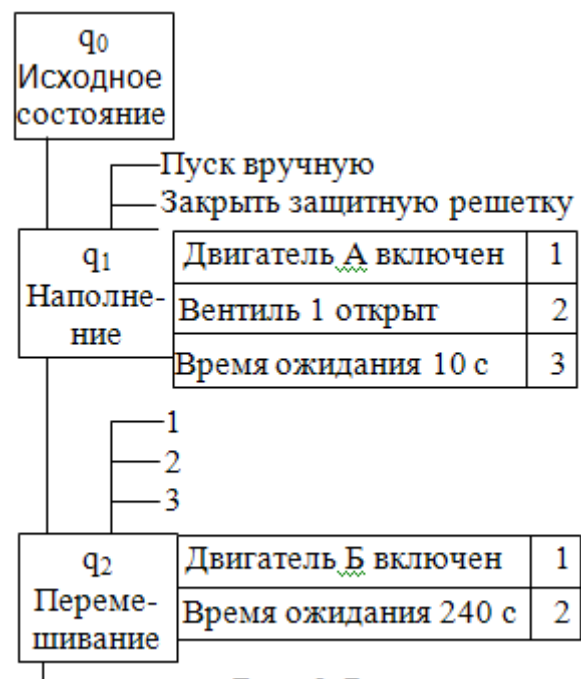


Рис. 3.7

Изображается зависящая от появления определенных событий последовательность отдельных шагов или состояний процесса (q_0, q_1, q_2, \dots). Пример: сигнал "Пуск" и сигнал "закреть заслонку" начинают первый шаг процесса (например, наполнения мешалки). Как только поступают информационные сигналы 1, 2, 3 (двигатель А включен, вентиль 1 открыт, время ожидания истекло), следует шаг процесса 2 и т. д.

Вопросы для самопроверки

1. В чем суть методики машинного моделирования систем?

2. Какие требования пользователь предъявляет к машинной модели системы?
3. Каковы основные принципы построения моделирующих алгоритмов процессов функционирования систем?
4. Какие нормативные документы применяются для реализации алгоритмов функционирования систем?
5. Сколько этапов и подэтапов моделирования существует?
6. Какие схемы используются при разработке алгоритмического и программного обеспечения машинного моделирования?
7. Какая техническая документация оформляется по каждому этапу моделирования системы?

4. ПЛАНИРОВАНИЕ МАШИННЫХ ЭКСПЕРИМЕНТОВ С МОДЕЛЯМИ СИСТЕМ

4.1. Стратегия эффективного планирования машинного эксперимента

Методы планирования экспериментов (ПЭ) с моделями в основном базируются на использовании *пассивного* и *активного эксперимента*.

При *пассивном эксперименте* исследователь находится в роли пассивного наблюдателя. Эксперимент ведет сама природа. Экспериментатору приходится только фиксировать значения входных и выходных величин. Модели полученные методом пассивного эксперимента почти не удается проверить на адекватность.

При *активном эксперименте* исследователь вмешивается в процесс эксперимента путем варьирования уровней входных величин. В рамках активного эксперимента построение модели проходит следующие этапы: 1) выбирается форма модели процесса; 2) строится план эксперимента; 3) проводится экспериментирование; 4) дается анализ результатов эксперимента.

На практике экспериментатору приходится чаще планировать не один, а несколько экспериментов, выполняя и анализируя каждый и, в соответствии с результатами, изменять план эксперимента. Стратегия такого эксперимента показана на рис. 4.1 [21].

В результате анализа результатов эксперимента может возникнуть необходимость исправления формы модели и плана эксперимента, тогда эксперименты повторяются вновь по указанной схеме.

4.2. Выбор и анализ эмпирических моделей

Стратегия эффективного планирования эксперимента начинается с выбора формы модели процесса. В зависимости от априорной информации о процессе различают две задачи: 1) построение модели при неизвестной структуре процесса; 2) нахождение параметров модели при заданной структуре.

Априорная информация о процессе для первого случая почти отсутствует. При этом необходимо решать задачу выбора структуры модели процесса, устанавливая класс моделей, оценивая степень стационарности и линейности процесса и др. Решение подобных задач связано с большими трудностями. На заданном этапе приходится решать вопросы содержательного неформального эвристического характера.



Рис. 4.1. Стратегия эксперимента

При известной функциональной зависимости между выходными и входными величинами задача является более простой и доступной. При этом форма модели известна. Экспериментатору остается построить план, провести эксперимент и сделать анализ результатов экспериментов.

Модели в статике. Модели в статике бывают следующего вида.

1. Линейная по входным величинам x и по параметрам:

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (4.1)$$

2. Нелинейная по x и линейная по коэффициенту β :

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2^2 + \dots + \beta_n x_n^n$$

3. Линейная по x и нелинейная по коэффициенту β :

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2^2 x_2 + \dots + \beta_n^n x_n$$

4. Нелинейная по x и по β :

$$\left. \begin{aligned} \eta &= e^{\beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n} \\ \eta &= \sqrt{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n} \end{aligned} \right\}$$

где η – реакция модели на входные величины x .

Встречаются модели и другого вида, например, тригонометрические, логические и др.

Определение функциональной связи по опытным данным представляет трудную задачу. По крайней мере универсальных способов не существует.

Можно указать простой способ определения вида моделей с одной независимой переменной: строят график полученных экспериментальных

данных. Если экспериментальные точки образуют некоторую гладкую кривую, то форма этой кривой может подсказать аналитический вид кривой.

Для проверки согласованности выбранной формы модели с экспериментальными данными преобразуют переменные к линейной форме (модели):

$$y = \alpha + \beta \cdot x$$

При исследовании реального процесса всегда имеется возможность представить результаты экспериментальных данных множеством моделей. Чтобы выбрать одну из них, надо знать для какой *цели* находится модель и какие *требования* к ней предъявляются. Очевидно, *главным требованием* следует считать способность модели предсказывать направление дальнейших опытов. *Другим требованием* является минимизация суммы квадратов отклонений наблюдаемых значений от вычисленных. Далее, желательно, чтобы преобразованная величина подчинялась нормальному распределению и чтобы дисперсия преобразованной случайной величины не зависела от самой величины.

Важным требованием является простота модели. Если несколько различных моделей отвечают необходимым требованиям, то следует принять ту из них, которая имеет более простую зависимость.

Таким образом, выбор модели, как правило, следует искать среди алгебраических полиномов. Причем на первом этапе планирования эксперимента выбирают линейную модель в виде алгебраического полинома первой степени (4.1).

Динамические модели. Динамические модели можно подразделить на два больших класса: традиционные модели и современные модели или ещё их называют модели в пространстве состояний.

К *традиционным моделям* относят простейшие *инерционные модели, модели на базе передаточных функций и комплексных коэффициентов передачи.*

Многие технологические процессы обладают инерционностью. Говорят, что такие процессы имеют «память». Инерционные модели бывают дискретные и непрерывные, линейные и нелинейные.

Если *модель линейная дискретная*, то её представляют в виде *суммы свертки* (весовая функция):

$$Y[nT] = \sum_{m=0}^{m-1} g[m\Delta\tau]\Delta\tau \cdot x[n\Delta t - (m+1)\Delta\tau] + e[n\Delta t] \quad (4.2)$$

где $Y[nT]$ – дискретные значения выходного сигнала; $g[m\Delta\tau]$ – дискретная весовая функция; $x[n\Delta t - (m+1)\Delta\tau]$ – дискретные значения входного сигнала; $e[n\Delta t]$ – случайная помеха.

В этой модели по существу нет параметров, она непараметрическая. Роль величин, которые необходимо определить из экспериментальных данных, играют значения ординат импульсной характеристики, которые рассматривают как коэффициенты регрессивной модели, а роль факторов

здесь играют значения одной и той же входной величины, но в разные моменты времени.

Если *линейная модель непрерывная*, то модель будет типа *интеграла свертки*:

$$y(t) = \int_0^{\infty} g(\tau)x(t-\tau)d\tau + e(t) \quad (4.3)$$

Модель (4.3) также как и (4.2) является непараметрической. Она содержит неизвестную весовую функцию $g(\tau)$. На практике широко используется возможность представления весовой функции для стационарной системы в *форме Релея-Ритца* путем разложения функции в ряд по системе известных ортогональных функций:

$$g(\tau) = \sum_{i=1}^k a_i \phi_i(\tau, \alpha)$$

где $\phi_i(\tau, \alpha)$ – заданная система базисных функций (фильтров), зависящая от параметра α .

Такой прием делает модель параметрической. Теперь она содержит ограниченное множество параметров a_i , подлежащих определению.

Нелинейные инерционные модели могут быть представлены в виде сумм или рядов Вольтерра.

В импульсном варианте модель можно представить:

$$Y[nt] = g_0(nT) + \sum_{i=0}^{\infty} g_1(n, i) \cdot x(n-i) + \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} g_2(n, i, j) \cdot x(n-i) \cdot x(n-j) \quad (4.4)$$

Для непрерывного объекта:

$$y(t) = g_0(t) + \int_{\tau_1=0}^{\infty} g_1(t, \tau_1) \cdot x(t-\tau_1) d\tau_1 + \int_{\tau_1=0}^{\infty} \int_{\tau_2=0}^{\infty} g_2(t, \tau_1, \tau_2) \cdot x(t-\tau_1) \cdot x(t-\tau_2) d\tau_1 d\tau_2 + \dots + e(t) \quad (4.5)$$

В выражениях (4.4) и (4.5) g_1 и g_2 соответственно весовые функции, их называют *ядрами Вольтерра первого и второго порядка*; g_0 – составляющая не связанная с входным сигналом.

Непосредственное определение ядер по опытным данным представляет собой сложную задачу. Поэтому их обычно аппроксимируют путем разложения в ряд по системе известных ортогональных функций:

$$g_1(t, \tau_1, \alpha) = \sum_{i=1}^k a_i \phi_i(\tau_1, \alpha)$$

$$g_2(t, \tau_1, \tau_2, \alpha) = \sum_i^k \sum_j^k a_{ij} \phi_i(\tau_1, \alpha) \cdot \phi_j(\tau_2, \alpha)$$

Теперь задача построения модели сводится к определению параметров весовой функции по косвенным экспериментальным данным.

Модели на базе передаточных функций

Для непрерывных динамических объектов выходная величина определяется интегралом свертки (4.3). Применим преобразование Лапласа. На основании теоремы свертывания имеем:

$$y(p) = g(p) \cdot x(p) + e(p) \quad (4.6)$$

Для импульсного динамического объекта, согласно выражения (4.2) после z – преобразования получаем:

$$y(z) = g(z) \cdot x(z) + e(z) \quad (4.7)$$

Модели (4.6) и (4.7) имеют вид регрессионных зависимостей. Параметры этих моделей, которые необходимо определить из экспериментальных данных, содержатся в выражениях передаточных функций.

Модели на основе комплексного коэффициента передачи. К модели частотного вида можно перейти путем замены в выражении (4.6) аргумента p на jw :

$$y(jw) = g(jw) \cdot x(jw) + e(jw) \quad (4.8)$$

В роли аргументов в моделях (4.6),(4.7),(4.8) выступает не время, а соответственно параметры преобразования: z , p , jw . Все эти модели линейны по входным сигналам, но, как правило, не линейны по параметрам.

4.3. Методы планирования эксперимента на модели

Основная задача планирования машинных экспериментов заключается в получении необходимой информации об исследуемой системе при ограниченных ресурсах (затраты машинного времени, памяти и т.п.). К числу частных задач, решаемых при планировании машинных экспериментов, относятся задачи уменьшения затрат машинного времени на моделирование, уменьшения погрешности результатов моделирования, проверки адекватности модели и т.п.

Эффективность машинных экспериментов существенно зависит от выбора плана эксперимента, т.к. именно план определяет объём и порядок проведения вычислений на ПК, приёмы накопления и статистической обработки результатов моделирования системы. Поэтому основная задача планирования машинных экспериментов с моделью формируется следующим образом: необходимо получить об объёме моделирования, заданном в виде моделирующего алгоритма (программы) при минимальных или ограниченных затратах машинных ресурсов на реализацию процесса моделирования.

Таким образом, при машинном моделировании необходимо не только рационально планировать и проектировать саму модель системы, но и процесс её использования, т.е. проведения с ней эксперимента.

При планировании машинных экспериментов возникает целый ряд проблем, взаимно связанных как с особенностью функционирования моделируемого объекта, так и с особенностью машинной реализации модели и обработки результатов эксперимента. В первую очередь к таким относятся проблемы построения плана машинного эксперимента, стохастической

сходимости результатов, ограниченности машинных ресурсов, уменьшения дисперсии оценок, полученных на машинной модели и т.д.

Рассмотрим основные понятия теории планирования эксперимента. В планировании эксперимента различают входные (изогенные) и выходные (эндогенные) переменные: $x_1, x_2, \dots, x_k; y_1, y_2, \dots, y_e$. Входные переменные в ТПЭ называют факторами а выходные – реакциями. Каждый фактор $x_i, i = 1, 2, \dots, k$ может принимать в эксперименте одно или несколько значений, называемых уровнями. Фиксированный набор уровней факторов определяет одно из возможных состояний рассматриваемой системы. Одновременно этот набор представляет собой условия проведения одного из возможных экспериментов.

Каждому фиксированному набору уровню факторов соответствует определённая точка в многомерном пространстве, называемая факторным пространством. Эксперименты не могут быть реализованы во всех точках факторного пространства, а лишь в принадлежащих допустимой области, как это например оказано для случая двух факторов X_1 и X_2 на рисунке 4.2 [4].

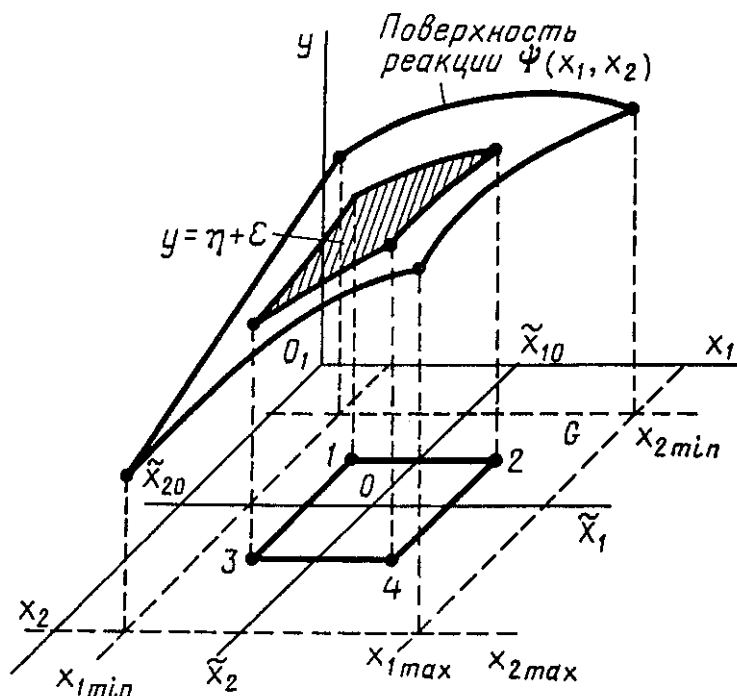


Рис. 4.2. Геометрическое представление поверхности реакции

Реакцию (отклик) системы можно представить в виде зависимости: $y_l = \Psi_l(x_1, x_2, \dots, x_k); e = 1 \dots m$. Функцию Ψ_e , связанную с факторами, называют функцией отклика, а её геометрический образ – поверхностью отклика. Исследователь заранее не известен вид зависимостей $\Psi_l, l = 1 \dots m$, поэтому используют приближение соотношения: $\tilde{y}_l = \Psi_l(x_1, x_2, \dots, x_k), l = 1, m$.

Зависимость и Ψ_l находятся по данным эксперимента. Последний необходимо поставить так, чтобы при минимальных затратах ресурсов (числе испытаний), варьируя выходные значения по специально сформулированным правилам, построить математическую модель системы и оценить её характеристики. Факторы при проведении эксперимента могут быть управляемыми и неуправляемыми, количественными или качественными, фиксированными и случайными. Фактор относится к изучаемым, если он включён в модель для изучения свойств системы. Количественными факторами являются интенсивности входящих потоков заявок, интенсивности потоков обслуживания, ёмкости накопителей, количество обслуживающих каналов и другие. Качественным факторам не соответствует числовая шкала (дисциплины постановки на очередь, обслуживание каналов и другие).

Фактор является управляемым, если его уровни целенаправленно выбираются экспериментатором.

При планировании эксперимента обычно изменяются несколько факторов.

Основными требованиями, предъявляемыми к факторам - независимость и совместимость. Совместимость означает, что все комбинации факторов осуществимы.

Для выбора конкретной модели планирования эксперимента необходимо сформулировать такие её особенности, как адекватность, содержательность, простота.

План эксперимента обычно используется для определения экстремальной характеристики объекта. Поэтому планирование эксперимента называется экстремальным. В планировании эксперимента наибольшее значение нашли модели в виде алгебраических полиномов.

Предполагаем, что изучается влияние K количественных факторов x_i на некоторую η в отведённый для экспериментирования локальной области факторного пространства ограниченного $x_{i \min} - x_{i \max}$, $i = 1 \dots k$.

Функцию отклика обычно выбирают линейной или квадратичной.

$$\eta = b_0 + \sum b_i x_i + \sum b_{ij} x_i x_j = f(\vec{x}) \vec{B} \quad (4.9)$$

где $\vec{f}(\vec{x})$ – вектор с элементами $f_\alpha(\vec{x})$, $\alpha = \overline{0, d}$, входящих в исходный полином; \vec{B} – вектор коэффициентов. Для двух факторов имеем: $f_0 = 1$, $f_1 = x_1$, $f_2 = x_2$, $f_{12} = x_1 x_2$, $f_{11} = x_1^2$, $f_{22} = x_2^2$. $\vec{B} = (b_0, b_1, b_2, b_{12}, b_{11}, b_{22})$.

Так как полином (4.9) содержит d коэффициентов, то план эксперимента должен содержать $N \geq d$ различных экспериментальных точек:

$$D = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{k1} \\ x_{12} & x_{22} & \dots & x_{k2} \\ \dots & \dots & \dots & \dots \\ x_{1N} & x_{2N} & \dots & x_{kN} \end{bmatrix}$$

где x_{in} – значение, которое принимает i -ая переменная в u -ом испытании. $i = 1 \dots k, u = 1 \dots N$. Матрица D называется планом эксперимента.

Реализовав испытания в N точках области факторного пространства, определённом планом эксперимента, получим вектор наблюдений имеющий следующий вид:

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}$$

где y_u – реакция соответствующей u -ой точке плана.

Плану эксперимента поставим в соответствие матрицу планирования:

$$x = \begin{bmatrix} f_{01} & f_{11} & f_{21} & \dots & f_{111} & f_{221} \\ f_{02} & f_{12} & f_{22} & \dots & f_{112} & f_{222} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ f_{0N} & f_{1N} & f_{2N} & \dots & f_{11N} & f_{22N} \end{bmatrix}$$

где f_{il}, f_{ijl} – координатные функции при соответствующих коэффициентах модели, в l -ом эксперименте.

Построению плана эксперимента предшествует проведение ряда неформализованных действий (принятия решения) направленных на выбор локальной области факторного пространства G .

Необходимо учитывать, что как только модель сформирована включение дополнительных факторов для уточнения модели невозможно. Вначале следует выбрать границы $x_i \min$ и $x_i \max$ области определения факторов исходя из свойств объекта. Например, температура при термобарических экспериментах не может быть ниже абсолютного нуля и выше температуры плавления материала из которого изготовлена термобарокамера.

После определения области G необходимо найти нулевые (основные) уровни факторов и интервалы варьирования $\Delta x_i, i = 1 \dots k$.

Эксперимент, в котором реализуются все возможные сочетания уровней факторов, называется полным факторным экспериментом (ПФЭ). Если выбранная модель включает только линейные члены полинома и их произведения, то для оценки коэффициентов модели используется ПФЭ с варьированием всех k факторов на двух уровнях, т.е. $q = 2$. Такие планы называются планами типа 2^k , где $n = 2^k$ – число всех возможных испытаний.

Начальным этапом ПЭ для получения коэффициентов линейной модели основан на варьировании факторов на двух уровнях: нижнем $x_{1н}$ и верхнем $x_{1в}$, симметрично расположенных относительно основного уровня x_{10} , $i = 1 \dots k$. Геометрическая интерпретация показана ниже на рис. 4.3.

Для упрощения записи условий каждого эксперимента факторы кодируют в виде безразмерных величин $x_i = (x_i - x_{i0}) / \Delta x_i$, $i = 1, 2, \dots, k$. Средний уровень кодированного фактора является нулём 0, граничные значения соответственно +1 и -1.

4.4. Проверка адекватности модели

Для того чтобы принять решение относительно модели необходимо проверить гипотезу: линейная модель по параметрам удовлетворительно описывает экспериментальные данные.

4.4.1. Критерий Фишера

Проверку адекватности модели выполняют с помощью *критерия Фишера*:

$$F = \frac{S_r^2}{S_e^2},$$

где S_r^2 – дисперсия неточности модели, S_e^2 – дисперсия ошибки эксперимента.

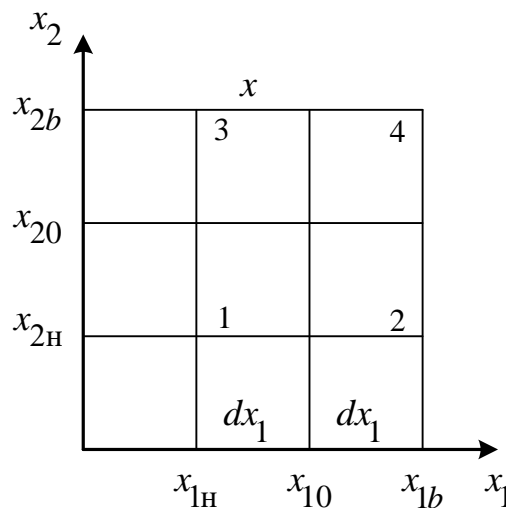


Рис. 4.3. ПЭФ типа 2^2

Если $F < F_{1-\alpha}$, то гипотеза о том, что линейная модель адекватна, принимается. Здесь $F_{1-\alpha}$ – табличное значение; α – уровень значимости; F – вычисленное значение.

При $F > F_{1-\alpha}$ гипотеза адекватности линейной модели по параметрам отвергается.

4.4.2. Определение дисперсий неточности модели и ошибки эксперимента

Дисперсия S_e^2 характеризует ошибку эксперимента. Её можно получить путем возведения в квадрат разности $(y_{ij} - \bar{y}_i)$ с последующим суммированием по всем экспериментальным точкам:

$$S_e^2 = \frac{\sum_{i=1}^n \sum_{j=1}^{P_i} (y_{ij} - \bar{y}_i)^2}{\sum_{i=1}^n P_i - n}$$

Знаменатель дисперсии равен разности между общим количеством экспериментальных точек $\sum_{i=1}^n P_i$ и числом наложенных связей n , т.к. каждая выборка дает одну связь.

Дисперсия S_r^2 характеризует неточность подгонки модели, её неадекватность. Поэтому её можно получить из разности $(\bar{y}_i - \hat{y}_i)$, которая после возведения в квадрат и суммирования представляет собой сумму квадратов отклонений относительно эмпирической линии регрессии:

$$S_r^2 = \frac{\sum_{i=1}^n P_i (\bar{y}_i - \hat{y}_i)^2}{n - 2}$$

Здесь число степеней свободы равно $(n - 2)$. Две связи вызваны двумя ограничениями, связанными с определением b_0 и b_1 .

Если математическое ожидание дисперсии S_r^2 равно дисперсии выходной величины η :

$$M[S_r^2] = \sigma_\eta^2,$$

то полученная эмпирическая модель будет корректной.

4.4.3. Определение дисперсии воспроизводимости эксперимента

Дисперсия S_y^2 характеризует ошибку эксперимента и называют *дисперсией воспроизводимости эксперимента*. Она может быть получена путем объединения дисперсий S_e^2 и S_r^2 :

$$S_y^2 = \frac{\sum_{i=1}^n \sum_{j=1}^{P_i} (y_{ij} - \bar{y}_i)^2 + \sum_{i=1}^n P_i (\bar{y}_i - \hat{y}_i)^2}{(\sum_{i=1}^n P_i - n) + (n - 2)} = \frac{\sum_{i=1}^n \sum_{j=1}^{P_i} (y_{ij} - \hat{y}_i)^2}{\sum_{i=1}^n P_i - 2}$$

Более простое определение $S_y^2 = S_e^2$.

4.4.4. Проверка однородности дисперсий

Простейшим критерием проверки однородности дисперсий является критерий Фишера, который представляет из себя отношение большей дисперсии к меньшей:

$$F = \frac{S_{\max}^2}{S_{\min}^2}$$

Полученная величина F сравнивается с табличной величиной $F_{1-\alpha}$. Если $F > F_{1-\alpha}$, то дисперсии значимо отличаются друг от друга и тогда они неоднородны. Если сравнивается много дисперсий (более двух) и одна из дисперсий значимо превышает остальные, то можно применять критерий Кохрена – это отношение максимальной дисперсии к сумме всех дисперсий:

$$G = \frac{S_{\max}^2}{\sum_{i=1}^n S_i^2},$$

Гипотеза об однородности подтверждается, если $G < G_t$, где G_t – табличное значение.

4.5. Проверка значимости коэффициентов модели

Проверка значимости коэффициентов проводится двумя равноценными способами: 1. по критерию Стьюдента; 2. путем построения доверительного интервала.

Согласно критерия Стьюдента влияние коэффициента b_j значимо, если $\frac{b_j}{S_{b_j}} > t_{\text{табл}}$, где S_{b_j} – дисперсия ошибки в определении коэффициента;

$t_{\text{табл}}$ – табличное значение выбирается с учетом уровня значимости.

Для определения доверительного интервала используют безразмерную t -статистику Стьюдента

$$t = \frac{b_j - \beta_j}{S_{b_j}} \quad (4.10)$$

и считают, что оценка b_j распределена относительно β_j по нормальному закону.

Согласно выражения (4.10), находим доверительные интервалы:

$$b_j - S_{b_j} t_{1-\frac{\alpha}{2}} \leq \beta_j \leq b_j + S_{b_j} t_{1-\frac{\alpha}{2}}.$$

где $t_{1-\frac{\alpha}{2}}$ – квантиль t -распределения.

4.6. Стратегическое планирование машинных экспериментов с моделями систем

Можно выделить стратегическое и тактическое ПЭ на моделях систем.

Стратегическое планирование – ставит своей целью получение необходимой информации о системе с помощью модели, реализованной на ПК. Оно аналогично внешнему проектированию при создании системы.

Тактическое планирование – определяет способы проведения каждой серии испытаний машинной модели. Оно аналогично внутреннему проектированию системы.

Рассмотрим элементы стратегического планирования ПЭ. Его целью может быть:

1. Получение функции реакции системы от независимых фактов: $y = f(b_0, b_1, \dots, x_1, x_2, \dots, x_k)$.

2. Нахождение экстремума: $f(b_0, b_1, \dots, x_1, x_2, \dots, x_k)$.

Во 2-ом случае для определения наилучшей комбинации фактов могут быть использованы методы систематической или случайной выборки.

К систематическим относятся методы: 1. одного фактора; 2. предельного анализа; 3. наискорейшего спуска; 4. равномерной сетки.

Проблемой является большое количество факторов. Для $k = 10$ ПЭФ должен состоять из 1024 точек. Используют неполные планы, метод "поверхности реакции".

Следующей проблемой является многокомпонентность функции реакции. Здесь можно использовать последовательное однокомпонентное ПЭ. Этот подход не всегда возможен из-за связанности компонентов. Используются интегральные оценки с применением весовых функций, функций полезности и т.д.

Другой проблемой является стохастическая сходимость результатов ПЭ. В качестве результатов ПЭ используются средние некоторых распределений, для оценки которых применяют выборочные средние, найденные путём многократных прогонов модели на ПК. Сходимость выборочных средних с ростом объема выборки называется стохастической. Эта сходимость, как правило, медленная. Если σ – стандартное отклонение среднего N наблюдений будет равно σ/\sqrt{N} , т.е. для уменьшения случайной выборки в k раз требуется увеличить объем выборки в k^2 раз.

Планирование машинного эксперимента представляет собой итерационный процесс, когда выбранная модель плана эксперимента проверяется на реализуемость, а затем, если это необходимо, вносят соответствующие коррективы в модель.

Планирование эксперимента с моделью проводится в несколько этапов: 1. построение структурной модели; 2. построение функциональной модели.

Структурная модель ПЭ характеризуется числом факторов и числом уровней для каждого фактора. Из опыта известно, что 20% факторов определяют 80% свойств системы.

Ортогональное распределение плана упрощает определение коэффициентов аппроксимации. Упрощение дает принятие числа уровней всех факторов одинаковыми (не больше 3). Функциональная модель ПЭ определяет количество элементов структурной модели N_f , т.е. необходимое число

различных информационных точек $N_{\text{ф}}$. Причём $N_{\text{ф}} < N_{\text{с}}$, где $N_{\text{с}} = q_1, q_2, \dots, q_k$ – число экспериментов ПФЭ.

4.7. Тактическое планирование машинных экспериментов с моделями систем

Здесь решают проблемы:

- определения начальных условий и их влияния на достижения установившегося результата при моделировании;
- обеспечения точности и достоверности результатов моделирования;
- уменьшения дисперсии оценок характеристик процесса функционирования моделируемых систем;
- выбора правил автоматической остановки имитационного эксперимента с моделями.

Рассмотрим ПФЭ типа 2^3 (таблица 4.1):

Таблица 4.1.

номер испытания	1	2	3	4	5	6	7	8
\tilde{x}_1	-1	-1	-1	-1	+1	+1	+1	+1
\tilde{x}_2	-1	-1	+1	+1	-1	-1	+1	+1
\tilde{x}_3	-1	+1	-1	-1	-1	+1	-1	+1

ПЭФ даёт возможность определить не только коэффициенты регрессии, соответствующие линейным эффектам, но и коэффициенты регрессии соответствующие всем эффектам взаимодействия. Эффект взаимодействия двух или более факторов появляется при одновременном варьировании этих факторов, когда действие каждого из них на выход зависит от уровня, на которых находятся другие факторы.

Для оценки свободного члена b_0 и определения эффектов взаимодействия $b_{12}, b_{13}, \dots, b_{123}$ план эксперимента D расширяют до матрицы планирования X путём добавления соответствующей фиктивной переменной: единичного столбца x_0 и столбцов произведений как показано, например, для ПЭФ типа 2^3 в таблице 4.2.

Как видно из рассмотренных ПЭ типа 2^2 и 2^3 количество испытаний ПЭФ значительно превосходит число определяемых коэффициентов линейной модели плана эксперимента, что увеличивает расход ресурсов ПК по времени. Возникает проблема сокращения количества экспериментов.

Таблица 4.2.

№ Исп.	x_0	План ПЭФ			x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	Реакция y
		x_0	x_2	x_3					
	+1	+1	+1	+1	+	+	+	+	y_1

+1	-1	+1	+1	-	-	+	-	y ₂
+1	+1	-1	+1	-	+	-	-	y ₃
+1	-1	-1	+1	+	-	-	+	y ₄
+1	+1	+1	-1	+	-	-	-	y ₅
+1	-1	+1	-1	-	+	-	+	y ₆
+1	+1	-1	-1	-	-	+	+	y ₇
+1	-1	-1	-1	+	+	+	-	y ₈

С этой целью рассмотрим построение планов так называемого дробного факторного эксперимента (ДФЭ). Пусть имеется ПЭФ типа 2^2 . Используя матрицу планирования X , например приведённую в предыдущей таблице, можно вычислить коэффициенты и предусмотреть результаты (таблица 4.3) в виде уравнения:

$$y = b_0 + b_1 \tilde{x}_1 + b_2 \tilde{x}_2 + b_{12} \tilde{x}_1 \tilde{x}_2$$

Таблица 4.3.

N/n	\tilde{x}_0	План ПЭФ		$\tilde{x}_1 \tilde{x}_2 (\tilde{x}_0)$	Отклик y
		\tilde{x}_1	\tilde{x}_2		
1	1	+1	+1	+1	y ₁
2	1	-1	+1	-1	y ₂
3	1	+1	-1	-1	y ₃
4	1	-1	-1	+1	y ₄

Если в выбранных интервалах варьирования уровня процесс можно описать линейной моделью, то достаточно определить три коэффициента b_0, b_1, b_2 . Таким образом, остаётся одна степень свободы, которую можно использовать для построения плана эксперимента D для 3-х переменных, в которых уровни 3-го фактора изменяются как в таблице рассмотренной немного раньше для столбца $x_1 x_2$ (эффектов взаимодействия).

Мы получим так называемый дробный факторный эксперимент. В нём уже не будет отдельных оценок для коэффициентов регрессии, как в ПЭФ, они будут рассчитываться по формулам:

$$\tilde{b}_1 = \beta_1 + \beta_{23}; \quad \tilde{b}_2 = \beta_2 + \beta_{13}; \quad \tilde{b}_3 = \beta_3 + \beta_{12}.$$

При постулировании линейной модели все парные взаимодействия не учитывают. Таким образом, вместо γ испытанный в ПЭФ для 3-х факторов получим 4 испытания в ДФЭ. Правило проведения ДФЭ формулируется так: для сокращения числа испытаний новому фактору присваивается значение вектор-столбца матрицы, принадлежащего взаимодействию, которым можно пренебречь.

При проведении эксперимента из 4-х испытаний для оценки влияния 3-х факторов пользуются половинный ПЭФ типа 2^3 , так называемой "полу

репликой". Если приравнять x_3 и x_1x_2 , что можно получить 2-ую "полу реплику".

Для обозначения дробных реплик, в которых λ линейных эффектов приравнены к эффектам взаимодействия, пользуются условным обозначением $2^{2-\lambda}$. Например, "полу реплика" от 2^6 записывается в виде 2^{6-1} , а "четверть реплика" 2^{6-2} .

Смешивание надо производить так, чтобы основные коэффициенты были смешаны с коэффициентами при взаимодействиях самого высокого порядка. Не рекомендуется использовать реплики для $n \geq 15$.

Следует иметь в виду, что малый шаг варьирования Δx ; ($j = 1 \dots n$) может повлечь статистическую незначимость оценки коэффициента уравнения регрессии. В случае, если полученная мат. модель окажется неадекватной, проводятся эксперименты с меньшим шагом варьирования.

Если линейные модели, построенные с помощью ПЭФ и ДФЭ, неадекватны, то переходят к построению квадратичных моделей.

Оптимальный план для квадратичной модели целесообразно строить таким образом, чтобы он включал точки плана для линейной модели. Это позволяет сократить число опытов.

4.8. Планирование экспериментов при построении полной квадратичной модели

В некоторых случаях существенными могут оказаться коэффициенты при квадратичных переменных, их кубов и т.д.

Для двухфакторного эксперимента модель может быть представлена выражением

$$y = b_0x_0 + b_1x_1 + b_2x_2 + b_{12}x_1x_2 + b_{11}x_1^2 + b_{22}x_2^2$$

Полученные вектор - столбцы x_1^2 и x_2^2 являются единичными столбцами, совпадающие друг с другом и с фиктивным столбцом x_0 . Эти столбцы неразличимы, поэтому нельзя сказать за счет чего получилась величина b_0 . Очевидно, она включает в себя значения свободного члена β_0 и вклады квадратичных членов. Символически это можно записать:

$$b_0 \rightarrow \beta_0 + \sum_{i=1}^k \beta_{ii}$$

Для квадратичной модели получается следующая система смешивания: $b_0 \rightarrow \beta_0 + \beta_{11} + \beta_{22}$, $b_1 \rightarrow \beta_1$, $b_2 \rightarrow \beta_2$, $b_{12} \rightarrow \beta_{12}$.

Следовательно, планирование эксперимента на двух уровнях не дает возможности получить отдельные оценки коэффициентов при квадратичных членах и фиктивной переменной x_0 .

Согласно теории интерполяции, для решения задачи нахождения отдельных оценок число уровней каждой из независимых переменных долж-

но быть на единицу больше степени интерполяционного полинома. Для полинома второй степени число уровней должно быть равно трем.

Однако применение методов ПФЭ плана 3^n не является рациональным из-за резкого увеличения опытов эксперимента. Поэтому разработаны специальные методы построения планов второго порядка.

Например, в качестве двухфакторных планов второго порядка могут служить планы, представляемые вершинами и, по крайней мере, одной центральной точкой любого $(n - 1)$ мерного правильного многоугольника (который можно вписать в круг).

Например, имеем восьмиугольный план (рисунок 4.4, таблица 4.4).

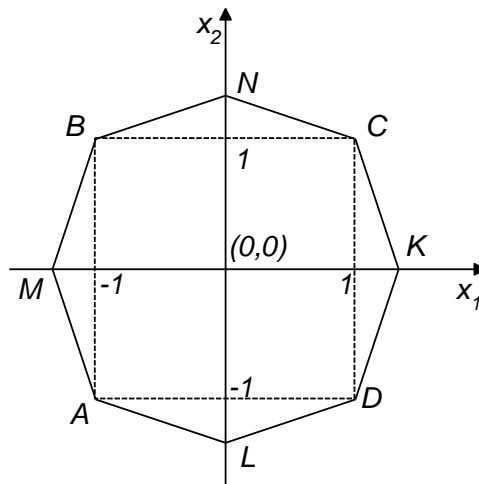


Рис. 4.4. Восьмиугольный план эксперимента

Таблица 4.4

Опыт	x_1	x_2	Описание
1	-1	-1	План 2^2 представлен квадратом ABCD
2	+1	-1	
3	-1	+1	
4	+1	+1	
5	$\sqrt{2}$	0	План представлен звёздными точками MNKL
6	$-\sqrt{2}$	0	
7	0	$\sqrt{2}$	
8	0	$-\sqrt{2}$	
9	0	0	Центральная точка

Этот пример можно обобщить на случай получения планов второго порядка. Для этого к ПФЭ типа 2^n добавляется центральная точка с координатами $(0, 0, \dots, 0)$ и, так называемые, *звёздные точки* с координатами $(0, 0, \dots, \pm\alpha, \dots, 0)$, лежащие на сфере диаметра 2α . Т.е. план ПФЭ достраивается до *плана второго порядка*. Такой план называется композиционным планом.

Для трехфакторного эксперимента МПЭ будет иметь следующий вид (таблица 4.5).

Добавление двух сфер, образованных звездными точками и центральной точкой, к ПФЭ позволяет получить отдельные оценки b_0 и b_{ii} . Все три сферы образуют композиционный план второго порядка.

В зависимости от критерия оптимальности плана, различают *ортогональное композиционное планирование* и *рототабельное композиционное планирование*.

4.8.1. Ортогональное центральное композиционное планирование

Критерием оптимальности является ортогональность столбцов матрицы планирования. В силу этого свойства все коэффициенты модели определяются независимо друг от друга.

Таблица 4.5.

Опыт	x_0	x_1	x_2	x_3	x_1^2	x_2^2	x_3^2	Описание
1	+1	-1	-1	-1	+1	+1	+1	ПФЭ 2^2
2	+1	+1	-1	-1	+1	+1	+1	
3	+1	-1	+1	-1	+1	+1	+1	
4	+1	+1	+1	-1	+1	+1	+1	
5	+1	-1	-1	+1	+1	+1	+1	
6	+1	+1	-1	+1	+1	+1	+1	
7	+1	-1	+1	+1	+1	+1	+1	
8	+1	+1	+1	+1	+1	+1	+1	
9	+1	$-\alpha$	0	0	α^2	0	0	Звездные точки
10	+1	$+\alpha$	0	0	α^2	0	0	
11	+1	0	$-\alpha$	0	0	α^2	0	
12	+1	0	$+\alpha$	0	0	α^2	0	
13	+1	0	0	$-\alpha$	0	0	α^2	
14	+1	0	0	$+\alpha$	0	0	α^2	
15	+1	0	0	0	0	0	0	
16	+1	0	0	0	0	0	0	Центральная нуль точка

Ортогональность столбцов x_0 и x_i^2 достигается путем преобразования вида:

$$\hat{x}_i^2 = x_i^2 - \frac{\sum_{j=1}^N x_{ji}^2}{N} = x_i^2 - \bar{x}_i^2 \quad (4.11)$$

С учетом выражения (4.11) условие ортогональности выполняется:

$$\begin{aligned} \sum_{j=1}^N x_{j0}^2 \cdot \hat{x}_{ji}^2 &= \sum_{j=1}^N x_{j0}^2 \left[x_{ji}^2 - \frac{\sum_{j=1}^N x_{ji}^2}{N} \right] = \sum_{j=1}^N 1 \cdot x_{ji}^2 - \frac{N \sum_{j=1}^N x_{ji}^2}{N} = \\ &= \sum_{j=1}^N x_{ji}^2 - N \frac{\sum_{j=1}^N x_{ji}^2}{N} = 0 \end{aligned}$$

Из условия ортогональности выбирают координату α звездной точки для n - независимых переменных.

Таблица 4.6 соответствует ортогональному композиционному плану для количества переменных $n = 2, 3, 4$, где N_α – число звездных точек; N_0 – число точек в центре эксперимента; N_c – количество точек куба (гиперкуба) при ПФЭ; N – общее число точек факторного пространства.

Таблица 4.6

n	α	N_α	N_0	N_c	N
2	1,0	4	1	4	9
3	1,215	6	1	8	15
4	1,414	8	1	16	25

Составим матрицу ортогонального планирования для трехфакторного эксперимента. В таблице 4.7 $\hat{x}_i^2 = x_i^2 - \bar{x}_i^2 = x_i^2 - 0,73$.

Таблица 4.7

Но- мер опы- та	x_0	x_1	x_2	x_3	\hat{x}_1^2	\hat{x}_2^2	\hat{x}_3^2	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$
	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}
1	+1	-1	-1	-1	0,27	0,27	0,27	+1	+1	+1	-1
2	+1	+1	-1	-1	0,27	0,27	0,27	-1	-1	+1	+1
3	+1	-1	+1	-1	0,27	0,27	0,27	-1	+1	-1	+1
4	+1	+1	+1	-1	0,27	0,27	0,27	+1	-1	-1	-1
5	+1	-1	-1	+1	0,27	0,27	0,27	+1	-1	-1	+1
6	+1	+1	-1	+1	0,27	0,27	0,27	-1	+1	-1	-1
7	+1	-1	+1	+1	0,27	0,27	0,27	-1	-1	+1	-1
8	+1	+1	+1	+1	0,27	0,27	0,27	+1	+1	+1	+1
9	+1	-1,215	0	0	0,75	-	-	0	0	0	0
						0,73	0,73				
10	+1	+1,21	0	0	0,75	-	-	0	0	0	0
						0,73	0,73				
11	+1	0	-1,215	0	-	0,75	-	0	0	0	0
						0,73	0,73				
12	+1	0	+1,21	0	-	0,75	-	0	0	0	0
						0,73	0,73				
13	+1	0	0	-1,215	-	-	0,75	0	0	0	0
						0,73	0,73				
14	+1	0	0	+1,21	-	-	0,75	0	0	0	0

					0,73	0,73					
15	+1	0	0	0	-	-	-	0	0	0	0
					0,73	0,73	0,73				

где 1 – 8 опыт соответствует ПФЭ 2^3 ; 9 – 14 опыт соответствует звездным точкам; 15 опыт – центральная точка.

Анализ результатов экспериментов при ортогональном композиционном планировании имеет некоторые особенности. Так оценки коэффициентов уравнения регрессии находятся с неодинаковой дисперсией.

$$S_{b_i}^2 = \frac{S_y^2}{P \sum_{j=1}^N z_{ji}} \quad (4.12)$$

Знаменатель выражения (4.12) для разных коэффициентов различен. В формуле (4.12) P – число параллельных опытов.

Уравнение регрессии для полного квадратичного полинома будет иметь вид:

$$\begin{aligned} y &= b_0 + \sum_{i=1}^n b_i x_i + \sum_{i,j=1}^n b_{ij} x_i x_j + \sum_{i=1}^n b_{ii} (x_i^2 - \bar{x}_i^2) = \\ &= b_0^* + \sum_{i=1}^n b_i x_i + \sum_{i,j=1}^n b_{ij} x_i x_j + \sum_{i=1}^n b_{ii} x_i^2, \end{aligned}$$

где $b_0^* = b_0 - \sum_{i=1}^n b_{ii} \bar{x}_i^2$, n – количество факторов.

Дисперсия коэффициента b_0^* будет оцениваться по формуле:

$$S_{b_0^*}^2 = S_{b_0}^2 + \sum_{i=1}^n S_{b_{ii}}^2 \bar{x}_i^2$$

Коэффициенты b_i , b_{ii} и b_{ij} определяются по формулам:

$$b_i = \frac{\sum_{j=1}^N x_{ji} \bar{y}_j}{\sum_{i=1}^n x_{ji}}, \quad b_{ii} = \frac{\sum_{u=1}^N x_{ui}^2 \bar{y}_u}{N}, \quad b_{ij} = \frac{\sum_{u=1}^N x_{ui} x_{uj} \bar{y}_u}{N},$$

где $u = \overline{1, N}$ – число опытов; $\bar{y}_u = \frac{\sum_{p=1}^P y_p}{P}$, P – число прогонов.

Проверку значимости коэффициентов можно провести по критерию Стьюдента

$$t_i = \frac{b_i}{S_{b_i}},$$

где $S_{b_i}^2 = \frac{S_y^2}{P \sum_{u=1}^N x_{ui}}$ – дисперсия коэффициента b_i .

Проверка адекватности модели проводится по выше изложенной методике с помощью F -критерия Фишера.

Из-за неодинаковой дисперсии коэффициентов регрессии критерий ортогональности является недостаточно сильным критерием оптимальности для планирования второго порядка. Поэтому точность предсказания выходной величины в различных направлениях факторного пространства неодинакова.

Лучшим методом планирования является такой метод, который обеспечивает одинаковую точность во всех направлениях на одинаковом расстоянии от центра. Таким методом является *рототабельное композиционное планирование*.

4.8.2. Рототабельное композиционное планирование

Критерием оптимальности в рототабельном планировании является условие $\sigma_y^2 = const$ при одинаковом удалении точек эксперимента от центра, т.е. $R = const$.

Если имеются двухфакторные планы, то, как уже было отмечено, типичными примерами рототабельных планов являются планы, представляемые вершинами и, по крайней мере, одной центральной точкой любого $(n - 1)$ – мерного правильного многоугольника, который можно вписать в круг.

Композиционные центральные рототабельные планы также как и ортогональные состоят из трех сфер: сфера нулевого радиуса – центральные точки; сфера точек куба или гиперкуба и сфера звездных точек. Равномерность расположения точек на сфере приводит к вырожденным матрицам. Для устранения вырожденности используют сферу нулевого радиуса с несколькими центральными точками (таблица 4.8).

Таблица 4.8

n	α	N_α	N_0	N_c	N
2	1,414	4	5	4	13
3	1,682	6	6	8	20
4	2	8	7	16	31

где N_α – число звездных точек; N_0 – число точек в центре эксперимента; N_c – количество точек куба (гиперкуба); N – общее число точек факторного пространства.

Матрица планирования рототабельного плана второго порядка для трехфакторного эксперимента будет представлена в таблице 4.9.

Эксперимент проводится аналогично ПФЭ, однако оценки коэффициентов рассчитываются по своим формулам:

$$b_0 = \frac{A}{N} [2\lambda^2(n+2) \sum_{j=1}^N x_{j,0} \bar{y}_j - 2\lambda C \sum_{i=1}^n \sum_{j=1}^N x_{j,i} \bar{x}_j]$$

$$b_{ii} = \frac{A}{N} \left\{ C^2[(n+2)\lambda - n] \sum_{j=1}^N x_{j,i} \bar{y}_j + C^2(1-\lambda) \sum_{i=1}^n \sum_{j=1}^N x_{j,i} \bar{x}_j - 2\lambda C \sum_{j=1}^N x_{j,0} \bar{y}_j \right\}$$

$$b_i = \frac{C}{N} \sum_{j=1}^N x_{j,i} \bar{y}_j$$

$$b_{ij} = \frac{C^2}{N\lambda} \sum_{u=1}^N x_{ui} x_{uj} \bar{y}_u$$

$$C = \frac{N}{\sum_{j=1}^N x_{j,i}}, A = \frac{1}{2\lambda[(n+2)(\lambda-n)]}, \lambda = \frac{nN \sum_{w=1}^k N_w P_w}{(n+2) \left(\sum_{w=1}^k N_w P_w^2 \right)^2}$$

где N_w – число точек на сфере радиуса P_w ; k – число сфер ($k = 3$).

Проводится проверка значимости коэффициентов по t - критерию Стьюдента. Оценки дисперсии и коэффициентов вычисляются по формулам:

$$S_{b_0}^2 = \frac{2A\lambda^2(n+2)S_y^2}{NP}$$

$$S_{b_i}^2 = \frac{A[(n+1)\lambda - (n-1)]C^2S_y^2}{NP}$$

$$S_{b_{ij}}^2 = \frac{C^2S_y^2}{\lambda NP}$$

Таблица 4.9

Номер опыта	x_0	x_1	x_2	x_3	x_1^2	x_2^2	x_3^2	x_1x_2	x_1x_3	x_2x_3
	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9
1	+1	-1	-1	-1	+1	+1	+1	+1	+1	+1
2	+1	+1	-1	-1	+1	+1	+1	-1	-1	+1
3	+1	-1	+1	-1	+1	+1	+1	-1	+1	-1
4	+1	+1	+1	-1	+1	+1	+1	+1	-1	-1
5	+1	-1	-1	+1	+1	+1	+1	+1	-1	-1
6	+1	+1	-1	+1	+1	+1	+1	-1	+1	-1
7	+1	-1	+1	+1	+1	+1	+1	-1	-1	+1
8	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
9	+1	-1,682	0	0	2,828	0	0	0	0	0
10	+1	+1,682	0	0	2,828	0	0	0	0	0
11	+1	0	-1,682	0	0	2,828	0	0	0	0
12	+1	0	+1,682	0	0	2,828	0	0	0	0
13	+1	0	0	-1,682	0	0	2,828	0	0	0
14	+1	0	0	+1,682	0	0	2,828	0	0	0
15	+1	0	0	0	0	0	0	0	0	0
16	+1	0	0	0	0	0	0	0	0	0
17	+1	0	0	0	0	0	0	0	0	0
18	+1	0	0	0	0	0	0	0	0	0
19	+1	0	0	0	0	0	0	0	0	0
20	+1	0	0	0	0	0	0	0	0	0

Проверка адекватности модели проводится известным методом Фишера.

Рототабельные планы нашли широкое применение на практике. Однако с точки зрения математиков, занимающихся развитием математической статистики, выбор такого критерия представляется *мало обоснованным*. Он не вытекает логически из тех идей, на которые базируется математика. Например, как выбрать расстояние α до звездных точек; не все пространства независимых переменных, отведенное для эксперимента, используется в композиционных планах и т.д. Невозможно из множества рототабельных планов при одном и том же числе факторов выбрать лучший, т.к. не было критерия оценки. Поэтому этот критерий стали относить к *эмпирико-интуитивным критериям*.

Наряду с развитием планирования экспериментов, основанных на эмпирико-интуитивных критериях Бокса в США стало развиваться второе чисто теоретическое направление, которое связывают с планом ученого Кифера. Он установил связь между некоторыми критериями оптимальности; теоретически доказал, что для отдельных видов регрессии одни и те же планы могут отвечать сразу нескольким критериям оптимальности.

4.8.3. Разбиение матрицы планирования 2^K на блоки

При проведении серии экспериментов, осуществляемых в течение некоторого периода времени или с использованием нескольких партий веществ может произойти изменения условий эксперимента. Для предотвращения смещения среднего уровня отклика точки экспериментального плана рекомендуется выбирать в случайной последовательности. Однако в некоторых случаях влияние внешних условий можно избежать путем подходящего разбиения матрицы на блоки по сырью или последовательности испытаний. Так, при наличии двух партий сырья матрицу 2^3 можно разбить на два блока таким образом, чтобы эффект сырья сказался на величине трехфакторного взаимодействия $x_1x_2x_3$ (таблица 4.10). В этом случае все линейные коэффициенты и парные взаимодействия будут освобождены от влияния неоднородности сырья.

Таблица 4.10

№ блока	x_0	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	y
1	+	-	-	+	+	-	-	+	$y_1 + \varepsilon$
	+	+	-	-	-	-	+	+	$y_2 + \varepsilon$
	+	-	+	-	-	+	-	+	$y_3 + \varepsilon$
	+	+	+	+	+	+	+	+	$y_4 + \varepsilon$
2	+	-	-	-	+	+	+	-	y_5
	+	+	-	+	-	+	-	-	y_6
	+	-	+	+	-	-	+	-	y_7

	+	+	+	-	+	-	-	-	y_8
--	---	---	---	---	---	---	---	---	-------

Различие в сырье рассматривают как новый фактор x_4 . Матрица 2^3 , представленная двумя блоками: первый блок ($x_1x_2x_3 = +1$) и второй блок ($x_1x_2x_3 = -1$), является полурепликой 2^{4-1} с определяющим контрастом $1 = x_1x_2x_3x_4$.

Коэффициенты модели будут равны

$$b_0 = \frac{1}{8}[(y_1 + \varepsilon) + (y_2 + \varepsilon) + (y_3 + \varepsilon) + (y_4 + \varepsilon) + y_5 + y_6 + y_7 + y_8]$$

$$b_0 \rightarrow \beta_0 + \frac{\varepsilon}{2}$$

$$b_1 = \frac{1}{8}[-(y_1 + \varepsilon) + (y_2 + \varepsilon) - (y_3 + \varepsilon) + (y_4 + \varepsilon) - y_5 + y_6 - y_7 + y_8]$$

$$b_2 \rightarrow \beta_2 \quad b_3 \rightarrow \beta_3 \quad b_{12} \rightarrow \beta_{12} \quad b_{13} \rightarrow \beta_{13} \quad b_{23} \rightarrow \beta_{23}$$

$$b_{123} = \frac{1}{8}[(y_1 + \varepsilon) + (y_2 + \varepsilon) + (y_3 + \varepsilon) + (y_4 + \varepsilon) - y_5 - y_6 - y_7 - y_8]$$

$$b_{123} \rightarrow \beta_{123} + \frac{\varepsilon}{2}.$$

Эффект сырья отразился на подсчете свободного члена b_0 и коэффициента b_{123} .

Аналогично можно разбить на два блока любой эксперимент 2^k . Нужно только выбрать такой элемент взаимодействия, которым можно безболезненно пожертвовать. Обычно, при отсутствии априорных сведений выбирают взаимодействие самого высокого порядка $x_1x_2x_3$ для 2^3 , $x_1x_2x_3x_4$ для 2^4 , $x_1x_2x_3x_4x_5$ для x_5 и т.д.

Если имеется четыре источника неоднородности, которые могут заметно исказить результаты эксперимента, то матрицу 2^4 разбивают на четыре блока таким образом, чтобы линейные эффекты были освобождены от влияния межблокового эффекта.

Схемы разбиения на блоки рототабельных планов второго порядка так же нужно выполнить с учетом того, чтобы различие между блоками не влияло на коэффициенты при линейных, перекрестных и квадратичных членах. Для того, чтобы не было искажения результатов эксперимента и результаты не зависели от блоковых эффектов число центральных точек должно быть выбрано подходящим образом.

4.9. Критерии оптимальности планов

Минимизацию чувствительности математической модели процесса к случайным факторам проведем с учетом дисперсионных оценок.

Предположим, что вид модели известен. Требуется найти лучшие оценки коэффициентов уравнения регрессии.

Если это уравнение линейное относительно параметров, то оценки уравнений регрессий определяются из системы нормативных уравнений

$$X^T W X b = X^T W Y,$$

где $X = \{X_{ji}\}$ – матрица независимых переменных; $X^T = \{X_{ji}\}$ – матрица, получаемая транспонированием матрицы X ; $Y = Xb$ – уравнение регрессии в матричной форме; W – функция веса; и тогда оценки коэффициентов

$$b = (X^T W X)^{-1} (X^T W Y)$$

Будем предполагать, что дисперсия выходной величины Y не зависит от независимых переменных X и поэтому примем функцию веса $W = 1$.

С учетом общего вида функциональной зависимости Y от X получаем следующие оценки:

$$b = (F^T F)^{-1} F^T Y = CZ, \quad (4.13)$$

где $F = \begin{vmatrix} f_{11} & f_{12} & \dots & f_{1k} \\ f_{21} & f_{22} & \dots & f_{2k} \\ \dots & \dots & \dots & \dots \\ f_{n1} & f_{n2} & \dots & f_{nk} \end{vmatrix} = \begin{vmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{vmatrix}$, $f_{ij}(x)$ – известные функции входных пере-

менных x_1, x_2, \dots, x_n , т.е.

$$\eta_i = \sum_{j=1}^k b_j f_{ij}(x), \quad (4.14)$$

$f = (f_{i1}, f_{i2}, \dots, f_{ik})$. Обозначим $Z = F^T Y$. Если F^T – транспонированная матрица F , то

$$C = (F^T F)^{-1}$$

– ковариационная матрица. Для линейной модели $C = (X^T X)^{-1}$.

В результате действия случайных факторов экспериментатор получает математическую модель отличную от (4.14): $Y_i = \eta_i + \varepsilon_i$.

Определим чувствительность решения (4.13) к небольшим изменениям выходной величины Y , а следовательно и Z за счет случайного фактора:

$$S_{\frac{b}{z}} = \frac{db}{dz} = C \frac{dz}{dz} \quad (4.15)$$

Из выражения (4.15) следует, что чувствительность решения "b" по отношению к случайным факторам выражается ковариационной матрицей C .

План эксперимента выбирается с учетом минимизации чувствительности решения (4.13) по отношению к случайным воздействиям. Однако в том виде, в каком она предоставлена в выражении (4.15) не представляется возможным дать количественную оценку чувствительности для разных

планов F . Поэтому в качестве числовой характеристики чувствительности принимают некоторую из числовых характеристик ковариационной матрицы C . В зависимости от этого получают ряд *критериев оптимальности*.

План называется *A-оптимальным*, если он минимизирует сумму квадратов главных полуосей эллипсоида рассеяния оценок. Известно, что при нормальном законе распределения результатов наблюдений, поверхностями постоянных значений функции плотности распределения $P\left(\frac{b}{\beta}\right)$ являются эллипсоиды. Характеристики эллипсоида полностью определяются элементами матрицы C . Для двухфакторного эксперимента дадим геометрическую интерпретацию (рисунок 4.5).

Через $S_p C$ обозначена сумма диагональных элементов матрицы C

$$S_p C = \sum_{i=1}^K AC_{ii}$$

На языке матричной алгебры минимизации суммы квадратов главных полуосей эллипсоида означает минимизацию следа (суммы диагональных элементов) ковариационной матрицы уравнения регрессии. Еще можно встретить название минимизация средней дисперсии.

План называется *D – оптимальным*, если он минимизирует величину определителя матрицы C . Это соответствует минимизации обобщенной дисперсии.

E-оптимальный план минимизирует максимальную ось эллипсоида рассеяния. Или еще можно назвать минимизацией максимального собственного числа матрицы C или минимизацией максимального характеристического значения ковариационной матрицы уравнения регрессии.

G-оптимальный план обеспечивает наименьшую по всем планам максимальную величину дисперсии предсказанных значений функции отклика в области факторного эксперимента. Эти планы называют *минимаксными*.

G – план можно определить из выражения

$$d(x) = f^T(x)(F^T F)^{-1} f(x) \sigma^2$$

План будет G – оптимальным, если

$$d(x^*) = \min \max f^T(x)(F^T F)^{-1} f(x) \sigma^2$$

Определитель матрицы C является наиболее полной числовой характеристикой чувствительности. D – оптимальным план учитывает все эле-

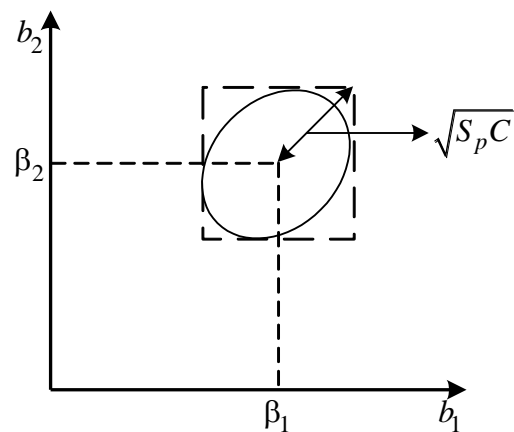


Рис. 4.5 Плоскость постоянных значений функции плотности при нормальном законе распределения

менты матрицы C . Поэтому он лучше A и E – оптимальных планов. Что касается G – оптимальным, то он эквивалентен D – оптимальному плану. D – оптимальный план минимизирует объем эллипсоида рассеяния оценок коэффициентов модели.

Таким образом, в качестве числовой характеристики оптимальной оценки модели принимаем определитель матрицы C

$$S_{\frac{b}{z}} = \det C = \det (F^T F)^{-1}$$

Критерием выбора оптимального плана является минимизация чувствительности по всем возможным планам

$$S_{\frac{b}{z}} = \min_{\text{планы}} S_{\frac{b}{z}} = \min \det (F^T F)^{-1}$$

т.к. определитель матрицы $C = F^T F$ всегда положителен.

4.10. Правило автоматической остановки имитационного эксперимента

Первый способ. Простейший способ состоит в задании требуемого количества реализаций N . Однако такой подход является грубым из-за того, что на этапе тактического планирования неизвестны распределения выходных переменных.

Второй способ состоит в задании доверительных интервалов для выходных переменных. Остановка прогона машинной модели происходит при достижении заданного доверительного интервала.

Третий способ выполняется путем двухэтапного проведения прогона, когда сначала делается пробный прогон из N_1 реализаций, позволяющий оценить необходимое количество реализаций N . Если $N_1 > N$, то прогон можно закончить, в противном случае необходимо набрать еще $(N - N_1)$ реализаций.

Четвертый способ выполняется путем последовательного анализа для определения минимально необходимого количества реализаций N , которое рассматривается при этом как случайная величина, зависящая от $N - 1$ предыдущих реализаций.

Построение линейной регрессионной модели.

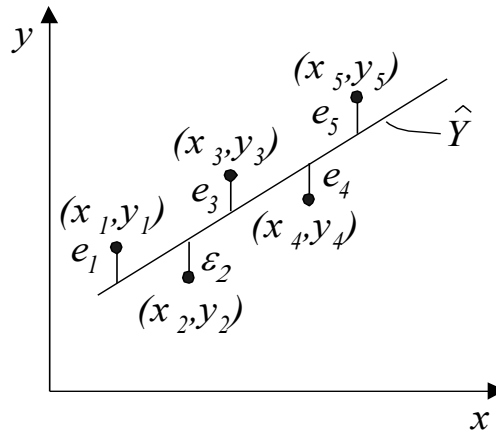


Рис. 4.6. График эксперимента

На рис. 4.6 показаны точки (x_i, y_i) , $i = \overline{1, N}$, полученные в машинном эксперименте с моделью системы. Делаем предположение, что модель результатов машинного эксперимента графически может быть представлена в виде прямой линии $\hat{Y} = b_0 + b_1 x$, где \hat{Y} – величина, предсказываемая регрессионной моделью. Требуется получить такое значение коэффициентов b_0 и b_1 , при которых сумма квадратов ошибок является минимальной.

На рис. 4.6 ошибка e_i для каждой экспериментальной точки определяется как расстояние по вертикали от этой точки до линии регрессии \hat{Y} .

$e_i = \hat{Y}_i - Y_i = b_0 + b_1 x_i - Y_i$, $\hat{O} = \sum_{i=1}^N (b_0 + b_1 x_i - Y_i)^2$ – сумма квадратов ошибок,

$$\left. \begin{aligned} \frac{\partial \hat{O}}{\partial b_0} &= 2 \left(N b_0 + b_1 \sum_{i=1}^N x_i - \sum_{i=1}^N Y_i \right) = 0 \\ \frac{\partial \hat{O}}{\partial b_1} &= 2 b_0 \sum_{i=1}^N x_i + 2 b_1 \sum_{i=1}^N x_i^2 - 2 \sum_{i=1}^N x_i Y_i = 0 \end{aligned} \right\}$$

Решая систему нормальных уравнений получим

$$b_0 = \frac{\sum Y_i \sum x_i^2 - \sum x_i \sum x_i Y_i}{N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2},$$

$$b_1 = \frac{N \sum x_i Y_i - \sum x_i \sum Y_i}{N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2},$$

N – число реализаций при моделировании системы.

Мерой ошибки регрессионной модели служит среднеквадратическое отклонение

$$\sigma_e = \sqrt{\frac{\sum_{i=1}^N e_i^2}{N-2}} = \sqrt{\frac{\sum_{i=1}^N (b_0 + b_1 X_i - Y_i)^2}{N-2}},$$

Для нормальных распределенных процессов приблизительно 67% точек находится в пределах одного отклонения σ_e от линии регрессии – труба A , и 95% точек – в пределах $2\sigma_e$ – труба B (рисунок 4.7).

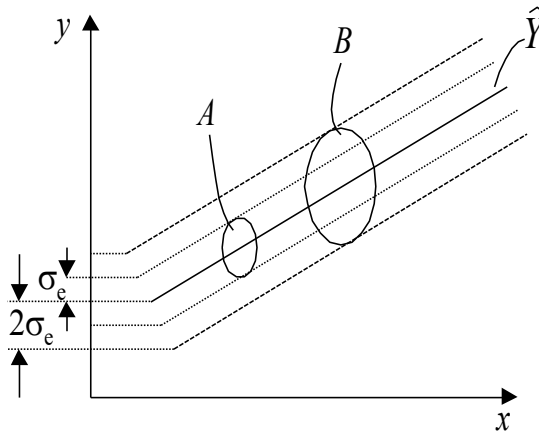


Рис. 4.7. Пределы отклонения экспериментальных точек

Для проверки точности оценок b_0, b_1 и адекватности модели используются критерии Стьюдента и Фишера.

4.11. Принятие решений после построения модели процесса

Остановимся в начале на принятии решения, когда линейная модель адекватна.

При этом возможны три варианта:

1. Все коэффициенты регрессии значимы.
2. Часть коэффициентов регрессии значима, а часть незначима.
3. Все коэффициенты регрессии незначимы.

Для первого варианта могут быть приняты следующие решения: окончания исследования, переход к планам второго порядка и движение по градиенту. Переход ко второй серии планирования для модели второго порядка позволяет получить экстремум параметра.

Во втором варианте приходится выдвигать гипотезу, что влияние данных эффектов незначимо. Причиной не значимости коэффициентов может быть неудачный выбор интервалов варьирования и включение второстепенных факторов, большая ошибка опыта и т.д.

Для этого либо расширяют интервалы варьирования по факторам, либо увеличивают число параллельных опытов.

И последний вариант, линейная модель адекватна, все коэффициенты не значимы. Это может произойти вследствие большой ошибки эксперимента или узких интервалов варьирования.

Вопросы для самопроверки

1. Каковы характерные особенности машинного эксперимента по сравнению с другими видами экспериментов?

2. Какие виды факторов бывают в имитационном эксперименте с моделями систем?
3. Что называется полным факторным экспериментом?
4. Какова цель стратегического планирования машинных экспериментов?
5. Какие проблемы стратегического планирования машинных экспериментов с моделями систем являются основными?
6. Какова цель тактического планирования машинных экспериментов?
7. Что называется точностью и достоверностью результатов моделирования систем на ЭВМ?
8. Как повысить точность результатов статистического моделирования системы в условиях ограниченности ресурсов инструментальной ПК?

5. МЕТОДЫ ОПРЕДЕЛЕНИЯ ХАРАКТЕРИСТИК МОДЕЛИРУЕМЫХ СИСТЕМ

5.1. Измеряемые характеристики моделируемых систем

При имитационном моделировании можно измерять значения любых характеристик, интересующих исследователя. Обычно по результатам вычислений определяются характеристики всей системы, каждого потока и устройства.

Для всей системы производится подсчёт поступивших в систему заявок, полностью обслуженных и покинувших систему заявок без обслуживания по тем или иным причинам. Соотношения этих величин характеризует производительность системы при определённой рабочей нагрузке.

По каждому потоку заявок могут вычисляться времена реакций и ожидания, количества обслуженных и потерянных заявок. По каждому устройству определяется время загрузки при обслуживании одной заявки M число обслуженным устройством заявок, время простоя устройства в результате отказов и количество отказов, возникших в процессе моделирования, длины очередей и занимаемые ёмкости памяти.

При статистическом моделировании большая часть характеристик – это случайные величины. По каждой такой характеристике у определяется N значений, по которым строится гистограмма относительных частот, вычисляется математическое ожидание, дисперсия и моменты более высокого порядка, определяются средние по времени и максимальные значения. Коэффициенты загрузки устройств вычисляются по формуле:

$$\rho_k = V_k * N_{ok} / T_m$$

где V_k – среднее время обслуживания одной заявки k -ым устройством; N_{ok} – количество обслуженных заявок устройством за время моделирования T_m .

Определение условий удовлетворения стохастических ограничений при имитационном моделировании производится путём простого подсчёта

количества измерений, вышедших и не вышедших за допустимые пределы.

5.2. Расчёт математического ожидания и дисперсии выходной характеристики

В случае стационарного эргодического процесса функционирования системы вычисление $M(y)$ и $D(y)$ выходной характеристики y производится усреднением не по времени, а по множеству $N_{\text{знач.}}$ измеренных по одной реализации достаточной длительности. В целях экономия ОЗУ ПК $M(y)$ и $D(y)$ вычисляются по рекуррентным формулам:

$$m_n = m_{n-1} \cdot (n-1)/n + y/n;$$

где m_{n-1} – математическое ожидание, вычисленное на предыдущем шаге.

$$d_n = d_{n-1} \cdot (n-2)/(n-1) + 1/n \cdot (y_n - m_{n-1})^2$$

где d_{n-1} – дисперсия, вычисленная на предыдущем шаге.

При большом количестве измерений эти оценки являются состоятельными и несмещёнными.

5.3. Расчёт среднего по времени значения выходной характеристики

Например, средняя длина очереди к каждому устройству вычисляется по формуле:

$$l_N = \sum_{i=1}^N l_i \tau_i / T_m$$

где i – номер очередного изменения состояния очереди (занесение заявки в очередь или исключение из очереди); N – количество изменений состояния очереди; τ_1 – интервал времени между двумя последними изменениями очереди.

Ёмкость накопитель:

$$q_n = \sum_{i=1}^N q_i \tau_i / T_m$$

где q_i – ёмкость накопителя, занятая в интервале между двумя последними обращениями к накопителю для ввода-вывода заявки.

5.4. Построение гистограммы для стационарной системы

Эмпирическая плотность Γ распределения вероятностей. Задаются границы изменения интересующей характеристики $y_i \rightarrow [y_i; y_a]$, числом интервалов N_g . Определяется ширина интервала $\Delta = (y_i - y_a) / N_g$.

Затем в процессе моделирования по мере появления значений y_i определяется число попаданий этой случайной величины в каждый из интервалов R_i гистограммы. По этим данным вычисляется относительная частота

по каждому интервалу: $G_i = R_i / (n \cdot \Delta)$, где N – общее число измерений y . Площадь гистограммы равна единице, равна сумме площадей:

$$S_i = \sum_1^{N_g} G_i \cdot \Delta = \sum \frac{R_i}{N \cdot \Delta} \cdot \Delta = \sum \frac{R_i}{N} = 1, \quad \text{т.к. } N = \sum_1^{N_g} R_i$$

При необходимости выдвигается гипотеза о том, что эмпирическое распределение согласуется с некоторым теоретическим распределением. Эта гипотеза проверяется по тому или иному критерию. Например, при использовании критерия χ^2 в качестве меры расхождения используется выражение

$$\chi^2 = \frac{\sum_1^{N_g} (R_i - N \cdot P_i)^2}{N \cdot P_i};$$

где P_i – определяется из выбранного теоретического распределения вероятность попадания случайной величины в i -ый интервал.

$$P_i = \int_{y_i}^{y_{i+1}} \varphi(x) dx = F(y_{i+1}) - F(y_i).$$

Из теоремы Пирсона следует, что для любой функции распределения $F(y)$ случайной величины y при $N \rightarrow \infty$ распределения величины χ^2 имеет вид:

$$M_k(z) = P(\chi^2 < z) = \frac{1}{2^{k/2} \cdot \Gamma(k/2)} \int_0^z t^{k/2-1} \cdot e^{-t/2} dt,$$

где z – значение случайной величины χ^2 , $k = N_g - (r + 1)$ – число степеней свободы распределения χ^2 ; r – количество параметров теоретического распределения, $\Gamma(k/2)$ – гамма функция.

Функция распределения χ^2 табулирована. По вычисленному значению χ^2 и числу степеней свободы с помощью таблиц определяется вероятность $P(\chi^2 < Z)$. Если она превышает заданный уровень значимости C , то выдвинутая гипотеза принимается.

5.5. Моделирование случайных воздействий

Важной задачей в практике имитационного моделирования систем на ПК является расчёт случайных величин. В языках программирования существуют датчики равномерно распределённых псевдослучайных величин в интервале $\{0,1\}$. Остановимся на вопросах преобразования последовательности псевдослучайных величин $\{X_i\}$ в последовательности $\{Y_i\}$ с заданным законом распределения и моделировании различных случайных событий.

Рассмотрим особенности моделирования случайных событий.

Пусть имеются случайные числа x_i , т.е. возможные значения случайной величины ξ , равномерно распределённой в интервале $\{0,1\}$. Необходимо реализовать случайное событие A , наступающее с заданной вероятностью P . Определим A как событие, состоящее в том, что выбранное значение x_i удовлетворяет неравенству:

$$x_i \leq P \quad (5.1)$$

Тогда вероятность события A будет : $P(A) = \int_0^P dx = p$. Противоположное событию A состоит в том, что $x_i > p$. Тогда $P(\bar{A}) = 1 - P$. Процедура моделирования состоит в этом случае в выборе значений x_i и сравнение их с P . При этом, если условие (5.1) удовлетворяется, то исходом испытания будет событие A .

Таким же образом можно рассмотреть группу событий. Пусть A_1, A_2, \dots, A_n – полная группа событий, наступающая с вероятностями P_1, P_2, \dots, P_n соответственно. Определим A_m как событие, состоящее, в том, что выбранное значение x_i случайной величины ξ удовлетворяет неравенству:

$$l_{m-1} < x_i < l_m, \text{ где } l_k = \sum_{i=1}^k P_i. \quad (5.2)$$

Тогда $P(A_m) = \int_{l_{m-1}}^{l_m} dx = P_m$. Процедура моделирования испытаний в этом

случае состоит в последовательности сравнений случайных чисел x_i со значениями l_k . Исходом испытания оказывается событие A_m , если выполняется условие (5.2). Эту процедуру называют определением исхода по жребию в соответствии с вероятностями P_1, P_2, \dots, P_n .

При моделировании систем часто необходимо осуществить такие испытания, при которых искомый результат является сложным событием, зависящим от 2-х и более простых.

Пусть например, независимые события A и B имеют вероятности наступления P_A и P_B . Возможными исходами совместных испытаний в этом случае будут события $AA, \bar{A}B, A\bar{B}, \bar{A}\bar{B}$ с вероятностями $P_A P_B, (1 - P_A) \cdot P_B, P_A \cdot (1 - P_B), (1 - P_A) \cdot (1 - P_B)$. Для моделирования совместных испытаний можно использовать последовательную проверку условия (5.1). Он требует двух чисел x_i .

Рассмотрим случай, когда события A и B являются зависимыми и наступают с вероятностями P_A и P_B . Обозначим через $P(B/A)$ условную вероятность события B при условии, что событие A произошло. Считаем, что

$P(B/A)$ задана. Из последовательности случайных чисел $\{X_i\}$ извлекается определённое число x_m и проверяется справедливость неравенства $x_m < P_A$. Если это неравенство справедливо, то наступило событие A . Для испытания, связанного с событием B используется вероятность $P(B/A)$. Из совокупности чисел $\{X_i\}$ берётся очередное число x_{m+1} и проверяется условие $x_{m+1} \leq P(B/A)$. В зависимости от того выполняется или нет это неравенство, исходом испытания является AB или \overline{AB} . Если неравенство $x_m < P_A$ не выполняется, то наступило событие \overline{A} . Поэтому для испытания, связанного с событием B необходимо определить вероятность:

$$P(B/\overline{A}) = [P(B) - P(A) / P(B/A)] / (1 - P(A))$$

Выберем из совокупности $\{X_i\}$ число x_{m+1} и проверим справедливость неравенства $x_m \leq P(B/\overline{A})$. В зависимости от того, выполняется оно или нет, получаем исходы испытания \overline{AB} .

5.6. Преобразование случайных величин

Дискретная случайная величина η принимает значения $y_1 \leq y_2 \leq y_3 \dots y_i$ с вероятностями P_1, P_2, \dots, P_i составляющими дифференциальное распределение вероятностей:

$$\begin{array}{ccc} y & y_1 \ y_2 \dots \dots y_j \dots & \\ P(\eta = y) & P_1, P_2 \dots \dots P_j \dots & \end{array} \quad (5.3)$$

При этом интегральная функция распределения $F_\eta(y) = P(\eta \leq y) = \sum_{j=1}^m P_j$;

$y_m \leq y_{m+1}; m = 1, 2, \dots$

$$F_\eta(y) = 0, y < y_1. \quad (5.4)$$

Для получения дискретных случайных величин можно использовать метод обратной функции. Если ξ – равномерно распределённая на интервале $(0, 1)$, случайная величина η получается с помощью преобразования

$$\eta = F_\eta^{-1}(\xi), \quad (5.5)$$

где F_η^{-1} – функция, обратная F_η .

Алгоритм вычисления по (5.4) и (5.5) сводится к выполнению следующих действий:

$$\begin{array}{l} \text{если } x_1 < P_1 \text{ то } \eta = y_1 \text{ иначе,} \\ \text{если } x_2 < P_1 + P_2 \text{ то } \eta = y_2 \text{ иначе,} \\ \dots \\ \text{если } x_j < \sum_{j=1}^m P_j \text{ то } \eta = y_m \text{ иначе} \end{array} \quad (5.6)$$

При счёте по (5.6) среднее число циклов сравнения равняется $\sum_{j=1}^m j P_j$

Вычисление непрерывных случайных величин. Непрерывная случайная величина η задана интегральной функцией распределения:

$$F_{\eta}(y) = P(\eta \leq y) = \int_{-\infty}^y f_{\eta}(y) dy,$$

где $f_{\eta}(y)$ – плотность вероятностей.

Для получения непрерывных случайных величин с заданным законом распределения, как и для дискретных величин можно использовать метод обратной функции. Взаимно однозначная монотонная функция $\eta = F_{\eta}^{-1}(\xi)$, полученная решением относительно η уравнения $F_{\eta}(y) = \xi$ преобразует равномерно распределённую на интервале $(0,1)$ величину ξ в η с требуемой плотностью $f_{\eta}(y)$.

Действительно, если случайная величина η имеет плотность распределения $f_{\eta}(y)$, то распределение случайной величины $\xi = \int_0^{\eta} f_{\eta}(y) dy$ является равномерным.

Таким образом, чтобы получить число, принадлежащее последовательности случайных чисел $\{y\}$, имеющих функцию плотности $f_{\eta}(y)$, необходимо разделить относительно y_i уравнение

$$\int_{-\infty}^{y_i} f_{\eta}(y) dy = x_j$$

5.7. Моделирование нормально распределённой случайной величины

Моделирование нормально распределённой случайной величины y может быть осуществлено на основании центральной предельной теоремы, согласно которой закон распределения суммы независимых случайных величин стремится к нормальному с увеличением числа слагаемых. Для решения некоторых задач практически сумму $y = \sum_{i=1}^n y_i$ значений, выданных с генератором случайных чисел с характеристиками $f(x_i) = 1, 0 \leq x_i \leq 1, m_x = 0.5, \sigma_x = 0.5/\sqrt{3}$.

Можно считать значениями распределённой случайной величины $y = \sum_{i=1}^n x_i$ при $n \geq 8$. Так как все слагаемые x_i имеют одинаковые математические ожидания m_x и дисперсии D_x , то $m_y = n \cdot m_x, D_y = n \cdot D_x$.

6. ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ МОДЕЛИРОВАНИЯ ИУС

6.1. CASE-технологии

CASE-технология в широком смысле представляет собой совокупность методологий анализа, автоматического проектирования, разработки и сопровождения сложных систем программного обеспечения. Аббревиатура CASE используется для двух направлений проектирования систем. Первое – Computer Aided System Engineering – направлено на решение задач концептуального проектирования сложных слабоструктурированных систем. CASE-технологии этого направления называют *системами CASE для концептуального проектирования*. Второе направление – Computer Aided Software Engineering – решает задачи автоматизированного проектирования программного обеспечения. Эти CASE-системы называют *инструментальными CASE* или инструментальными средами разработки ПО.

Среди систем CASE первого направления выделяют системы функционального и информационного (поведенческого) проектирования. Наиболее распространенной методикой *функционального проектирования* сложных систем является методика SADT (Structured Analysis and Design Technique). Эта методика стала основой стандарта IDEF0 (Integrated Definition 0). Программные средства информационного проектирования реализуют методики инфологического проектирования баз данных. Широкое распространение получила методика создания информационных моделей IDEF1X. Применение инструментальных CASE-систем позволяет сократить затраты на разработку ПО за счет уменьшения числа итераций и числа ошибок, а также улучшить качество ПО вследствие лучшего взаимопонимания разработчика и заказчика. При этом облегчается сопровождение готового программного продукта.

Под CASE-средствами обычно понимаются программные продукты, используемые как для автоматизированной разработки определенных видов моделей, например, функциональных, информационных, так и для автоматизированного создания программного обеспечения информационных и других систем. В последнем случае CASE-средства поддерживают все этапы проектирования прикладного программного обеспечения и баз данных от формулировки требований до генерации кода, тестирования, документирования и сопровождения в процессе эксплуатации.

С позиции решения задач моделирования CASE-технологии широко применяются для моделирования систем большинства предметных областей. Исключительно большое значение CASE-технологии имеют для разработки моделей деятельности предприятий, в частности функциональных и информационных, которые необходимы для решения задач системного анализа, проектирования, реинжиниринга и др.

Большинство CASE-средств основано на парадигме *методология – метод – нотация – средство*.

Методология определяет руководящие указания для оценки и выбора проекта разрабатываемого ПО, шаги работы и их последовательность, а также правила распределения и назначения методов.

Метод – это систематическая процедура или техника генерации описаний компонентов ПО, например, проектирование потоков и структур данных.

Нотации предназначены для описания структуры системы, элементов данных, этапов обработки; они включают графы, диаграммы, таблицы, блок-схемы, формальные и естественные языки.

Средства – инструментарий для поддержки и усиления методов, они поддерживают работу пользователей при создании и редактировании графического проекта в интерактивном режиме, способствуют организации проекта в виде иерархии уровней абстракции, выполняют проверки соответствия компонентов.

Основными достоинствами CASE-технологий для моделирования процессов и систем являются следующие:

- значительно сокращается время на разработку моделей;
- исследователь (разработчик, проектировщик) освобождается от рутинной работы, связанной с оформлением, представлением и хранением результатов моделирования за счет автоматизации соответствующих процессов, это позволяет ему основное внимание уделять творческой части разработки;
- разрабатываемые модели соответствуют действующим нормативным документам, их описание пригодно для широкого использования без дополнительных пояснений;
- улучшается качество создаваемых моделей за счет средств автоматического контроля;
- автоматизирован процесс развития и сопровождения результатов моделирования.

Для современных CASE-средств характерны следующие особенности.

1. CASE-технологии обеспечивают всех участников проекта, в том числе заказчиков, а иногда и пользователей, единым, строгим, наглядным и интуитивно понятным графическим языком, позволяющим получать обзорные компоненты с простой и ясной структурой. При этом компьютерные программы представляются двумерными схемами, которые более удобны в использовании, чем многостраничные описания. Эти программы позволяют заказчику участвовать в процессе разработки, проектировщикам – общаться с экспертами предметной области, а также разделять деятельность системных аналитиков, конструкторов, программистов и других участников проекта, облегчая им выполнение проекта, а также обеспечивая легкость сопровождения при эксплуатации и внесения изменений в систему.

2. В основе CASE-технологии лежит использование базы данных проекта (репозитория) для хранения всей информации о выполняемом проекте и проектах, связанных с разработкой изделий прототипов. Эта база данных может использоваться всеми разработчиками в соответствии с их правами доступа. Репозиторий включает не только информационные объекты различных типов, но и отношения между их компонентами, правила использования или обработки этих компонентов. В репозитории могут храниться объекты различных типов: структурные диаграммы, описания данных, модели данных, проекты отчетов, исходные коды, элементы данных и т. п.

3. На основе репозитория осуществляются интеграция CASE-средств, а также разделение системной информации между участниками проекта. Возможности репозитория обеспечивают несколько уровней интеграции: общий пользовательский интерфейс, передачу данных между средствами, интеграцию этапов, единую систему представления фаз жизненного цикла проекта, передачу данных и средств между различными платформами.

4. CASE-технология на основе репозитория поддерживает групповую работу над проектом, обеспечивает работу в сети в режиме удаленного доступа, экспорт-импорт любых фрагментов проекта для их развития или модификации, планирование, контроль и руководство проектом.

5. Вся документация по проекту генерируется автоматически в соответствии с требованиями действующих стандартов. При этом документация всегда соответствует текущему состоянию дел, так как любые изменения в проекте автоматически отражаются в репозитории.

6. CASE-технология обеспечивает автоматическую верификацию и контроль проекта на полноту и состоятельность на ранних этапах разработки, что влияет на конкурентоспособность создаваемого изделия.

7. Генерация программного кода, осуществляемая на основе репозитория, позволяет автоматически построить до 80–90 % текстов на языках высокого уровня.

8. CASE-технологии позволяют быстро строить макеты (прототипы) проектируемой системы, на ранних этапах разработки оценить, насколько система устраивает заказчика и приемлема для будущих пользователей.

Сопровождение системы при использовании CASE-технологий характеризуется сопровождением проекта, а не программных кодов. Средства реинжиниринга позволяют создавать модель системы из ее кодов и интегрировать полученные модели в проект, автоматически обновлять документацию при изменении кодов, автоматически изменять спецификации при редактировании кодов и т. п.

Необходимым инструментом совершенствования производственных систем и технологических процессов является функциональное моделирование бизнес-процессов.

Построение и вид функциональной модели бизнес-процессов регламентируются на международном уровне федеральными рекомендациями

США FIPS PUB 183 и стандартом IDEF0 – Integrated Definition for Process Modeling, первоначально разработанным ВВС США. В них описываются метод (язык), правила и методика структурированного графического описания бизнес-процессов.

Разработка любой сложной, в том числе программной, системы должна начинаться с функционального анализа и моделирования системы в целом и всех ее подсистем вплоть до неделимых элементов. Для этой цели разработана методология IDEF0, представляющая собой совокупность методов, правил и процедур, предназначенных для построения функциональной структуры сложных иерархических систем.

Основной принцип, заложенный в функциональное моделирование систем, состоит в их пошаговой нисходящей декомпозиции до уровня, необходимого для моделирования. При этом на всех уровнях используются функциональные блоки, принадлежащие к одному и тому же классу, который можно назвать «объект-функция». В экспертном программировании в качестве суперкласса используется объект-функция IDEF0.

Однако в отличие от математических функций, допускающих использование в качестве переменных только числовых величин, в объект-функциях могут использоваться как числовые, так и нечисловые переменные.

Все стрелки в диаграммах IDEF0 имеют метку, т. е. стрелочную надпись, в качестве которой могут использоваться либо идентификаторы, либо наименования переменных.

При построении диаграмм в IDEF0 функциональные блоки соединяются с помощью стрелок, идущих от выхода одного блока к входу и (или) управлению другого. Такая диаграмма представляет собой семантическую сеть, т. е. граф с помеченными с помощью идентификаторов или наименований вершинами (объект-функциями) и ребрами. С математической точки зрения, диаграмма эквивалентна сложной функции.

При построении функциональной модели используется метод декомпозиции, т. е. сначала описывается общее действие получения продукта (нулевой уровень), затем общее действие раскладывается на несколько основных крупных действий (первый уровень), далее каждое крупное действие описывается с помощью более мелких операций (второй уровень) и т. д. Соответственно раскладываются управления и механизмы при переходе от крупных структур к более мелким.

Важной особенностью функционального моделирования бизнес-процессов является то, что описание строится вокруг действий, а не вокруг организационной структуры. Функциональная модель показывает непосредственных участников бизнес-процессов, элементы оргструктуры фирмы, задействованные в получении продукции, работы, выполняемые различными подразделениями, и оборудование.

Построение функциональной модели рекомендуется выполнять в следующей последовательности:

- идентификация основных видов деятельности, представление их в форме иерархической структуры;
- описание входных элементов каждого процесса;
- описание преобразования входов под воздействием процесса в выходные элементы (O1, O2, ...);
- описание элементов управления (C1, C2, ...), в качестве которых могут быть инструкции, руководства, расписания, графики, стандарты и т. п.;
- указывание механизмов или ресурсов (M1, M2, ...), используемых для реализации бизнес-процессов.

Представление информационных структур и данных, используемых в функциональной модели, описывается и графически изображается с помощью информационной модели.

Информационная модель отражает структуру баз данных и информационные потоки с позиций семантики, т. е. описания данных в контексте их взаимосвязи с другими данными. Конструктивными элементами этой модели являются сущности, изображаемые блоками, отношения между сущностями, которые обозначаются линиями, соединяющими блоки, и атрибуты (имена внутри блоков).

Построение информационной модели регламентируется стандартом IDEF1X (FIPS 184) – Integrated Definition for Information Modeling.

6.2. Системы моделирования электронных устройств информационно-управляющих систем

Системы моделирования электронных устройств (ЭУ) позволяют резко уменьшить объем экспериментальных исследований, для проведения которых требуется приобретение дорогостоящих измерительных приборов, радиодеталей, трудоемкая сборка и длительная настройка макетов.

Применение программ моделирования ЭУ позволяет всесторонне исследовать разрабатываемые устройства в различных режимах работы (например, в предельно допустимых режимах), что сложно выполнить экспериментальными методами. Результаты макетирования дают ограниченный объем информации о характеристиках разрабатываемой аппаратуры. Экспериментальные исследования отражают характеристики лишь конкретных единичных макетов. Они не позволяют оценить влияние статистического разброса параметров полупроводниковых и других элементов ЭУ, и поэтому трудно делать обобщающие выводы по результатам макетирования. Экспериментально сложно определить, какие последствия вызовет наихудшее сочетание параметров радиоэлементов и что произойдет при отказе отдельных радиоэлементов. Опытным путем не просто исследовать влияние дестабилизирующих факторов, например, внешней температуры.

Перечисленные проблемы, возникающие при экспериментальных исследованиях, легко преодолеваются путем моделирования работы ЭУ.

На *этапе разработки структурной схемы* применяются такие программы, как SysCalc, SystemView, Simulink, LabVIEW, Hyper-Signal Block Diagram, Dynamo, P-CAD.

К примеру, программа SystemView представляет собой конструктор, с помощью которого из стандартных «кубиков» строится структурная схема. В имеющейся библиотеке выбирают нужный модуль, который затем с помощью мыши переносят на схему.

Детальное *исследование электронных устройств на ПК* на уровне принципиальных схем можно провести с помощью множества специальных программ: PSpice (Design Center, DesignLab), OrCAD, TangoPRO, Electronics Workbench, MicroCAP, CircuitMaker, Aplac и др.

При *проектировании устройств сверхвысокочастотного диапазона* могут быть использованы программы Super Compact, Touchstone, Libra, Microwave Office.

Одной из первых удачных программ моделирования ЭУ была программа схемотехнического моделирования SPICE (Simulation Program with Integrated Circuit Emphasis), разработанная в Калифорнийском университете для больших ПК в начале 70-х гг. XX в., а в конце 80-х гг. эта программа была адаптирована (приспособлена) для ПК и получила название PSpice.

В *системе PSpice* использован таблично-топологический метод описания схемы: выделяются узлы и указывается, какие элементы установлены между этими узлами. Затем программа автоматически составляет алгебраические и дифференциальные уравнения для описания работы моделируемого устройства.

Последние версии программы PSpice позволяют синтезировать схему из готовых графических элементов. Для этого системы комплектуются библиотеками стандартных элементов. Заметим, что разработку математических моделей радиоэлементов ведут фирмы-производители, которые в наибольшей степени заинтересованы в широком распространении своей продукции.

На базе программы PSpice были разработаны многие программы, например, Design Center и DesignLab.

DesignLab 8.0 – интегрированный программный комплекс корпорации «MicroSim», предназначенный для сквозного проектирования аналоговых, цифровых и смешанных аналого-цифровых устройств, синтеза устройств программируемой логики и аналоговых фильтров.

Программа OrCAD v.9.2 предъявляет достаточно высокие требования к компьютеру. Состав системы OrCAD 9.2:

- OrCAD Capture – графический редактор схем;
- OrCAD Capture CIS (Component Information System) – графический редактор схем, дополненный средством ведения баз данных компонентов;

при этом зарегистрированные пользователи получают через Интернет (с помощью службы ICA, Internet Component Assistant) доступ к каталогу компонентов, содержащему более 200 тыс. наименований;

- PSpice Schematics – графический редактор схем, заимствованный из пакета DesignLab;

- OrCAD PSpice A/D – программа моделирования аналоговых и смешанных аналого-цифровых устройств, данные в которую передаются как из PSpice Schematics, так и из OrCAD Capture;

- OrCAD PSpice Optimizer – программа параметрической оптимизации;

- OrCAD Layout – графический редактор печатных плат;

- OrCAD Layout Plus – программа OrCAD Layout, дополненная бессточным автотрассировщиком SmartRoute, использующим методы оптимизации нейронных сетей (используется также в системах Protel 99 SE и P-CAD 2000);

- OrCAD Layout Engineer's Edition – программа просмотра печатных плат, созданных с помощью Layout или Layout Plus, средство общей расстановки компонентов на плате и прокладки наиболее критических цепей, выполняемых инженером-схемотехником перед выдачей задания на проектирование печатной платы конструктору;

- OrCAD GerbTool – программа создания и доработки управляющих файлов для фотоплоттеров (разработка фирмы «WISE Software Solutions» специально для OrCAD, аналог программы CAM350);

- Visual CADD – графический редактор фирмы «Numera Software» (упрощенный аналог AutoCAD).

- OrCAD 9.2 функционирует на ПК с процессорами Pentium и совместимых с ними под управлением ОС Windows NT, XP.

Программа *Electronics Workbench* (фирма «Interactive Image Technologies»). В программе имеются виртуальные приборы (вольтметры, амперметры, генераторы, осциллограф, измеритель амплитудно-частотной характеристики и т. п.). Система схемотехнического моделирования Electronics Workbench предназначена для моделирования и анализа электрических схем.

Electronics Workbench может проводить анализ схем на постоянном и переменном токах. При анализе на постоянном токе определяется рабочая точка схемы в установившемся режиме работы. Результаты этого анализа не отражаются на приборах, они используются для дальнейшего анализа схемы. Анализ на переменном токе использует результаты анализа на постоянном токе для получения линеаризованных моделей нелинейных компонентов. Анализ схем в режиме АС может проводиться как во временной, так и в частотной областях.

В Electronics Workbench можно исследовать переходные процессы при воздействии на схемы входных сигналов различной формы. Программа также позволяет производить анализ цифроаналоговых и цифровых схем большой степени сложности. Имеющиеся в программе библиотеки включают в себя большой набор широко распространенных электронных компонентов. Есть возможность подключения и создания новых библиотек компонентов.

Широкий набор приборов позволяет измерять различные величины, задавать входные воздействия, строить графики. Все приборы изображаются в виде, максимально приближенном к реальному, поэтому работать с ними просто и удобно.

Результаты моделирования можно вывести на принтер или импортировать в текстовой или графический редактор для их дальнейшей обработки.

Программа Electronics Workbench совместима с программой PSpice, т. е. предоставляет возможность экспорта и импорта схем и результатов измерений в различные ее версии.

Electronics Workbench позволяет разместить схему таким образом, чтобы были четко видны все соединения элементов и одновременно вся схема целиком. Программа использует стандартный интерфейс Windows, что значительно облегчает ее применение.

В библиотеки компонентов программы входят пассивные элементы, транзисторы, управляемые источники, управляемые ключи, гибридные элементы, индикаторы, логические элементы, триггерные устройства, цифровые и аналоговые элементы, специальные комбинационные и последовательные схемы. Активные элементы могут быть представлены моделями как идеальных, так и реальных элементов. Возможно также создание своих моделей элементов и добавление их в библиотеки элементов.

В программе используется большой набор приборов для проведения измерений: амперметр, вольтметр, осциллограф, мультиметр, Боде-плоттер (графопостроитель частотных характеристик схем), функциональный генератор, генератор слов, логический анализатор и логический преобразователь.

Electronics Workbench позволяет строить схемы различной степени сложности при помощи следующих операций: 1) выбор элементов и приборов из библиотек; 2) перемещение элементов и схем в любое место рабочего поля; 3) поворот элементов и групп элементов на углы, кратные 90° ; 4) копирование, вставка или удаление элементов, групп элементов, фрагментов схем и целых схем; 5) изменение цвета проводников; 6) выделение цветом контуров схем для более удобного восприятия; 7) одновременное подключение нескольких измерительных приборов и наблюдение их показаний на экране монитора; 8) присваивание элементу условного обозначения; 9) изменение параметров элементов в широком диапазоне.

Путем настройки приборов можно: изменять шкалы приборов в зависимости от диапазона измерений; задавать режим работы прибора; задавать вид входных воздействий на схему (постоянные и гармонические токи и напряжения, треугольные и прямоугольные импульсы).

Графические возможности программы позволяют: одновременно наблюдать несколько кривых на графике; отображать кривые на графиках различными цветами; измерять координаты точек на графике; импортировать данные в графический редактор, что позволяет произвести необходимые преобразования рисунка и вывод его на принтер.

Electronics Workbench позволяет использовать результаты, полученные в программах PSpice, PCB, а также передавать результаты на Electronics Workbench в эти программы. Можно вставить схему или ее фрагмент в текстовый редактор и напечатать в нем пояснения или замечания по работе схемы.

В последнюю версию *MultiSim & Electronics Workbench* MultiSim – средство разработки и моделирования электронных схем. NI MultiSim позволяет создать схему, используя обширную библиотеку компонентов, и эмулировать поведение интегральной схемы с помощью стандартного промышленного симулятора SPICE. Начиная с версии 10.1 в MultiSim интегрирован MCU Module, позволяющий добавить в SPICE эмулированную интегральную схему микроконтроллер и запрограммировать его на C или Ассемблере. Модуль позволяет эмулировать работу интегральной схемы с микроконтроллером и различными дополнительными устройствами: RAM, ROM, клавиатурой, а также графическими и буквенно-цифровыми жидкокристаллическими дисплеями. Программа имеет большую библиотеку аналоговых и цифровых радиодеталей. В библиотеке содержатся сведения о транзисторах, диодах, резисторах, конденсаторах, триггерах, счетчиках, индикаторах и т. п.

Испытуемая схема «монтируется» на виртуальном столе, и затем делаются необходимые измерения. При этом настройка виртуальных измерительных приборов осуществляется практически так же, как и настройка реальных приборов. Создается полная иллюзия работы с конкретными действующими конструкциями.

Программа допускает возможность внесения изменений в схему прямо в процессе моделирования (например, с помощью переменных резисторов или коммутируемых клавишами переключателей). Это существенно повышает наглядность моделирования. Изменение параметров измерительных приборов (например, осциллографа) дает возможность исследовать сигналы в разных масштабах непосредственно в процессе моделирования. Это наиболее существенное отличие данной системы (другие работают в пакетном режиме моделирования и не допускают изменения параметров устройства в процессе его моделирования).

Бесплатные аналоги программы MultiSim: 1) для Windows – TinyCAD; 2) мультиплатформенные ПО – Kicad, Qucs, XCircuit; 3) для Linux – gEDA, KLogic, KTechLab.

MultiSim 11 Professional в сочетании с новым NI Ultiboard 11 – программным обеспечением для проектирования топологии печатных плат представляет собой платформу сквозного проектирования. Тесная интеграция со средой графической разработки NI LabVIEW позволяет специалистам внедрять собственные алгоритмы анализа и улучшать верификацию своих проектов.

Основные преимущества MultiSim 11:

- новый улучшенный синтаксический SPICE-анализатор, обновленные модели полевых транзисторов BSIM, поддержка дополнительных параметров, а также улучшенные возможности для моделирования цифровых устройств;

- прямое и обратное взаимодействие между схемотехническим проектом в MultiSim и топологией печатной платы в Ultiboard;

- улучшенная система связей на схемотехническом листе за счет использования специальных графических коннекторов и нового алгоритма назначения имен WYSIWYG.

Сходными возможностями обладает *программа CircuitMaker* (фирма «MicroCode Engineering Inc.»). CircuitMaker 2000 – виртуальная электронная лаборатория, предназначенная для моделирования смешанных аналого-цифровых устройств и разработки печатных плат на компьютерах под управлением ОС Windows.

Пакет программ CircuitMaker, разработанный фирмой «MicroCode», в настоящее время принадлежит компании «Titleium Limited». С его помощью производится графический ввод принципиальных электрических схем, выполнение моделирования смешанных аналого-цифровых устройств и разработка печатных плат (ПП) – фактически в одном пакете имеется компьютерная виртуальная электронная лаборатория, выполняющая цикл сквозного проектирования электронной аппаратуры широкого профиля. Программа CircuitMaker обеспечивает графический ввод принципиальных схем и их моделирование, разработка ПП производится с помощью входящего в состав пакета программы TraxMaker (в дальнейшем эта программа в соответствии с технической политикой компании «Titleium Limited» может быть заменена соответствующим модулем пакета Protel) или внешних программ PADS, P-CAD, Protel и др.

Стандартная версия (Standard Edition) обладает следующими характеристиками:

- редактор принципиальных схем;

- чертеж схемы располагается на одном или нескольких листах;

- при создании схемы используются символы стандартных электрорадиоэлементов и макромодели нестандартных элементов, создаваемых пользователями;

- имеются удобные средства для изменения параметров источников сигналов и других компонентов;

- имеются средства имитации неисправностей (короткое замыкание контактов, разрыв цепи, неверные номиналы компонентов и др.);

- списки соединений схемы составляются в форматах TraxMaker, Protel, Tango (используется в P-CAD для Windows) и OrCAD;

- в текстовых надписях символы кириллицы не допускаются;

- ограничений на количество компонентов, выводов, узлов цепей не имеется.

Программа моделирования: использует расширенную версию алгоритма SPICE 3F5 для моделирования аналоговых устройств и алгоритм XSPICE для моделирования смешанных аналого-цифровых устройств. При этом полностью поддерживаются встроенные модели и макромодели аналоговых компонентов в формате SPICE, разрабатываемые фирмами-производителями электронных компонентов и публикуемые в Интернете;

- имеет встроенные модели полупроводниковых приборов (диодов, фотодиодов, тиристоров, биполярных и полевых транзисторов, МОП- и БСИП-транзисторов (последние в виде макромоделей), оптоэлектронных приборов), электронных ламп, индикаторов, ключей, примитивов элементов цифровой логики, цифровых интегральных схем (ИС), аналоговых линейных ИС, регуляторов напряжения, линий задержки, линий передачи, устройств выполнения математических операций, источников аналоговых и цифровых сигналов, виртуальных инструментов (осциллографов, измерителей частотных характеристик, цифровых мультиметров, анализаторов логических сигналов);

- библиотека моделей содержит модели 4000 компонентов;

- выполняет стандартные виды анализа: расчет режима нелинейных схем по постоянному току (OP) и передаточных характеристик в режиме по постоянному току (DC), расчет частотных характеристик (AC) и переходных процессов (Transient), включая расчет спектров (Fourier), выполняет одновременную вариацию двух параметров (Parameter Sweep);

- предусмотрена возможность отображения текущих результатов моделирования;

- для аналоговых устройств рассчитываются токи, напряжения и другие характеристики, для цифровых – временные диаграммы логических состояний;

- имеется постпроцессор обработки результатов моделирования;

- предусмотрена «горячая» связь между схемой и экраном отображения результатов моделирования.

Профессиональная версия (Professional Edition) отличается от стандартной наличием шести дополнительных видов моделирования (расчет комплексных сопротивлений двухполюсников, вариация температуры, статистический анализ по Монте-Карло, анализ наихудшего случая, расчет передаточных функций по постоянному току, анализ уровня внутреннего шума), новыми встроенными моделями компонентов, библиотека увеличена до 6000 моделей, расширением возможностей импорта (экспорта) данных и усовершенствованием алгоритмов трассировки проводников ПП (в частности, добавлен новый волновой алгоритм с возможностями «Ripup & Retry»), библиотека корпусов компонентов расширена до 5400 типов. Кроме того, предоставляется возможность описания моделей цифровых устройств на языке высокого уровня SimCode. В ее состав включена также программа TraxCAM PRO для просмотра и редактирования Gerber-файлов.

Программный пакет схемотехнического проектирования (ППСТП) *Micro-Cap* (Microcomputer Circuit Analysis Program) рекомендуется для выполнения исследовательских работ, не предусматривающих немедленной конструкторской реализации (т. е. разводки печатной платы и оформления конструкторской документации).

ППСТП *MicroCap* представляет гибридный (аналого-цифровой) пакет синтеза и анализа электронных схем. Основные характеристики: Windows-подобный интерфейс; мощный схемный редактор; эмулятор аналоговых и цифровых элементов; анализатор осциллограмм; библиотека моделей элементов, которую можно модифицировать; функции измерения и черчения результирующих диаграмм.

Синтез схем производится как в графической, так и в текстовой форме.

Все функции ППСТП реализуются через управляющее (главное) меню. Синтезированная схема может быть подвергнута трем видам анализа:

- переходные процессы (Transient Analysis);
- на переменном токе (AC Analysis);
- на постоянном токе (DC Analysis).

Запуск вида анализа производится через команду главного меню Analysis.

Каждый вид анализа требует задания конкретных параметров. Данный процесс реализуется через соответствующие диалоговые окна установок (ДОУ).

Существует стандартный режим анализа, предполагающий на первом этапе определение количества и типов графиков, на втором – проведение самого анализа, и режим пробника (Probe), в котором сначала производится анализ с сохранением всех параметров точек схемы в виде файла на диске, а затем отображение требуемых графиков и параметров.

Micro-Cap позволяет анализировать не только аналоговые и цифровые устройства, но также проводить смешанное моделирование аналого-цифровых электронных устройств. Опытные пользователи могут также в

нестандартной ситуации создавать собственные макромодели, облегчающие имитационное моделирование без потери существенной информации о поведении системы.

Основные возможности программы Micro-Cap 9:

1. *Моделирование:*

- *Transient Analysis* – анализ переходных процессов в схемах при подаче напряжения питания и (или) воздействия (воздействий) произвольной формы с построением графиков переменных состояния схемы и их функций: зависящих от времени; зависящих друг от друга; разложенных в ряд Фурье по гармоническим составляющим;

- *AC Analysis* – анализ малосигнальных частотных характеристик схемы (линеаризованной в окрестности режима по постоянному току при воздействии на нее одного или нескольких источников гармонического сигнала с постоянной амплитудой и меняющейся частотой. При этом возможен вывод следующих графиков: зависимости комплексных значений переменных состояния (амплитуда, фаза, групповая задержка) от частоты в линейном, логарифмическом масштабах; зависимости составляющих комплексных величин переменных состояния друг от друга (например, построение годографа); зависимости спектральных плотностей напряжений шума, приведенных к указанным входному и выходному узлам, от частоты;

- *DC Analysis* – анализ передаточных характеристик по постоянному току. Возможно проведение анализа при изменении двух входных переменных, что позволяет строить на графике семейства характеристик устройства (например, семейство выходных характеристик биполярного транзистора);

- *Dynamic DC Analysis* – динамический анализ схемы по постоянному току с отображением на схеме (по выбору) напряжений, токов, мощностей, состояний полупроводниковых приборов при «ручном» изменении величин источников ЭДС, тока, сопротивлений резисторов;

- *Dynamic AC Analysis* – динамический малосигнальный анализ схемы по переменному току (линеаризованной в окрестности рабочей точки по постоянному току) с показом на схеме величин комплексных переменных состояния схемы при различных частотах (задаваемых списком) при «ручном» изменении величин пассивных компонентов (резисторы, индуктивности, конденсаторы);

- *Sensitivity Analysis* – расчет чувствительностей в режиме по постоянному току. В этом режиме рассчитываются чувствительности одной или нескольких выходных переменных к изменению одного или нескольких входных параметров (частные производные по входным параметрам). В качестве входных изменяемых параметров для этого вида анализа могут выступать все параметры моделей, величины пассивных компонентов, символьные параметры;

- *Transfer Function Analysis* – расчет малосигнальных передаточных функций в режиме по постоянному току. Рассчитывается отношение измеренного изменения заданного пользователем выходного выражения к вызвавшему это изменение малому возмущению заданного пользователем входного источника постоянного напряжения (тока). При этом автоматически рассчитываются входное (относительно клемм входного источника) и выходное (относительно узлов выходного напряжения) сопротивления схемы на постоянном токе;

- *Distortion Analysis* – расчет нелинейных искажений усилительных схем с использованием математического аппарата спектрального Фурье-анализа. Фактически в этом режиме запускается анализ переходных процессов при гармоническом воздействии на входе схемы и используются функции спектрального анализа для выходной переменной для расчета нелинейных искажений;

- *Probe Analysis* – использование графического постпроцессора PROBE, позволяющего получать различные характеристики моделируемой схемы без перезапуска того или иного вида анализа;

- *Stepping* – многовариантный анализ, при котором могут изменяться номиналы простых компонентов, величины параметров моделей компонентов, значения символьных переменных с линейным и логарифмическим шагом с возможностью выбора одновременного изменения до 20 параметров или организации до 20 вложенных циклов. Имеется возможность автоматической подписи на выходных графиках каждого варианта анализа. При использовании многовариантного анализа актуально 3D-моделирование, вызываемое из меню применяемого вида анализа. При этом по оси z откладывается значение варьируемого параметра и в пространстве строится поверхность, на которой лежат все кривые многовариантного анализа;

- *Performance Window* – возможность построения графиков зависимости определенного ряда характеристик функций схемы (функции Performance), таких как длительность нарастания/спада импульса, глобальный или локальный максимумы/минимумы, значение функций в заданных точках, ширина импульса по заданному уровню, период или частота колебаний и т. п. (всего имеется 23 функции Performance), от значения варьируемого параметра компонента или модели при многовариантном анализе;

- *Optimize* – параметрическая оптимизация схемы путем подбора номиналов или параметров модели указанных компонентов для достижения заданных значений функций Performance;

- *Monte-Carlo* – многовариантный анализ Монте-Карло при статистическом разбросе параметров компонентов с возможностью вывода гистограмм распределения характеристик заданной функции по интервалам значений.

2. Синтез аналоговых фильтров:

– в соответствии с заданными параметрами: типа фильтра (ФНЧ, ФВЧ, полосно-пропускающего – ФПП, полосно-заграждающего – ФПЗ, фазового корректора – ФК), полиномиальной аппроксимации (Батеррворта, Чебышева 1-го и 2-го родов, Бесселя, Кауэра), параметрами АЧХ (коэффициент передачи, пульсации, ослабление, полоса частот пропускания);

– синтез пассивных фильтров в виде последовательного соединения *RLC*-звеньев;

– синтез активных фильтров в виде последовательного соединения различных звеньев 2-го порядка на основе операционных усилителей (ОУ) (Саллена–Ки, с многопетлевой обратной связью MFB, Тоу–Томаса, Флейшера–Тоу, Кервина–Хьюлсмана–Ньюкомба, Аккерберга–Мосберга, звена 2-го порядка с гиратором на ОУ).

3. Создание новых моделей компонентов:

– создание моделей диодов, биполярных и полевых транзисторов, операционных усилителей, магнитных сердечников на основе справочных или экспериментальных данных с помощью встроенного оптимизатора;

– оформление моделей в виде схем-макроопределений с обозначенными выводами и списком параметров, которым затем в редакторе компонентов присваивается имя и тем самым добавляются новые компоненты, доступные для построения схем и моделирования.

4. Основные возможности обработки результатов анализа:

– панорамирование (протаскивание) активного окна графиков с помощью правой клавиши мыши; широкие возможности масштабирования полученных графиков;

– нанесение расстояния по горизонтали и по вертикали между двумя выбранными точками графика, а также значений функций, отображенных на графике, в любой точке;

– нанесение на график текстовых надписей в относительных (относительно графика выходной переменной) и абсолютных координатах;

– управление параметрами графических окон (цвет, толщина и тип линий координатной сетки и графиков, цвет, тип и размер шрифтов для текстовой информации, цвет окна, цвет заднего фона и т. п.). Возможность нанесения на графики специальных значков для их идентификации в режиме черно-белой печати;

– задание различных способов нормирования графиков;

– возможность синхронного перемещения курсоров по всем графическим окнам;

– перемещение курсоров в точки с заданными свойствами, выбранными с помощью функций Performance.

Австралийская фирма «Protel International» разработала *систему проектирования* аналоговых, цифровых и аналого-цифровых устройств *Protel 98*, обладающую примерно теми же функциональными возможностями, что и OrCAD 9.0.

Программа Aplac 7.0 предназначена для проектирования и моделирования РЭУ во временной и частотной областях. Программа помогает изучить характер изменения электрических сигналов при изменении времени наблюдения, а также зависимости выходных сигналов от частоты входного сигнала.

В состав моделируемых устройств могут входить как цифровые, так и аналоговые устройства, в том числе устройства диапазона сверхвысоких частот.

Характерной особенностью программы Aplac является наличие большого числа библиотечных элементов, из которых составляются принципиальные схемы.

Программа SPECCTRA представляет собой программу автоматического и интерактивного размещения компонентов и трассировки проводников, основанную на так называемой Shape-Based или бессеточной технологии, согласно которой все объекты ПП моделируются в виде совокупности геометрических фигур (прямоугольник, круг, дуга, трасса, полигон), которым приписаны определенные электрические и физические характеристики и правила проектирования. В отличие от привязанных к сеткам технологиям (Grid-Based), используемым, в частности, в P-CAD PRO Route, при бессеточной технологии каждый объект моделируется не набором узлов сетки, а геометрически точно, за счет чего достигается более плотный монтаж с меньшим числом слоев. Характерная особенность бессеточной технологии – меньшие затраты памяти. Другая особенность – поддержка сложных правил проектирования. Для каждого объекта, расположенного на определенном слое ПП или в определенной области или принадлежащего некоторому классу, можно задать ряд индивидуальных правил. Это означает, что при работе над сложным проектом не приходится прибегать к искусственным приемам при размещении компонентов и трассировке проводников. В интерактивном режиме разрешающая способность SPECCTRA составляет 0,01 мкм в метрической и 0,0001 мил в английской системе. Внутренняя точность представления данных еще выше. В связи с этим систему единиц можно заменять на любой фазе работы с проектом без потери точности.

Система SPECCTRA имеет два основных режима:

1-й режим. Routing – трассировка проводников.

2-й режим. Placement – размещение компонентов. Реализуется с помощью опции AP (AutoPlace) – автоматическое размещение компонентов на одной или двух сторонах ПП. В своем составе она содержит утилиту EP (EditPlace) – интерактивное размещение компонентов.

Пакет TangoPRO имеет схожий интерфейс с OrCAD и запускается в среде Microsoft Windows. Пакет TangoPRO для Windows выполняет полный цикл проектирования печатных плат, включающий графический ввод схем, упаковку схемы на печатную плату, ручное размещение компонентов, ручную и (или) автоматическую трассировку проводников, контроль ошибок в

схеме и печатной плате и выпуск документации. Поддерживается механизм переноса изменений печатной платы на схему и наоборот (Engineering Change Order – ECO). Применяется как английская, так и международная метрические системы единиц, причем изменить систему единиц можно на любой стадии работы с проектом, что, безусловно, облегчает работу с компонентами различных фирм (в других пакетах, например в P-CAD, система единиц устанавливается до начала работы с проектом и в дальнейшем не может быть изменена). Применение 32-разрядной программы повышает точность измерения линейных размеров от 0,01 мм и угловых размеров от 0,1° (в пакете P-CAD разрешающая способность линейных размеров выше, для угловых размеров составляет 1°). Поддержка текстовых форматов DXF и PDIF описания баз данных позволяет обмениваться информацией с такими распространенными пакетами, как AutoCAD, P-CAD, OrCAD, ViewLogic и др.

TangoPRO состоит из следующих модулей:

1. *TangoPRO Schematic* – графический редактор принципиальных схем. Имеет удобную систему выпадающих меню, наиболее часто встречающиеся команды привязаны к пиктограммам. Очень удобно, что при размещении на схеме символа компонента в окне выводится его изображение. При этом для цифровых интегральных схем имеется три варианта графики: Normal – нормальная (в стандарте США), DeMorgan – обозначение логических функций, IEEE – в стандарте Института инженеров по электротехнике и электронике (изображение, наиболее близкое к отечественным стандартам).

В пакете TangoPRO создаются текстовые описания схем двух типов. Во-первых, в форматах DXF и PDIF для переноса схем в другие программы, такие как AutoCAD и P-CAD (этого, к сожалению, нет в большинстве других аналогичных пакетов), и, во-вторых, создается для упаковки схем на печатных платах в пакетах TangoPRO, P-CAD и для моделирования в программах ViewLogic (формат EDIF 200) и PSpice.

2. *TangoPRO PCB* – графический редактор печатных плат. Большое достоинство пакета TangoPRO заключается в возможности передавать в него базы данных схем, печатных плат и библиотек, созданных в последних версиях P-CAD.

3. *TangoPRO Route* – автотрассировщик. В базовый комплект поставки TangoPRO входит простейший автотрассировщик печатных проводников QuickRoute, использующий стандартный алгоритм трассировки лабиринтного типа. Он поддерживает только английскую систему единиц и однородную сетку с шагом 10; 12,5; 20 и 25 мил (1 мил = 0,001 дюйма). Трассировка производится быстро, но необходима большая ручная доработка. Дополнительно поставляются два более эффективных трассировщика TangoPRO Route и TangoPRO Route 2/4, ориентированные на платы различного уровня сложности с разным количеством слоев. TangoPRO Route

трассирует платы, содержащие до 30 слоев, не накладывая других ограничений на сложность проекта (все определяется доступными ресурсами памяти). Процесс трассировки проводников разбивается на три фазы, каждая из которых выполняется за один или несколько проходов. На ранних проходах применяется метод раздвигания и проталкивания проводников (*push and shove technology*) для быстреего завершения трассировки всей платы. На заключительном этапе минимизируются количество переходных отверстий и общая длина проводников. *TangoPRO Route 2/4* имеет те же алгоритмы трассировки, но сложность платы ограничена двумя слоями без ограничений на общее количество выводов компонентов или четырьмя слоями при ограничении количества выводов до 4000.

4. *TangoPRO Library Manager* – администратор библиотеки. Библиотеки *TangoPRO* содержат большое количество символов и корпусов компонентов. Библиотеки легко пополняются с помощью имеющихся графических редакторов, а информация о номерах выводов компонентов, логической эквивалентности выводов и т. п. координируется администратором библиотек. Тем самым исключаются ошибки несогласованного ввода этой информации, возможные в системе P-CAD.

В настоящее время для синтеза фильтров применяются программы *FilterShop* фирмы «*LinearX Systems*» (www.linearx.com), *Filter Solutions* фирмы «*Kahler Engineering Solutions*» (www.kahlereng.com) и *Filter Wiz PRO* фирмы «*Schematica Software*» (www.nstarsolutions.com).

Filter Wiz PRO – профессиональная программа структурного синтеза активных фильтров под Windows. В нее интегрированы средства синтеза аналоговых и цифровых фильтров и анализа схем их реализации. После синтеза структуры фильтра выполняется расчет его принципиальной схемы (библиотека программы содержит примерно 500 типовых вариантов) и ее оптимизация методом «проб и ошибок». В схемный редактор включены компоненты 22 типов, используемые для создания аналоговых и цифровых фильтров, а также фильтров с переключаемыми конденсаторами (схема фильтра ограничена 2000 компонентами). В результате синтеза фильтров рассчитываются их частотные и переходные характеристики, карты нулей и полюсов передаточных функций, спектральные плотности внутренних шумов.

6.3. Моделирование полупроводниковых преобразователей

Типы полупроводниковых преобразователей, применяемых в компьютеризированных электроприводных системах, описаны в [22], [23]. Но кроме силовых полупроводниковых элементов в состав полупроводникового преобразователя, как правило, входят и другие элементы, к ним относятся: 1) неуправляемые и управляемые источники напряжения и тока; 2) дроссели, катушки индуктивности, конденсаторы; 3) силовые трансформаторы, измерительные трансформаторы; 4) коммутационно-защитная аппаратура

(выключатели, переключатели, автоматы, предохранители и т. д.); 5) система управления (реализуется либо на элементах интегральной микросхемотехники, либо на контроллере); 6) сигнализация режимов работы.

При расчетах электромагнитных процессов в силовых преобразователях реальную электрическую цепь представляют в виде эквивалентной схемы, состоящей из отдельных идеальных участков, каждый из которых обладает только сопротивлением, только индуктивностью или только емкостью. Например, катушку можно заменить схемой двух последовательно соединенных участков, из которых первый обладает только сопротивлением, второй – индуктивностью, емкостью же катушки часто можно пренебречь или учесть ее параллельно подключенным конденсатором.

Соотношения между напряжениями, токами и мощностями цепи подчинены законам Ома, Фарадея, Кулона, Кирхгофа и Джоуля–Ленца. Все эти законы справедливы и в цепи переменного тока для мгновенных значений напряжения, токов и мощностей, если эти значения берутся для одного момента времени. Но при переменных напряжениях и токах электрические и магнитные поля тоже переменны и будут создавать в самой цепи переменные ЭДС; эти внутренние ЭДС, так же, как и напряжение, приложенное к цепи, необходимо учитывать при применении основных законов. Таким образом, расчеты цепей сводятся по существу к моделированию эквивалентных схем с сосредоточенными параметрами, что и рассматривается в дальнейшем. В цепях переменного тока напряжения и токи непрерывно изменяют свое направление. При математическом описании электрической схемы составление уравнений осуществляется по законам Ома и Кирхгофа. При этом необходимо задаться произвольным условным направлением напряжений и токов в элементах.

Теория работы и моделирование полупроводниковых преобразователей подробно рассмотрены в [34].

6.4. Программные средства для решения задач моделирования ИУС

При решении многих задач анализа и синтеза ИУС могут использоваться различные программные средства, предназначенные для математических вычислений в технических приложениях – Matlab, Matcad, Excel, Maple и др. Система Matlab (MATrix LABoratory – матричная лаборатория, фирма «MathWorks, Inc.») создана «как язык программирования высокого уровня для технических вычислений». Система имеет открытую архитектуру, современные версии поставляются вместе с пакетом расширения Simulink. Наиболее полно функциональные возможности системы проявляются в рамках комплекса «Matlab + Simulink + пакеты расширения». Число пакетов расширения насчитывает несколько десятков.

В системе реализован принцип визуально-ориентированного программирования; уравнения состояний, описывающие динамические системы, формируются автоматически; имеются виртуальные средства регистрации

и визуализации результатов моделирования. Функции системы Matlab позволяют в интерактивном режиме выполнять сложные математические вычисления: разрабатывать алгоритмы; выполнять вычислительный эксперимент и имитационное моделирование; анализировать данные и визуализировать результаты.

Основным объектом в Matlab является массив, для которого не требуется указывать размерность явно. Это позволяет решать многие вычислительные задачи, связанные с векторно-матричными формулировками, существенно сокращая время, которое понадобилось бы для программирования на скалярных языках типа C или FORTRAN.

В Matlab включены следующие основные приложения для синтеза и исследования ИУС:

- Simulink for Windows – система моделирования;
- Optimization Toolbox – оптимизация;
- Control System Toolbox – системы управления;
- Nonlinear Control Design Toolbox – проектирование нелинейных систем;
- LMI (Linear Matrix Inequality) Control Toolbox – синтез систем управления на основе линейных матричных неравенств.

Так, для оптимизации параметров системы управления ИУС из Optimization Toolbox пакета Matlab может быть применен алгоритм наискорейшего градиентного спуска [24].

Часто для большей эффективности работы алгоритмов оптимизации в целевую функцию (критерий качества) вводят штрафную или барьерную функцию. Штрафная функция равна 0, если $\alpha \in U$, и быстро возрастает при удалении точки $\alpha \in U$ от допустимого множества.

Перечислим некоторые другие программные продукты, используемые для моделирования систем. Полные сведения об их возможностях можно получить в Интернете.

AnyLogic – графическая среда для моделирования сложных дискретно/непрерывных (гибридных) систем. Удобно, когда много движущихся объектов, они исчезают, появляются, видят друг друга, взаимодействуют и т. д.

DyMoLa – лаборатория (пакет) для динамического моделирования во временном домене механических, электронных, гидравлических, энергетических систем, в блочном представлении, с открытыми интерфейсами для интеграции и поддержкой языка Modelica.

DYNAST – ПО для расчета переходных процессов, символического и частотного анализа линеаризованных систем, описываемых системами дифференциальных и алгебраических уравнений, физическими процессами и блок-схемами. Обслуживается в пакетном режиме удаленным решателем.

EASY5 (MSC. Software.) – специализированный продукт для моделирования широкого круга сложных технических систем и устройств на

схемном уровне. Средствами этого программного продукта моделируются цифровые и аналоговые системы управления, гидроприводы, трансмиссии, двигатели внутреннего сгорания, пневматические, механические и электрические устройства, системы кондиционирования, управления аэродинамическими поверхностями, системы впрыска топлива и т. д.

Смоделированные в EASY5 системы включаются в модели, созданные в Patran и (или) Adams с целью проведения всесторонних исследований функционирования изделий на различных режимах работы и в различных условиях эксплуатации. При этом моделируется конструкция машин, механизмов, приборов и других технических объектов в совместной работе с системами управления, гидравлическими, пневматическими, электрическими системами и т. д. Для углубленного анализа и моделирования используется тесная интеграция с другими системами MSC Software., а также с системами других компаний-разработчиков: MSC Nastran, Adams, Marc, Dytran, LS-Dyna, Flight Loads & Dynamics, Fatigue, Mvision, Simdesigner, Sofy, Explore, Matlab/Simulink, dads, ricardo wave, ANSYS, Abaqus и др.

Система EASY5 является одним из важнейших компонентов современных vpd-технологий. Программный пакет EASY5 включает множество готовых математических моделей, реализованных в виде отдельных функциональных блоков (сумматоров, делителей, фильтров, интеграторов, клапанов, двигателей, теплообменников, редукторов, муфт сцепления и др.). С помощью языков C или FORTRAN можно создавать собственные функциональные блоки.

Пользователь EASY5 – моделирование изделия происходит на уровне функциональных блоков, и оптимизирует параметры разрабатываемых систем изделия. EASY5 поддерживает возможность экспорта модели (например, модели системы управления) в модель Adams или, наоборот, импорта модели механизма из Adams в EASY5. Кроме того, модели EASY5 можно встраивать в модели системы Matlab/Simulink. В состав EASY5 включен модуль matrix algebra tool, предназначенный для выполнения операций над матрицами (может использоваться для подготовки данных, анализа результатов и т. п.).

Пользователям EASY5 предложены библиотеки компонентов для многодисциплинарного анализа, включая специализированные наборы для аэрокосмической, автомобильной и других отраслей промышленности. Модуль library developer toolkit позволяет создавать собственные библиотеки компонентов.

На базе перечисленных возможностей системы EASY5 и в интеграции с другими системами MSC Software. создаются полные функциональные модели самолетов, автомобилей, танков, экскаваторов, приборов и т. д., исследуется их работа при выполнении различных задач и в различных условиях эксплуатации.

Adams – программное средство для виртуального моделирования сложных машин и механизмов. С помощью *Adams* быстро создается полностью параметризованная модель изделия: она строится непосредственно в препроцессоре или импортируется из наиболее популярных САД-систем. Задав связи компонентов модели, приложив нагрузки, определив параметры кинематического воздействия и запустив расчет, можно получить данные, полностью идентичные результатам натуральных испытаний системы.

Пользователю доступны: выявление параметров изделия, определяющих его работоспособность и точность; проверка компонентов машины на столкновения, определение габаритных размеров пространства, необходимого для ее движущихся частей; определение уровня действующих нагрузок, необходимой мощности приводов; оптимизация параметров изделия.

Adams позволяет исследовать десятки, сотни и даже тысячи вариантов конструкции, выбирать лучший, совершенствовать и совершенствовать будущее изделие, затрачивая на это во много раз меньше времени и средств, чем при традиционном подходе.

Adams может использоваться для улучшения самых разных конструкций: от простых механических и электромеханических устройств до автомобилей и самолетов, железнодорожной техники и космических аппаратов.

Adams от других систем моделирования и проектирования отличают:

- широкий набор видов кинематических связей, упругих и диссипативных звеньев, нагрузок, кинематических воздействий;
- совместимость с системами моделирования систем автоматического регулирования и управления Easy5 и Matlab/Simulink, а также пользовательскими программами, что обеспечивает моделирование и исследование сложных гетерогенных динамических систем;
- легкость в изучении и использовании – моделирование соответствует основным шагам построения физического макета (построение виртуальных прототипов, выполнение набора тестов, совершенствование конструкции);
- полная параметризация виртуальных моделей – любые параметры прототипа могут быть связаны функциональной зависимостью, модификация какого-либо размера модели автоматически приводит к изменению ее конфигурации и т. п.;
- эффективные средства визуализации результатов моделирования, включая анимацию и построение графиков.

Наряду с развитием универсальных возможностей пакета, разработчиками *Adams* созданы проблемно-ориентированные модули, обеспечивающие точное и быстрое моделирование самых сложных объектов, например, устройств автоматического управления (*Adams / Control*) и др.

В программном пакете *Adams* предусмотрена возможность учета податливости компонентов исследуемой машины (механизма). Для этого упругие характеристики частей машины определяются в конечно-

элементной системе (например, MD Nastran или MSC Nastran), а затем в специальном формате с помощью модуля Adams/Flex передаются в Adams и включаются в виртуальную модель. Использование такой модели позволяет не только учесть влияние податливости звеньев на динамику механизма, но также вычислить (с помощью модуля Adams/Durability) и отобразить на экране компьютера непосредственно в среде Adams общий уровень напряжений в упругом компоненте. Результаты расчёта деформаций упругого компонента в Adams могут быть также «переданы» в конечно-элементную систему (MD Nastran или MSC Nastran) для точного определения уровня напряжений с учётом всех особенностей её конфигурации. В дальнейшем эти данные могут быть использованы и для анализа долговечности детали с помощью системы Fatigue. Такая технология обеспечивает существенное повышение точности моделирования современных, оптимизированных по массе высокодинамичных механизмов. В программном пакете Adams обеспечивается возможность создания моделей, включающих контактное взаимодействие упругого тела с твёрдым телом, а также двух упругих тел.

Основой Adams являются системы дифференциальных уравнений, описывающих динамику исследуемого объекта.

MBTU – программа «Моделирование в технических устройствах» является отечественной разработкой с классическим интерфейсом блочного моделирования. Исследование моделей возможно как во временном, так и в частотном доменах.

Model Vision Studium – новационный инструмент для визуального интерактивного моделирования во временном домене структурно-сложных гибридных систем с применением карт поведения и объектно-ориентированного языка описаний.

Modelica – универсальный объектно-ориентированный язык для моделирования физических систем. Поддерживается пакетами: Dymola, MathModelica (Mathematica), ALLAN, 20-sim, SMILE/M.

SamSim – моделирование линейных и нелинейных цепей САУ, построение временных, частотных характеристик, фазовых портретов и годографов. ПО начального уровня, поможет студентам визуализировать процессы в любой точке модели.

SIMPLORER Simulation Center – интегрированные инструменты для моделирования и анализа систем под контролем графа конечного автомата, описанных элементами электрических и блок-схем + собственный SML-язык для компилятора.

VTB Virtual Test Bed – моделирующий комплекс для визуальной (3D) симуляции технических систем различной физической природы – энергетика, электромеханика и др. Модели описываются на языках: ACSL, Modelica, Siber, SPICE, Simulink + аппаратура ввода-вывода.

КОПРАС – комплекс программ для анализа и синтеза автоматических систем (для ТАУ). Имеет большой набор решающих, функциональных элементов и сервисных функций. Динамические модели описываются структурными схемами.

Пакет **PDELab** – решение систем нелинейных дифференциальных уравнений в частных производных. Компьютерное моделирование в физике, химии, биологии и медицине.

APM WinMachine (НТЦ «АПМ») – CAD/CAE-система автоматизированного расчета и проектирования механического оборудования и конструкций в области машиностроения. Эта система в полном объеме учитывает требования государственных стандартов и правил, относящихся как к оформлению конструкторской документации, так и к расчетным алгоритмам.

Комплекс APM WinMachine позволяет решать следующие задачи:

- проектирование и анализ механического оборудования и его элементов с использованием инженерных методик;
- анализ напряженно-деформированного состояния трехмерных объектов любой сложности при произвольном закреплении, статическом или динамическом нагружении;
- создание конструкторской документации;
- проектирование и расчет технологических процессов.

В состав APM WinMachine входят следующие APM:

1. **APM Graph** – двумерный графический редактор, который используется для оформления графической части конструкторской документации в различных областях техники, науки, в архитектуре и строительстве. Модуль имеет стандартный набор инструментов для оформления чертежей, а также инструменты параметризации объектов и расчетов размерных цепей.

2. **APM Studio** – модуль моделирования и импорта (поддерживается импорт из формата STEP) трехмерных поверхностных и твердотельных моделей с инструментами указания опор и приложения различных нагрузок и встроенным генератором разбиения на конечно-элементную сетку.

3. **Конечно-элементный анализ. APM Structure3D** – модуль предназначен для расчета напряженно-деформированного состояния стержневых, пластинчатых, оболочечных и твердотельных конструкций, а также их произвольных комбинаций.

Модуль APM Structure3D позволяет решать следующие задачи: 1) сопределение полей эквивалентных напряжений и их составляющих; 2) расчет линейных, угловых и результирующих перемещений; 3) определение внутренних усилий; 4) расчет устойчивости и формы потери устойчивости;

5) определение частот собственных колебаний и собственных форм; 6) расчет вынужденных колебаний и анимация колебательного процесса по заданной вынуждающей нагрузке, расчет на вибрацию оснований; 7) расчет

температурных полей и термонапряжений; 8) расчет усталостной прочности; и др.

4. *APM Dynamics* – модуль предназначен для кинематического анализа рычажных механизмов. Результатами расчета модели являются траектории перемещения, графики скоростей, ускорений, силовых факторов в узловых точках и анимация работы механизма.

5. *APM Beam*. С помощью модуля APM Beam можно выполнить расчет балки и подбор ее наиболее подходящего поперечного сечения.

6. *Инженерный анализ. APM Bear* – модуль расчета и проектирования неидеальных подшипников качения. С помощью APM Bear можно рассчитать: перемещения (жесткость); долговечность; момент трения; наибольшие контактные напряжения; потери мощности; тепловыделение; силы, действующие на тела качения.

7. *APM Shaft* – модуль проектирования валов и осей. Модуль APM Shaft позволяет выполнить весь цикл проектирования валов и осей, начиная от разработки конструкции и заканчивая статическим и динамическим анализом.

8. *APM Trans* – модуль проектирования и проверки механических передач вращения. С помощью APM Trans можно проектировать и проверять следующие типы передач: 1) цилиндрические с прямым зубом как внешнего, так и внутреннего зацепления; 2) цилиндрические с косым зубом внешнего зацепления; 3) шевронные; 4) конические с прямым и круговым зубьями; 5) червячные, с архимедовым, эвольвентным и конволютным червяком и глобоидные; и др.

9. *APM Drive* – модуль автоматизированного проектирования привода вращательного движения произвольной структуры.

10. *APM Spring* – модуль проектирования упругих элементов машин.

11. *APM Joint* – модуль проектирования соединений элементов машин. Модуль позволяет выполнить комплексный расчет и анализ соединений, которые наиболее часто используются в машиностроении.

12. *APM Plain* – модуль проектирования и расчета подшипников скольжения.

13. *APM Cam* – модуль проектирования кулачковых механизмов.

14. *APM Screw* – модуль проектирования неидеальных винтовых передач. С помощью APM Screw могут быть рассчитаны наиболее широко распространенные типы винтовых передач: винтовая передача скольжения, шариковая винтовая передача и планетарная винтовая (роликовая винтовая) передача. Модуль предназначен для комплексного анализа винтовых передач и расчета их основных параметров.

15. *Технологическая подготовка производства. APM Technology* – модуль проектирования технологических процессов.

С помощью APM Technology можно: 1) разрабатывать технологические процессы, используя справочные данные, базы данных оборудования

и технологической оснастки; 2) выполнять расчеты режимов обработки и норм времени; 3) оформлять технологическую документацию (маршрутные и операционные карты, карты технологических процессов, карты эскизов).

16. *APM Calculation Instrument* – модуль, предназначенный для выполнения инженерных (конструкторских и технологических) расчетов. Вместе с модулем поставляется *APM Technology Calculation* – база стандартных технологических расчетов, выполняемых при проектировании технологических процессов. В модуле также предусмотрена возможность добавления собственных расчетных программ (алгоритмов расчетов).

17. *Базы данных*. В состав комплекса *APM WinMachine* входят следующие базы данных: а) *APM Mechanical Data* – база данных стандартных деталей и узлов, справочных данных по общему машиностроению; б) *APM Technology Data* – база стандартных информационных данных для проектирования технологических процессов; в) *APM Material Data* – модуль хранения и редактирования параметров материалов.

Работа с существующими базами данных проходит в модуле создания и редактирования баз данных – *APM Base*. *APM Base* содержит необходимый функционал для поиска, редактирования и расширения информационного массива вышеперечисленных баз данных или составления собственных.

MD Nastran (от *MultiDiscipline Nastran*) – система, объединяющая лучшие в своем классе системы компьютерного инженерного анализа, включая *MSC Nastran*, *Marc*, *Dytran* и *LS-DYNA*, в одну полностью интегрированную суперсистему для проведения «многодисциплинарного» моделирования в масштабах предприятия.

MD Nastran дает возможность пользователям перейти от узкоспециализированных инструментов моделирования и анализа к интегрированному решению в масштабах предприятия. Использование единой расчетной модели для «многодисциплинарного» моделирования позволяет минимизировать время, обычно затрачиваемое на подготовку множества расчетных моделей, преобразование и передачу данных от одной расчетной модели в одной системе к другой модели в другой системе (решается проблема исключения вносимых при таких преобразованиях ошибок).

MD Nastran построен на базе *MSC Nastran* – системы конечно-элементного анализа. В дополнение к возможностям *MSC Nastran*, *MD Nastran* включает в себя решатель *LS-DYNA* в виде последовательности решения *SOL 700*, расширенный нелинейный статический и динамический анализ с помощью новой последовательности решения *SOL 400*, алгоритмы решения контактных задач в рамках линейного и нелинейного анализа, и т. д.

Решение *SOL 700* предназначено, преимущественно, для моделирования высокоскоростного нагружения различных изделий, включая их большие деформации и разрушение (например, соударения различных объек-

тов, технологические процессы типа штамповки,ковки и т. п.) и т. д. В SOL 700 для решения нелинейных задач применяется явная схема интегрирования. Производительность SOL 700 особенно высока при решении задач на многопроцессорных вычислительных комплексах.

Вычислительная производительность MD Nastran оптимизирована для решения больших и сверхбольших задач и выполнения комплексных, связанных видов «многодисциплинарного» анализа.

MSC Nastran – продукт MSC, обеспечивает полный набор расчетов, включая расчет напряженно-деформированного состояния, запасов прочности, собственных частот и форм колебаний, анализ устойчивости, исследование установившихся и неустойчивых динамических процессов, решение задач теплопередачи, акустических явлений, нелинейных статических и нелинейных переходных процессов, анализ сложного контактного взаимодействия, расчет критических частот и вибраций роторных машин, анализ частотных характеристик при воздействии случайных нагрузок и импульсного широкополосного воздействия, исследование аэроупругости на дозвуковых и сверхзвуковых скоростях.

В MSC Nastran моделируются практически все типы материалов, включая композитные и гиперупругие. Расширенные функции включают технологию суперэлементов (подконструкций), модальный синтез, анализ динамики сложных структур на основе метода Крейга–Бемптона, макроязык DMAP для создания пользовательских приложений.

Предусмотрена возможность передачи моделей упругих тел в Adams, которые будут использоваться в этом программном комплексе при построении расчётных моделей машин и механизмов.

MSC Nastran располагает аппаратом автоматической оптимизации параметров, формы и топологии конструкций. Возможности оптимизации позволяют использовать MSC Nastran для автоматической идентификации компьютерной расчетной модели и планирования экспериментов.

В составе MSC Nastran имеются специальные возможности моделирования динамики роторных машин, что обуславливает незаменимость этого программного продукта в отраслях, связанных с разработкой турбомашин.

MSC Nastran предоставляет возможности расчёта характеристик работы конструкций из композиционных материалов.

MSC Nastran используется для планирования экспериментов (определение мест расположения датчиков) и оценки полноты полученных экспериментальных данных. С помощью MSC Nastran решаются задачи моделирования систем управления, систем терморегулирования с учетом воздействия этих систем на конструкцию.

Тесная связь MSC Nastran с системой Patran обеспечивает полностью интегрированную среду для моделирования и анализа результатов. Все ведущие производители пре- и постпроцессоров, а также систем автоматизи-

рованного проектирования предусматривают прямые интерфейсы с этой системой.

MSC Nastran работает на ПК, рабочих станциях, кластерах, суперкомпьютерах и предусматривает возможности векторной и параллельной обработки данных на вычислительных комплексах, которые поддерживают эти функции.

Patran – это интегрирующая среда систем проектирования, моделирования и анализа на основе современного графического пользовательского интерфейса.

Patran является интегрирующей средой и графической оболочкой для MD Nastran, MSC Nastran, Marc, Dytran, Sinda и других конечно-элементных систем анализа. Patran обеспечивает импорт геометрических моделей из CAD-систем, создание расчетных моделей, запуска их на расчет, графическое отображение и обработку полученных результатов.

Помимо доступа к стандартным геометрическим форматам, таким как Parasolid, ACIS, STEP, IGES и др., предоставляет прямой доступ к программным пакетам автоматизированного проектирования: CATIA V4 и V5, NX, Pro/ENGINEER.

Возможность использования данных о материалах обеспечивается через прямой интерфейс с системой Mvision.

При использовании Patran, как правило, именно CAD-геометрия становится основой конечно-элементной модели. Поэтому Patran включает в себя специальные методы и функции для контроля и автоматизированного исправления дефектов, импортированных из CAD-систем геометрических моделей, а также преобразования их в расчетные модели, что при современном уровне сложности и подробности компьютерных моделей существенно облегчает работу конструкторов и расчетчиков.

Patran включает в себя обширные функции создания и модификации геометрических моделей, в том числе твердотельное моделирование (включая Булевы операции), создание срединных поверхностей, автоматическое распознавание и параметризацию отверстий, скруглений и фасок.

Наличие инструментария для создания, редактирования и контроля качества конечно-элементных (КЭ) сеток позволяет в короткое время создавать оптимальные расчетные модели. Наряду со стандартными генераторами КЭ-сеток в Patran присутствуют специализированные генераторы: SheetBodyMesher, AdvancedSurfaceMesher и др., позволяющие создавать высококачественные КЭ-сетки на наборе поверхностей «плохого» (с точки зрения генерации КЭ-сеток) качества.

Patran предоставляет обширный набор возможностей для задания нагрузок, граничных условий, свойств материалов и элементов, параметров расчета, а также для визуализации, обработки и преобразования результатов счета. Кроме того, широкие возможности языка Patran Command

Language (PCL) позволяют адаптировать все упомянутые функции к конкретным требованиям пользователя.

Функции «Группы», «Списки», «Суперэлементы», «Области» и другие дают возможность создавать и обрабатывать полноразмерные подробные модели сложных изделий, таких как самолет, автомобиль и т. д., включающие миллионы конечных элементов.

Наряду с интерфейсами к САД- и САЕ-системам Patran имеет ряд специальных встроенных приложений:

- THERMAL – анализ тепловых процессов и гидравлических сетей;
- LAMINATE MODELER – подготовка моделей и обработка результатов расчетов для конструкций из композитных материалов;
- ANALYSIS MANAGER – многофункциональная система, позволяющая управлять процессами выполнения заданий.

Autodesk Algor Simulation (Autodesk) обеспечивает решение широкого спектра задач механики деформируемого твердого тела, гидродинамики, электростатики, а также связанных задач.

Autodesk Algor Simulation – инструмент расчета деталей и сборок конструкций, который теперь тесно интегрирован с Moldflow 2011, что позволяет инженерам пользоваться результатами анализа и обширной библиотекой материалов последнего, чтобы принимать важные решения на более ранней стадии проекта.

Разнообразие типов конечных элементов, богатые возможности для построения конечно-элементных сеток и встроенная в Autodesk Algor Simulation библиотека моделей материалов позволяют создавать цифровые прототипы, эквивалентные реальным изделиям. С большинством САД-программ поддерживается прямой ассоциативный обмен данными, что обеспечивает возможность интерактивно вносить изменения в проект (в САД-системе) без необходимости снова определять данные моделирования (в Autodesk Algor Simulation).

Autodesk Algor Simulation поддерживает прямой импорт геометрии из Autodesk Inventor, Autodesk Mechanical Desktop, IronCAD, KeyCreator, Pro/ENGINEER, Rhinoceros, Solid Edge, SolidWorks и импорт моделей в универсальных форматах: ACIS, STEP, IGES, STL.

Marc 2008 (MSC Software) представляет собой универсальную конечно-элементную программу для проведения углубленного анализа высоконелинейного поведения конструкций и решения задач теплопередачи. Marc также используется для компьютерного моделирования технологических процессов прокатки, прессования, листового и объемного формования, производства шин, суперпластического формования и т. д. В Marc доступны также неструктурные типы анализа: 1) электростатический; 2) магнито-статический; 3) электромагнитный; 4) акустический; 5) гидродинамический анализ подшипников; 6) теплопередача и ряд других.

Дополняя возможности MSC Nastran и Dytran, программа позволяет проводить комплексный анализ ситуаций, когда элементы конструкции испытывают большие перемещения и повороты, а свойства материалов существенно нелинейны. Также возможен эффективный анализ сложного контактного взаимодействия конструкций.

Marc обладает широкими возможностями решения сложных нелинейных задач. Так, для изучения поведения резиновых уплотнений, строительных конструкций и некоторых других специальных объектов требуются особые модели материала. Контакт зубьев шестерен может быть корректно представлен лишь в том случае, если модель включает в себя соответствующие алгоритмы анализа контакта.

Множество специальных типов анализа в среде Marc поддерживается полным набором конечно-элементных формулировок. Программа предусматривает применение пользовательских подпрограмм, с помощью которых упрощается моделирование поведения изделия в особых ситуациях.

AFEA – интегрированное персональное рабочее место на основе комплекса «Patran – Marc». В единой среде графического интерфейса пре- и постпроцессора Patran пользователь может формировать задачи, осуществлять их расчет и обрабатывать результаты.

AFEA – комплекс на базе решателя Marc, ориентированный на решение физически и геометрически нелинейных задач механики, а также задач теплопередачи, включая связанные задачи теплопрочности. Функциональные возможности решателя Marc (полный набор моделей материала, автоматический трехмерный контакт, расчет больших пластических и упругих деформаций, циклическое нагружение, глобальное перестроение сетки, перенос решения из двумерного случая в трехмерный и т. д.) делают этот продукт высокоэффективным средством исследования сложного поведения различных конструкций.

SimDesigner – семейство встроенных решений для САД-системы CATIA V5, обеспечивающее моделирование динамики механизмов и быстротекущих по времени существенно нелинейных процессов, расчет напряженно-деформированного и теплового состояний элементов конструкции.

6.5. MATLAB Simulink и его приложения

Для моделирования, имитации и анализа динамических систем применяется интерактивный инструмент Math Works MATLAB Simulink. Он позволяет при помощи графических блок-диаграмм моделировать динамические системы, исследовать их работоспособность. Simulink полностью интегрирован с MATLAB, обеспечивая немедленный доступ к широкому спектру инструментов анализа и проектирования. Simulink также интегрируется с приложением MATLAB Stateflow для моделирования поведения, управляемого событиями. Эти преимущества делают Simulink наиболее

популярным инструментом для проектирования систем управления и коммуникации, цифровой обработки и других приложений.

Simulink Performance Tools содержит четыре инструмента для пользователей Simulink, проектирующих крупномасштабные, сложные модели: Simulink Accelerator ускоряет выполнение кода при имитации моделей от двух до десяти раз; Model Differencing графически подчеркивает отличия между двумя моделями; Model Profiler собирает данные функционирования и генерирует HTML-отчет; Model Coverage выдает отчет о возможных путях имитации модели.

Для мониторинга сигналов и параметров в Simulink-моделях применяется приложение Dials & Gauges Blockset.

При создании и настраивании отчетов из MATLAB, Simulink и Stateflow моделей и данных в различных выходных форматах, включая HTML, RTF, XML и SGML, применяются приложения MATLAB Report Generator и Simulink Report Generator. Это дает возможность автоматически документировать создаваемые системы, включая в них любые данные из рабочей среды MATLAB.

MATLAB Compiler включает MATLAB C/C++ Math and Graphics Libraries. Он позволяет компилировать m-файлы в коды на C и C++. Это дает возможность создавать независимые программные продукты. Компилятор содержит новые алгоритмы оптимизации, существенно увеличивающие скорость выполнения. Модуль для Visual Studio позволяет использовать MATLAB-компилятор из Microsoft Visual Studio.

В MATLAB включены специализированные Toolbox, предназначенные для автоматизации проектирования систем управления (табл. 6.1).

Таблица 6.1

Пакет прикладных программ	Назначение
Control System	Моделирование, анализ и проектирование как непрерывных, так и дискретных во времени линейных автоматических систем управления
Nonlinear Control Design (NCD) Blockset	Проектирование нелинейных систем
Fuzzy Logic	Проектирование, моделирование и анализ систем с нечетким откликом. Он имеет средства для преобразования входных данных в выходные данные системы правил и связей произвольной сложности, выраженных обычным языком. Системы могут быть имитированы в MATLAB или включены в блочные диаграммы Simulink с возможностью генерации кода для независимого выполнения
LMI Control	Синтез систем управления на основе линейных матричных неравенств. Содержит функции для проектирования и анализа таких характеристик систем управления, как помехоустойчивость, производительность и др.
Model Predictive	Управление с эталонной моделью. Полезен для управления

Control	системами с большим количеством входных и выходных переменных, имеющих много связей
M-Analysis and Synthesis	μ -анализ и синтез. Содержит набор функций для использования в анализе и проектировании устойчивых линейных систем со многими переменными
Robust Control	Робастное управление. Содержит специализированный набор инструментов для анализа и синтеза систем управления, устойчивых по отношению к случайным возмущениям
Quantitative Feedback Theory (QFT) Control Design	Проектирование робастных систем с обратной связью
Neural Network	Нейронные сети. Применяется в областях, в которых формальный анализ чрезвычайно труден или невозможен, как, например, распознавание образов, идентификация и управление нелинейными системами
System Identification	Идентификация параметров в системах управления
Frequency Domain System Identification	Идентификация в частотной области
xPC Target	Моделирование и тестирование взаимосвязанного оборудования, управляющих систем и систем цифровой обработки сигналов
Communications	Разработка современных коммуникационных систем, включая моделирование в реальном масштабе времени

В MATLAB также включены специализированные приложения, предназначенные для автоматизации проектирования в реальном времени (табл. 6.2).

Таблица 6.2

Приложение	Назначение
Real-Time Workshop	Дополняет Simulink, позволяя автоматически генерировать С-код прямо из блочных диаграмм пакета Simulink
Real-Time Workshop Embedded Coder	Расширяет возможности приложения Real-Time Workshop, позволяя осуществлять генерацию высококачественного встраиваемого кода для Simulink-моделей, и осуществляет генерацию кода из Stateflow
Real-Time Windows Target	Обеспечивает простую в использовании среду макетирования в реальном времени, полностью контролируемую через пользовательский интерфейс Simulink
DSP Blockset	Моделирование и симуляция, позволяющие проектировать системы обработки сигналов в режиме реального времени и осуществлять расчет этих систем. DSP Blockset позволяет использовать инструменты Simulink для обработки потоковых данных и выполнения многоступенчатых операций

Таблица 6.3

Библиотека	Назначение
Sim Power Systems	Моделирование энергетических систем, включая устройства электротехники и промышленной электроники
Sim Mechanics	Моделирование механических систем, составленных из отдельных элементов, таких как механические блоки,

	соединения, узлы крепления, исполнительные элементы и датчики. С помощью этих элементов формируется описание сложной механической системы с присоединенными к ней другими динамическими моделями и контроллером
Aerospace	Моделирование, интегрирование и имитация авиационных, космических, реактивных и турбореактивных систем и подсистем в приложении Simulink
Filter Design	Проектирование, моделирование и анализ цифровых фильтров
Image Processing	Набор средств для анализа и обработки цифровых изображений

Для моделирования и исследования электромеханических систем полезными будут следующие библиотеки MATLAB (табл. 6.3).

6.6. Специализированный Toolbox Sim Power Systems

Исходя из приведенного в 6.5 состава компьютеризированных электроприводных систем для моделирования наиболее подходит Toolbox Sim Power Systems.

Toolbox Sim Power Systems имеет восемь основных разделов:

- 1) Application Library – библиотека инженерных приложений;
- 2) Electrical Sources – источники электрической энергии;
- 3) Elements – электротехнические элементы;
- 4) Extra Library – дополнительные библиотеки;
- 5) Machines – электрические машины;
- 6) Measurements – измерительные и контрольные устройства;
- 7) Phasor Elements – статический компенсатор;
- 8) Power Electronics – устройства силовой электроники.

В состав библиотеки Sim Power Systems входят модели пассивных и активных электротехнических элементов, источников энергии, электродвигателей, трансформаторов, линий электропередачи и другое оборудование. Имеется также возможность моделирования полупроводниковых устройств, включая системы управления ими.

В случае если в библиотеке нет нужной модели, пользователь имеет возможность создать свою модель как с помощью уже имеющихся в библиотеке моделей, так и на основе моделей основной библиотеки Simulink и управляемых источников тока или напряжения.

Используя Toolbox Sim Power Systems, пользователь в состоянии за короткое время создать компьютерную модель сложной электромеханической системы.

Toolbox Sim Power Systems обеспечивает широкие возможности для расчета электротехнических устройств и систем и в настоящее время может считаться одним из лучших пакетов для моделирования устройств силовой электроники.

Применение Toolbox Sim Power Systems для расчета и исследования электрических цепей и электрических машин постоянного и переменного токов подробно изложено в [29], [30].

Рассмотрим отдельные библиотеки Toolbox Sim Power Systems.

6.6.1. Источники электрической энергии

В состав библиотеки Electrical Sources входят неуправляемые и управляемые источники постоянного и переменного напряжения и тока (рис. 6.1).

В полях окна настройки параметров каждого блока устанавливаются значения амплитуды напряжения, начальной фазы и частоты.

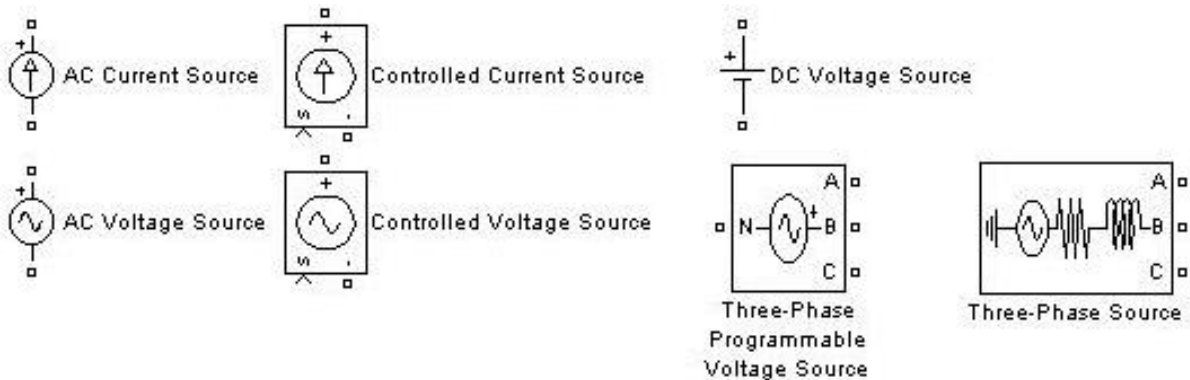


Рис. 6.1. Состав библиотеки Electrical Sources

Идеальный источник постоянного напряжения (DC Voltage Source). Модуль вырабатывает постоянное по уровню напряжение. В полях окна настройки параметров задаются: Amplitude (V) – амплитуда (В) выходного напряжения источника.

Идеальный источник переменного (синусоидального) напряжения (AC Voltage Source). В полях окна задания параметров задаются: Peak Amplitude (V) – амплитуда выходного напряжения источника; Phase (deg) – начальная фаза в градусах; Frequency (Hz) – частота источника; Sample time – параметр задает шаг дискретизации по времени выходного напряжения источника при создании дискретных моделей.

Модуль является идеальным источником напряжения, т. е. его собственное сопротивление равно нулю.

Идеальный источник переменного (синусоидального) тока (AC Current Source). В полях окна задания параметров задаются: Peak Amplitude (V) – амплитуда выходного тока источника; Phase (deg) – начальная фаза в градусах; Frequency (Hz) – частота источника; Sample time – параметр задает шаг дискретизации по времени выходного тока источника при создании дискретных моделей. Блок является идеальным источником тока, т. е. его собственное сопротивление равно бесконечности.

Управляемый источник напряжения (Controlled Voltage Source). Модуль вырабатывает напряжение в соответствии с сигналом управления.

Управляемый источник тока (Controlled Current Source). Модуль вырабатывает ток в соответствии с сигналом управления.

В полях задания этих элементов задаются начальные параметры – амплитуда, фаза и частота; выбирается тип источника (постоянного или переменного напряжения (тока)).

Трехфазный источник синусоидального напряжения (3-Phase Source). В полях окна задаются: *Phase-to-phase rms voltage (V)* – действующее значение линейного напряжения, В; *Phase angle of phase A (deg)* – начальная фаза напряжения в фазе А, ...°; *Frequency (Hz)* – частота источника, Гц; *Internal connection* – соединение фаз источника (Y – звезда; Yn – звезда с нулевым проводом; Yg – звезда с заземленной нейтралью); *X/R ratio* – отношение индуктивного и активного сопротивлений и др.

6.6.2. Электротехнические элементы

На рис. 6.2 представлен состав библиотеки Elements, которая содержит:

1. Последовательные и параллельные, одно- и трехфазные пассивные элементы R , L , C .
2. Одно- и трехфазные магнитосвязанные цепи (цепи с взаимной индукцией).
3. Линейные одно- и трехфазные трансформаторы (Linear Transformer, Three-Phase Transformer) и трансформатор с реальным магнитным сердечником, учитывающим его насыщение (Saturable Transformer).
4. Одно- и трехфазный ключ (Breaker), параметры (сопротивление, индуктивность) которого в открытом состоянии задаются в полях настройки. Там же задается состояние ключа (открыт, закрыт).
5. Трехфазный двух- и трехобмоточный трансформаторы (Three-Phase Transformer, Two windings, Three windings).
6. Модули, реализующие параметры однофазной и трехфазной линии передачи (PI Section Line, Distributed Parameters Line).

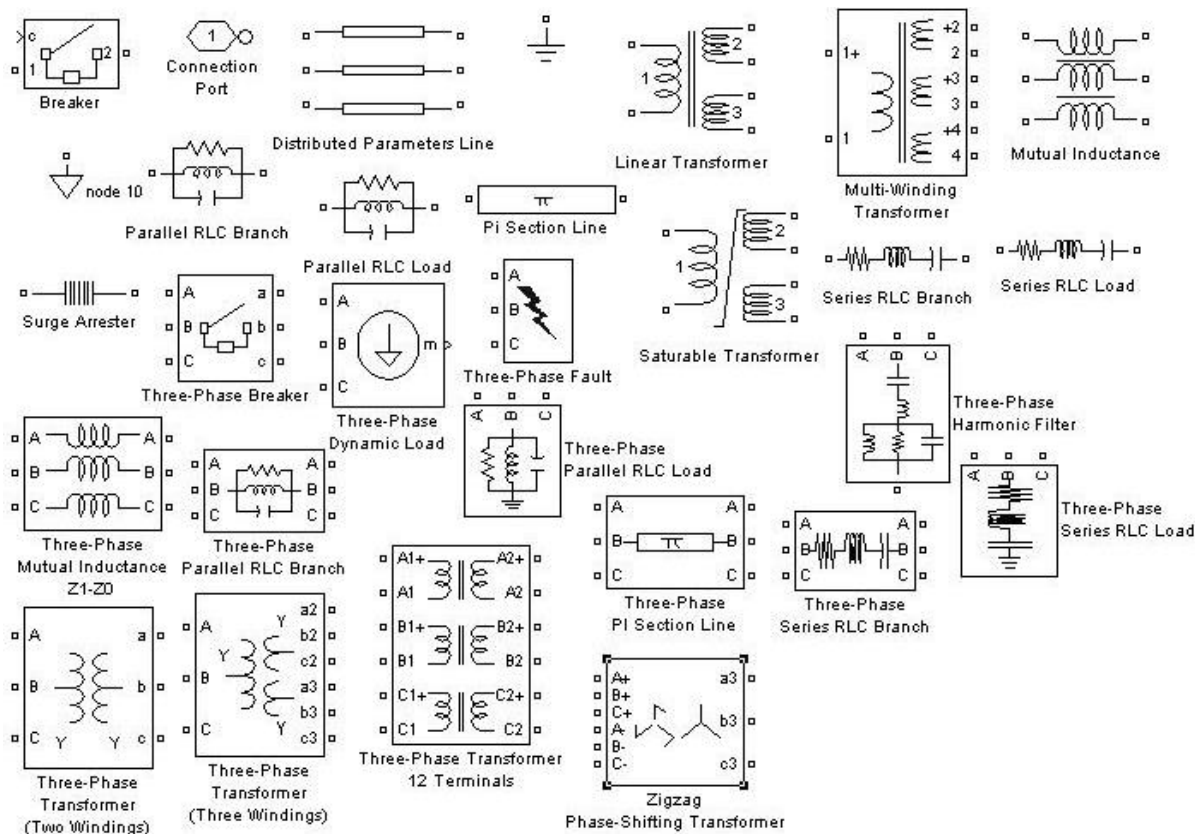


Рис. 6.2. Состав библиотеки Elements

7. Нелинейный элемент (Surge Arrester), позволяющий сформировать требуемую нелинейную зависимость между входным и выходным сигналами.

6.6.3. Устройства силовой электроники

В библиотеке устройств Power Electronics (рис. 6.3) находятся модели полупроводниковых приборов (диод, тиристор, запираемый тиристор, транзисторы IGBT и MOSFET), а также универсальный мост, позволяющий мо-

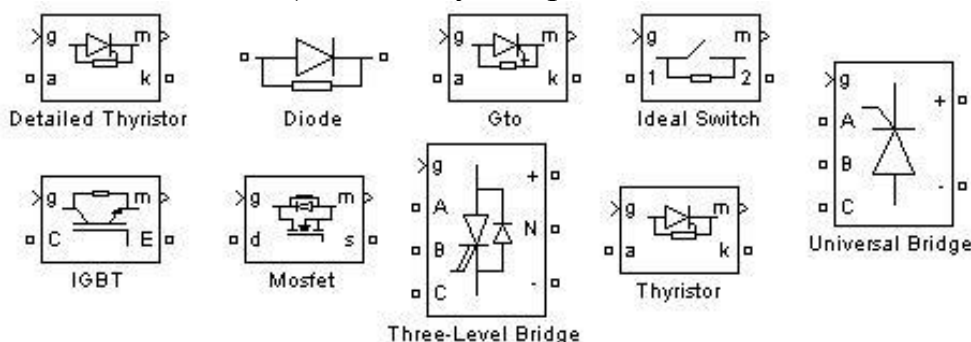


Рис. 6.3. Состав библиотеки Power Electronics

делировать одно-, двух- и трехфазные выпрямительные или инверторные схемы. Кроме того, в библиотеке присутствует модель трехуровневого моста и модель идеального ключа.

В дополнительных библиотеках Extras помещены дополнительные дискретные и непрерывные модули управления.

6.6.4. Измерительные и контрольные устройства

Состав библиотеки Measurements показан на рис. 6.4. В библиотеке измерительных устройств находятся модули измерения мгновенных значений тока и напряжения, модуль измерения токов и напряжений в трехфазной системе, измеритель полного сопротивления участка электрической схемы, а также модуль Multimeter, позволяющий измерять переменные модулей без непосредственного подключения измерительных устройств.

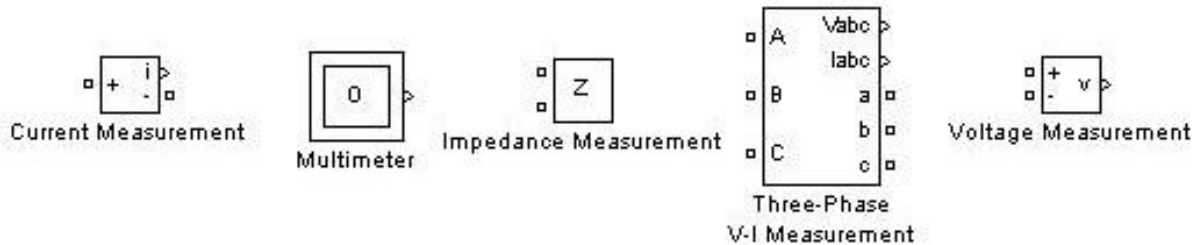


Рис. 6.4. Состав библиотеки Measurements

Выходными сигналами измерительных модулей являются сигналы Simulink, которые можно подать на модули отображения (Scope, Display) или на вход любого модуля.

6.6.5. Расширенные библиотеки

Среди дополнительных библиотек Extra Library (рис. 6.5) находится большое количество разнообразных модулей.

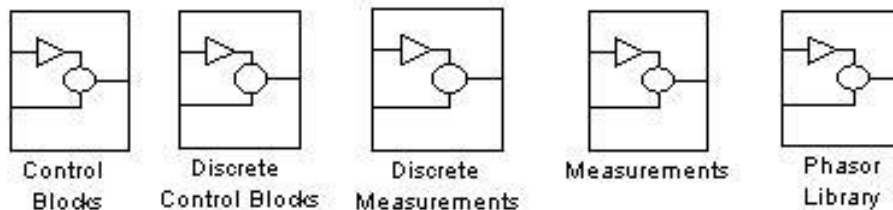


Рис. 6.5. Состав дополнительных библиотек Extra Library

В частности, библиотека дополнительных измерительных приборов, библиотека устройств управления преобразователями и т. д.

6.6.6. Трансформаторы

В состав библиотеки Sim Power System входят следующие модели трансформаторов (рис. 6.6):

- линейный однофазный двух- или трехобмоточный трансформатор (Linear Transformer);
- многообмоточный однофазный трансформатор (Multi-Winding Transformer);

- двухобмоточный трехфазный трансформатор (Three-Phase Transformer (Two Windings));
- трехобмоточный трехфазный трансформатор (Three-Phase Transformer (Three Windings));
- двухобмоточный трехфазный трансформатор с выведенными наружу зажимами всех обмоток (Three-Phase Transformer 12 Terminals);
- трехобмоточный трехфазный трансформатор (Zigzag Phase-Shifting Transformer). У него первичная и одна из вторичных обмоток соединены в зигзаг, соединение другой вторичной обмотки выбирается из выпадающего списка.

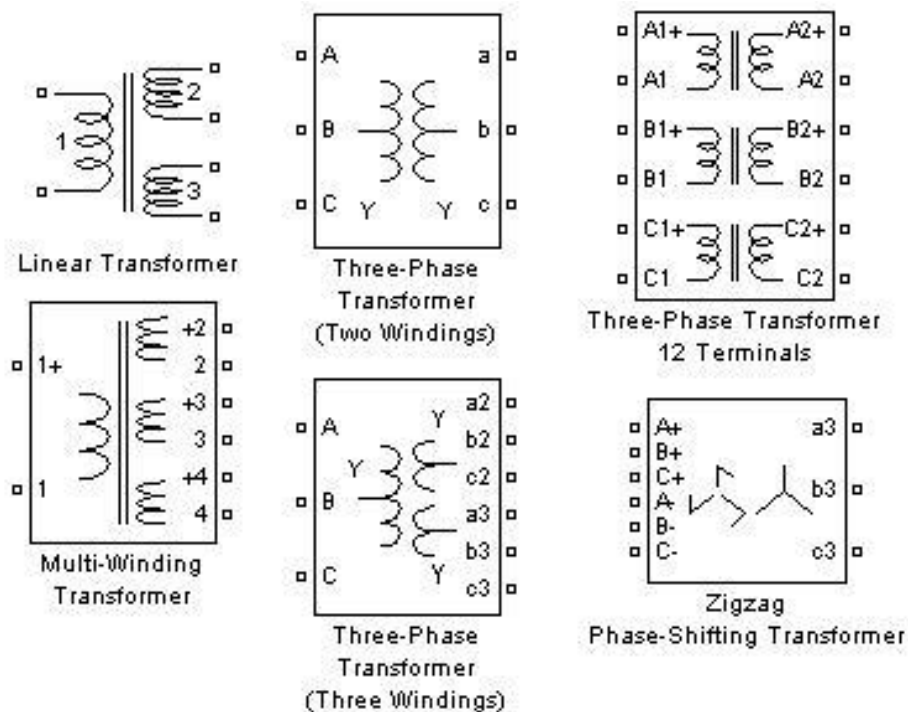


Рис. 6.6. Состав библиотеки Sim Power System

Для всех трансформаторов параметры схемы замещения задаются в относительных величинах. Эти параметры рассчитываются исходя из паспортных данных на трансформатор.

6.7. Оптимизация параметров ИУС

Процедура постановки задачи оптимизации параметров ИУС носит неформальный характер и включает следующие этапы: выбор критериев оптимальности, формирование целевой функции, выбор управляемых параметров, назначение ограничений, нормирование управляемых и выходных параметров.

Под оптимизацией понимается процесс поиска наилучшего варианта проектируемой ИУС в условиях множества альтернатив. При проектировании ИУС необходимо найти их структуру и параметры, обеспечивающие

наилучшее сочетание показателей качества и эффективности. При этом возникает проблема формализации понятия «наилучший». Для выбора наилучшего варианта среди определенного множества необходимо сформулировать некоторое правило предпочтения. Основой такого правила может быть однозначная численная характеристика ИУС, представляющая собой скалярную функцию. Эта характеристика содержательно отображает цель поиска, в связи с чем ее называют целевой функцией. Она позволяет количественно выразить качество объекта и поэтому называется также функцией качества. Таким образом, в основе построения правила предпочтения лежит целевая функция.

В качестве критериев оптимальности принимаются выходные параметры объекта проектирования. При этом часто выбирают не все выходные параметры модели, а только те, которые оказывают наибольшее влияние на достижение конечной цели проектирования всей ИУС. Остальные выходные параметры используются при формировании функций ограничений.

На основе выбранных критериев осуществляется формирование целевой функции.

Основная проблема постановки задачи оптимизации заключается в выборе критериев и формировании целевой функции. Выбор критериев оптимальности требует глубокого понимания сущности решаемой задачи. Всесторонняя оценка эффективности и качества объекта возможна при использовании множества критериев. Задача оптимизации в этом случае становится многокритериальной.

Векторный характер критериев оптимальности создает проблему формирования целевой функции. Сложность ее обусловлена не только большим количеством используемых критериев, но и их характером. Обычно улучшение одного из критериев приводит к ухудшению других критериев. Такие критерии называются конфликтными. Если определить оптимальные параметры объекта по каждому из критериев в отдельности, то они окажутся различными. Поэтому при наличии векторного критерия возможно лишь некоторое компромиссное решение, которое в наибольшей мере отвечает задачам проектирования.

В зависимости от вида математической модели ИУС критерии могут быть детерминированные или статистические.

Если оптимизация параметров технического объекта осуществляется на основе детерминированной математической модели, то используемые

критерии оптимальности (выходные параметры объекта) будут также детерминированными. Если же для оптимизации параметров используется вероятностная математическая модель объекта, которая учитывает случайные воздействия внешней среды и (или) случайный разброс параметров, то значения критериев представляют собой статистические оценки выходных параметров объекта, т. е. критерии оказываются статистическими. Значения критериев на каждом шаге поиска при этом представляют собой случайные числа, так как они обусловлены не только изменением управляемых параметров, но и случайными факторами. Поиск решения в этом случае обеспечивается вероятностными итеративными алгоритмами. Эти алгоритмы можно разделить на три группы: использующие детерминированный поиск; использующие случайный поиск; комбинированного типа, использующие детерминированный и случайный поиск.

6.8. Методика оптимизации параметров ИУС

Задача параметрической оптимизации ИУС заключается в поиске параметров, при которых целевая функция достигает экстремального значения.

В достаточно общем виде математическую задачу оптимизации можно сформулировать следующим образом: минимизировать (максимизировать) целевую функцию с учетом ограничений на управляемые переменные.

При записи математических задач оптимизации в общем виде обычно используется следующая символика:

$$Q(\alpha) \xrightarrow{\alpha \in U} \min(\max)$$

где $Q(\alpha)$ – целевая функция; U – допустимое множество, заданное ограничениями на управляемые переменные.

Часто для большей эффективности работы алгоритмов оптимизации в целевую функцию (критерий качества) вводят штрафную или барьерную функцию. Штрафная функция равна 0, если $\alpha \in U$, и быстро возрастает при удалении точки $\alpha \in U$ от допустимого множества.

Разработку алгоритмов управления ИУС выполняют, как правило, с учетом двух важнейших оценок качества: быстродействия (с учетом ограничений на потребляемую мощность) и связанной с ним производительности; интегральной квадратичной оценки ошибок управления и связанным с ней качеством оказания услуг.

Методы формального синтеза алгоритмов управления ИУС и сложности, возникающие при этом, достаточно хорошо освещены в литературе по теории инфокоммуникационных систем. Синтез оптимальных алгоритмов управления локальными и взаимосвязанными системами может производиться в соответствии с оптимизирующими функционалами вида [31].

Особенностями задачи оптимизации параметров ИУС являются следующие: 1) исходные модели ИУС представляются в виде сложных структурных схем со значительным числом блоков, операторы которых принадлежат разным классам; 2) оптимизация по различным параметрам (время, надежность передачи данных и др.) предполагает гибкое формирование различных функционалов качества; 3) оптимизируемые переходные процессы в ряде случаев имеют специфический, нетиповой вид, автоматическая оценка которых общепринятыми показателями качества (время регулирования, перерегулирование и т. д.) представляет собой некоторую проблему при оптимизации по прямым показателям качества; 4) использование типовых регуляторов предполагает относительно невысокую размерность пространства оптимизируемых параметров в пределах одной подсистемы ИУС.

В случае когда при проектировании ИУС производится декомпозиция (горизонтальная или вертикальная), необходимо говорить о локальной оптимизации каждого элемента ИУС. Сущность локальной оптимизации заключается в том, что на каждом уровне декомпозиции применяются свои критерии оптимальности и осуществляется оптимизация лишь некоторой части параметров ИУС, относящихся к внутренним параметрам проектируемого элемента ИУС. В результате оказывается, что при проектировании любого элемента ИУС используемые критерии получены как результат оптимизации параметров более крупного элемента ИУС. По существу декомпозиция объекта приводит к декомпозиции критериев. Следовательно, критерии на всех уровнях декомпозиции объекта взаимосвязаны и подчинены конечной цели – достижению высоких показателей эффективности и качества функционирования технического объекта.

Проектирование элементов ИУС, выделяемых при декомпозиции ИУС, основано на моделировании некоторого характерного режима их функционирования. Это может быть переходный процесс, статическое состояние (состояние покоя или равномерной передачи данных), стационарный случайный процесс и др. Методы решения систем уравнений этих моделей и способы оценки выходных параметров объектов проектирования

существенно различны. Проектные задачи различаются также видами зависимостей критериев от оптимизируемых параметров, количеством используемых критериев и способом формирования целевой функции.

Многообразие исследовательских и проектных задач привело к разработке множества методов оптимизации, обладающих различными свойствами и возможностями поиска экстремума целевой функции с учетом особенностей математических моделей объектов. Проектировщик должен хорошо знать особенности решаемой задачи и свойства существующих методов оптимизации, предвидеть характер изменения целевой функции, что позволит осуществить обоснованный выбор метода и повысить вероятность решения задачи с минимальными затратами.

6.9. Программные средства для оптимизации

Для оптимизации требуется применение таких программных средств, которые обеспечивают достижение требуемого результата за минимальное время.

В настоящее время существует множество работ, в которых подробно изложены различные методы оптимизации, поэтому не имеет смысла излагать подробно их суть. Приведем только наиболее важные методы и их иерархию.

Существуют следующие методы оптимизации [24]:

- *Одномерная оптимизация*, включающая: а) прямые методы (перебора, поразрядного поиска, исключения отрезков, парабол); б) методы, использующие производные функции (средней точки, хорд, Ньютона, кубической аппроксимации); в) минимизации многомодальных функций (перебора, ломаных).

- *Безусловная минимизация функций многих переменных*: а) прямые методы (минимизация по правильному симплексу, по деформированному симплексу, циклического покоординатного спуска, Хука–Дживса, случайного поиска, сопряженных направлений); б) использующие производные функции (градиентного спуска, наискорейшего спуска, сопряженных градиентов, Ньютона, квазиньютоновские).

- *Многомерная минимизация при наличии ограничений*: а) линейное программирование (графический, симплекс-таблица, симплекс-метод, искусственного базиса, симплекс-метод при отсутствии условий неотрицательности переменных); б) нелинейное программирование (возможных направлений (случай линейных ограничений, случай нелинейных ограничений),

градиентные (проекции градиента, условного градиента), последовательной безусловной минимизации (штрафных функций, барьерных функций)).

- *Решение дискретных задач оптимизации*: отсечений, ветвей и границ, динамического программирования.

- *Оптимизация в бесконечномерных пространствах*: а) приближенные методы безусловной минимизации функционалов в гильбертовом пространстве (градиентного спуска, наискорейшего спуска, сопряженных градиентов); б) приближенные методы минимизации функционалов при наличии ограничений (проекции градиента, штрафных функционалов, условного градиента).

Выбор алгоритма оптимизации зависит от таких факторов, как точность поиска экстремума, быстродействие, условия функционирования системы, и определяется решением конкретной задачи. К этой группе методов можно отнести методы безусловной минимизации функций многих переменных и многомерной минимизации при наличии ограничений (нелинейное программирование).

Для решения поставленной задачи возможно применение двух toolbox (приложений) SIMULINK и OPTIMIZATION пакета MATLAB.

В Optimization Toolbox используется три метода для решения нелинейных задач без наличия ограничений: *квази-Ньютоновский метод*; *метод Нелдера Мида*; *метод доверительных областей*.

Задачи нелинейной оптимизации при наличии ограничений представляют собой комбинацию неких нелинейных целевых функций и могут иметь линейные или нелинейные ограничения. В Optimization Toolbox используется два основных метода для решения подобных задач: *доверительных областей*; *последовательное применение активного набора задач квадратичного программирования*.

Многокритериальная оптимизация предназначена в основном для минимизации многоцелевых функций с учетом некоего набора ограничений. В Optimization Toolbox реализованы два типа постановки задачи многокритериальной оптимизации: *задача достижения цели* и *задача минимакса*.

Optimization Toolbox может решать нелинейные задачи методом наименьших квадратов. Для этого в Optimization Toolbox используется три метода: *доверительных областей*; *Левенберга–Марквардта*; *Ньютона–Гаусса*.

В задачах линейного программирования целевая функция представляет собой некое линейное выражение, на которое могут быть наложены ограничения в виде линейных равенств или неравенств. Для решения дан-

ного типа задач используются следующие два метода: симплексный; внутренней точки.

Кроме известных и традиционных методов оптимизации могут применяться и современные методы, основанные на нейронных сетях [32] и нечеткой логики.

6.9.1. MATLAB Optimization Toolbox

Основные типы задач оптимизации, решаемых системой MATLAB Optimization Toolbox, представлены в табл. 6.4.

В табл. 6.4 приняты следующие обозначения: a – скалярный аргумент; x, γ – в общем случае векторные аргументы; $f(a), f(x)$ – скалярные функции; $F(x), c(x), seq(x), K(x, w)$ – векторные функции; A, Aeq, C, H – матрицы; $b, beq, d, f, w, goal$ – векторы; x_L, x_U – соответственно, нижняя и верхняя границы области изменения аргумента.

Но этапу поиска экстремума функционала качества, как правило, предшествует подготовительная работа, связанная с созданием удобного пользовательского интерфейса. Это необходимо делать, так как напрямую методы оптимизации, приведенные в табл. 6.4, использовать сложно. В [33] приводятся примеры такого подхода к оптимизации.

Таблица 6.4

Тип задачи	Математическая запись	Используемая функция MATLAB
Скалярная (одномерная) минимизация	$\min f(a), a_1 < a < a_2$ a	fminbnd
Безусловная минимизация (без ограничений)	$\min f(x)$ x	fminunc, fminsearch
Линейное программирование	$\min f^T x$ при условиях x $A \cdot x \leq b, Aeq \cdot x = beq, x_L \leq x \leq x_U$	linprog
Квадратичное программирование	$\min 0.5x^T H x + f^T x$ при условиях x $A \cdot x \leq b, Aeq \cdot x = beq, x_L \leq x \leq x_U$	quadprog
Минимизация при наличии ограничений	$\min f(x)$ при условиях $c(x) \leq 0, seq(x) = 0, A \cdot x \leq b, Aeq \cdot x =$ $=$ $= beq, x_L \leq x \leq x_U$	fmincon
Достижение цели	$\min \gamma$ при условиях x, e $F(x) - w\gamma \leq goal, c(x) \leq 0,$	fgoalattain

	$ceq(x) = 0, A \cdot x \leq b,$ $Aeq \cdot x = beq, x_L \leq x \leq x_U$	
Минимакс	$\min \max \{F_i(x)\}$ при условиях $x \{F_i\}$ $c(x) \leq 0, ceq(x) = 0,$ $A \cdot x \leq b, Aeq \cdot x = beq, x_L \leq x \leq x_U$	fminimax
Полубесконечная минимизация	$\min f(x)$ при условиях x $K(x, w) \leq 0$ для всех $w,$ $c(x) \leq 0, ceq(x) = 0, A \cdot x \leq b,$ $Aeq \cdot x = beq, x_L \leq x \leq x_U$	fsemif

7. КОМПЬЮТЕРНЫЕ МЕТОДЫ ИССЛЕДОВАНИЯ С ВИЗУАЛИЗАЦИЕЙ ДИНАМИЧЕСКИХ ПРОЦЕССОВ В ИУС

7.1. Средства визуального моделирования динамических систем

В настоящее время существует множество визуальных средств моделирования. Рассмотрим универсальные пакеты, позволяющие моделировать структурно-сложные гибридные (цифровые) системы, в том числе и ИУС. Их можно разделить условно на три группы:

- пакеты «компонентного моделирования»;
- пакеты «физического моделирования»;
- пакеты, ориентированные на схему гибридного автомата.

Пакеты «компонентного моделирования» ориентированы на графический язык иерархических блок-схем. Элементарные блоки либо являются predetermined, либо конструируются с помощью специального языка более низкого уровня. Новый блок можно собрать из имеющихся блоков с использованием ориентированных связей и параметрической настройки. В число predetermined блоков входят чисто непрерывные, чисто дискретные и гибридные блоки. К достоинствам этого подхода следует отнести простоту создания не очень сложных моделей, а также эффективность реализации элементарных блоков и простоту построения эквивалентной системы. Другими словами, этот подход работает хорошо, когда есть подходящие стандартные блоки. Наиболее известны следующие пакеты «компонентного моделирования»: пакет *Matuh* (подсистема *Simutink*); пакет *Matrix_x* (подсистема *SystemBuild*); пакет *VmSim*.

Пакеты «физического моделирования» позволяют использовать неориентированные и потоковые связи. Пользователь может сам определять новые классы блоков. Непрерывная составляющая поведения элементарного блока задается системой алгебродифференциальных уравнений и формул. Дискретная составляющая задается описанием дискретных событий, при возникновении которых происходят мгновенные присваивания пере-

менным новых значений. Дискретные события могут распространяться по специальным связям. Изменение структуры уравнений возможно только косвенно через коэффициенты в правых частях, что обусловлено необходимостью символьных преобразований при переходе к эквивалентной системе. Подход очень удобен для описания типовых блоков физических систем. К пакетам «физического моделирования» следует отнести: Dymola, Otola, OmSim.

Пакеты, основанные на использовании *схемы гибридного автомата*, позволяют наглядно и естественно описывать гибридные системы со сложной логикой переключений. Необходимость определения эквивалентной системы при каждом переключении заставляет использовать только ориентированные связи. Пользователь может сам определять новые классы блоков. Непрерывная составляющая поведения элементарного блока задается системой алгебродифференциальных уравнений и формул. К этому направлению относятся пакеты: Shift (ориентирован на описание сложных динамических структур); Model Vision Studium (ориентирован на описание сложных поведений).

7.2. UML как средство проектирования моделей динамических систем

Унифицированный язык моделирования (UML – Unified Modeling Language) представляет собой универсальный язык, позволяющий одновременно с анализом создавать документацию для проектирования моделей сложных иерархических систем. UML относится к языкам *визуального моделирования*, предназначен для общения разработчиков при работе над одним проектом и для однообразного описания различных проектов. UML объектно-ориентирован, но в то же время никак не связан с конкретными объектно-ориентированными языками программирования. Разработанный в терминах UML проект можно легко воплотить на любом существующем языке, поддерживающем объектно-ориентированную технологию.

Основные компоненты UML включают описание семантики UML, его графической нотации и дополнительных понятий. Документация по UML содержит подробное описание этих компонентов и в виде PDF-файлов представлена на сайте Rational Software (www.rational.com). Описание языка, представленное в данной работе, является весьма сокращённым и включает лишь те конструкции, которые образуют методы и средства проектирования технических систем.

В процессе разработки система представляется в виде объединения нескольких проекций, каждая из которых описывает определенный аспект разрабатываемой системы, а вместе они определяют систему в целом. Эти проекции представляются в UML следующими диаграммами:

– *диаграммы вариантов использования (use case diagrams)*. Обычно у сложных систем много функций и существует много вариантов их исполь-

зования. Список всех групповых сценариев определяет функциональные требования к системе, с помощью которых может быть сформулировано техническое задание. Диаграммы вариантов использования представляют собой граф, с помощью которого показаны все типичные действующие лица и их взаимодействие с системой. Взаимодействие представлено сценариями применения;

– *диаграммы классов (class diagrams)*. Диаграммы классов показывают статическую структуру системы, т. е. определяют тип объектов системы и различного рода статические связи и отношения между ними. Диаграммы классов содержат набор статических (декларативных) элементов, таких как классы, типы и их связи, изображённых в виде графов. Диаграммы классов могут быть логически объединены в пакеты;

– *диаграммы взаимодействия (interactio diagrams)* подразделяются на диаграммы последовательности и кооперативные диаграммы. На обеих диаграммах представлена временная последовательность использования объектов при реализации конкретного сценария и сообщений, которыми они при этом обмениваются;

– *диаграммы последовательности (sequence diagrams)* показывают, в какой последовательности появляются объекты при выполнении той или иной операции (сценария) и какой поток сообщений при этом возникает. Диаграммы последовательности имеют две оси: вертикальная представляет время, горизонтальная – различные объекты. Обычно интерес представляет только последовательность появления объектов по мере выполнения операции, но в случае систем реального времени могут понадобиться и конкретные значения времени;

– *диаграммы состояний (state diagrams)*. Любой объект системы может менять своё поведение в зависимости от происходящих внутренних или внешних событий, или, иными словами, он может реагировать на события, меняя своё состояние. Диаграммы состояний показывают последовательность состояний, в которых может оказаться объект, в зависимости от происходящих событий, вместе с их реакциями на эти события. Диаграммы состояний описывают состояния только одного класса или объекта;

– *диаграммы деятельности (activity diagrams)*. Во многих случаях можно наблюдать смену состояний объектов, вызываемую только внутренними причинами. Внутри объекта выполняются последовательно или параллельно предопределённые длительные действия. Для такого типа систем UML и предлагает использовать диаграммы деятельности. Такая диаграмма предназначена для того, чтобы выразить переходы, вызванные внутренними процессами (в противоположность внешним событиям, с которыми имеет дело диаграмма состояний). В большей степени удовлетворяет этим целям формализм под названием «гибридный автомат».

7.3. Классы моделей и технология их реализации в пакетах визуального моделирования

Библиотеку моделей, используемых в современных пакетах моделирования, образуют следующие классы моделей.

Пространство. В законах Ньютона о движении наряду с *независимой переменной* «время» присутствуют пространственные координаты (x, y, z) движущейся точки. Четвёрка (t, x, y, z) определяет положение тела в пространстве и времени. Графики $x(t), y(t), z(t)$ называются временными диаграммами. Если следить за положением точки $\{x(t), y(t), z(t)\}$ в осях x, y, z , то можно говорить о фазовых диаграммах.

События. При любой модели времени поведение системы с фазовым вектором $x(t)$ в любой момент времени t^* характеризуется парой $\{t^*, x(t^*)\}$, которую принято называть событием. У события две координаты – временная t^* и пространственная $x(t^*)$. Первая указывает на время наступления события, а вторая характеризует явление. В случае если пространственные координаты и время непрерывны, обычно говорят не обо всём множестве событий, а лишь об отдельных событиях, называя их особыми. Если функция является кусочно-постоянной, то событием обычно называют вторую координату, т. е. явление. Это обусловлено тем, что в технических устройствах идеальный скачок (разрыв первого рода) реализовать невозможно. Использование только второй координаты события для кусочно-постоянных функций определило термин «дискретные события» и ещё один способ описания таких функций. В языках моделирования имеются специальные операторы, описывающие события и связанные с их наступлением действия. Если фиксировать только определённое явление, то можно выделить множество всех одинаковых явлений, случившихся в произвольные моменты времени. Соответственно, описать события можно следующим образом: задав обе координаты пары, только первую или только вторую координату.

Классическая динамическая система. Классическую динамическую систему можно задавать либо в виде систем дифференциальных уравнений, либо в виде явных формул решений этих уравнений.

Системы с непрерывным временем. Различают следующие формы описания непрерывного поведения:

- форма вида $z = z(t)$;
- нелинейные алгебраические уравнения относительно переменных, явно зависящих от времени $G(z(t), t) = 0$;
- дифференциальные уравнения в форме систем уравнений первого порядка $\dot{z} = F(z, t)$ с заданными начальными условиями; дифференциальные уравнения в форме систем линейных алгебраических уравнений относительно первых производных вида $A\dot{z} = F(z, t)$;
- дифференциальные уравнения в общей форме $F(\dot{z}, z(t), t) = 0$;

– алгебродифференциальные уравнения вида $\dot{z} = F(z, v, t)$; $0 = G(z, v, t)$ с согласованными начальными условиями.

Таким образом, можно говорить, что имеем описание обобщённой динамической системы, если её поведение задано любой совокупностью из вышеперечисленных форм.

Передаточные функции. В некоторых пакетах моделирования для описания поведения динамических систем используется язык передаточных функций. В этих случаях исследуемую систему типа «вход–состояние–выход» можно описать с помощью дифференциального оператора F , связывающего «выход» y с «входом» x : $y = Fx$. Полагая, что оператор F линейный и не зависит от времени, x и y – скаляры, приходим к линейным стационарным системам. Для таких систем можно использовать описание, основанное на передаточной функции $W(p) = Y(p) / X(p)$, где соответствующие

интегралы $Y(p) = \int_0^t y(t) e^{-pt} dt$; $X(p) = \int_0^t x(t) e^{-pt} dt$ вычислены при нулевых начальных условиях. Передаточными функциями конечномерных линейных стационарных объектов являются дробно-рациональные функции комплексной переменной p :

$$W(p) = b_m p^m + b_{m-1} p^{m-1} + \dots + b_1/p + a_n p^{n-1} + \dots + a_1.$$

По передаточной функции можно легко восстановить вид дифференциального уравнения, описывающего поведение системы.

Конечные автоматы. Опираясь на событийное время, конечный автомат можно рассматривать как динамическую систему. Находясь в некотором состоянии, он воспринимает очередной символ из последовательности входных сигналов, мгновенно реагирует на него, формируя последовательность выходных символов, и меняет своё состояние. О таких системах говорят, как о системах с «памятью», потому что, как и в дифференциальном уравнении, в конечном автомате текущее состояние зависит от предыстории и от значения входного сигнала.

Гибридные системы (автоматы). При построении модели гибридной системы используемый для этих целей формализм, очевидно, должен содержать средства для описания непрерывного поведения и механизм смены состояний. Тогда, объединяя язык дифференциальных уравнений и язык конечных автоматов в единый формализм, можно описать смену поведений в различных областях фазового пространства. В современных пакетах могут быть реализованы следующие типы гибридных систем:

- системы, сохраняющие единое описание во всём пространстве и меняющие только начальные условия при переходе из области в область;
- системы, меняющие значения параметров в правых частях уравнений в зависимости от наступления событий;
- системы, меняющие правые части и число уравнений в момент, когда происходит смена состояния.

Гибридные системы (автоматы) и карты состояний. Это в основном относится к динамическим системам, меняющим своё поведение во времени. Одной из черт сложного поведения является наличие у системы нескольких качественно различных, последовательно сменяющих друг друга во времени поведений. Такой тип сложного поведения можно реализовать, если описать всю совокупность допустимых, простых, в некотором смысле, частных поведений и указать правила переключения с одного поведения на другое. Организованная таким образом сложная динамическая система в каждый конкретный момент времени ведёт себя, как некоторая простая динамическая система. Каждое конкретное поведение можно отождествить со значением некоторой дискретной переменной, а мгновенные переключения текущего поведения – с дискретными событиями. Для передачи информации о переходе одних дискретных состояний в другие используют специальные переменные – сигналы. Набор дискретных состояний вместе с условиями переходов из одного состояния в другое образует обычное дискретное поведение.

В моменты переходов могут происходить мгновенные скачкообразные изменения значений переменных. Поскольку в каждом из дискретных состояний элементарный блок ведёт себя как некоторая непрерывная система, то поведение блока в целом является непрерывно-дискретным или гибридным.

В настоящее время стандартом описания дискретных «машин состояний» стала карта состояний (state chart), предложенная Д. Харелом и «узаконенная» в стандарте UML. В качестве примера рассмотрим динамическую систему $x' = f(x)$, $x \in R$, $R = R_1 \cup R_2$ с правой частью, представимой в виде $f(x) = k_1 f_1(x) + k_2 f_2(x)$.

Так как коэффициенты $k_1(x)$ и $k_2(x)$ – кусочно-непрерывные функции, то их можно представить с помощью конечного автомата. У этого автомата два состояния: R_1 и R_2 , выходные символы $\{k_1 = 1, k_2 = 0\}$ для состояния R_1 и $\{k_1 = 0, k_2 = 1\}$ для состояния R_2 , а входными сигналами являются два сигнала, сигнализирующие о достижении границы областей R_1 и R_2 . Состояние R_1 помечено как начальное. Автомат анализирует поступающие на него значения переменной x входного сигнала и меняет своё текущее состояние, только если выполняются условия, предписанные дугам. Если каждому состоянию поставить в соответствие различные значения некоторой переменной, то они будут меняться скачком, в момент смены состояния. Такие автоматы называются *картами состояний* и предназначены для дискретного изменения коэффициентов. В совокупности с приведенным дифференциальным уравнением карта состояний дает полное описание непрерывно-дискретной, или гибридной, системы.

Поведение гибридного автомата характеризуется последовательностью пар $B = \{([t_0, t_1], St_0), \dots, ([t_i, t_{i+1}], St_i), \dots\}$, в которых фиксируются текущие состояния $St_i \in Q$ и $[t_i, t_{i+1}]$ – времена пребывания в них. Из определения карты поведения следует, что это происходит, если предикат, определяющий условия выхода из узла, оказывается истинным уже для начальных значений переменных. Такая ситуация может повторяться неоднократно. При этом в точке $t_i = t_{i+1}$ непрерывное время «останавливается» и каждый новый текущий узел «присваивает» фазовым переменным новые значения, не продвигая вперёд время. Точка на оси времени, соответствующая неоднозначности фазовых переменных, называется «временной щелью». Последовательность только временных интервалов, в которой возможно равенство $t_i = t_{i+1}$ для некоторых интервалов, называют *гибридным временем*. Записав последовательность в виде $T = \{\tau_0, \tau_1, \tau_2, \dots, \tau_i, \dots\}$, $\tau_i = t_{i+1} - t_i$, $i = 0, 1, 2, \dots$, выделим в ней три типа интервалов (с конечной, нулевой и бесконечной длительностью):

$$\begin{aligned} \tau_i &= t_{i+1} - t_i, t_i \neq t_{i+1}, t_{i+1} \neq \infty; \\ \tau_i &= 0, t_i = t_{i+1}; \\ \tau_i &= \infty, t_{i+1} = \infty. \end{aligned}$$

Данным интервалам соответствуют специфические типы гибридных систем в зависимости от их поведения: нигде не останавливающаяся система, система с притягивающим центром, система с бесконечным временным интервалом.

Гибридные, или событийно-управляемые, системы являются обобщением классических динамических систем, у которых в различных областях фазового пространства меняются уравнения, размерность и состав фазового вектора. При их описании приходится использовать как непрерывные модели (системы дифференциальных уравнений), так и дискретные (конечные автоматы), опирающиеся на различные модели времени. Попытка совместить оба подхода, с тем чтобы было удобно одновременно представлять и непрерывное, и дискретное поведение, приводит к необходимости введения новой модели времени.

7.4. Технология моделирования в инструментальной среде Model Vision Stadium

Model Vision Stadium (MVS) – это интегрированная графическая оболочка для быстрого создания интерактивных визуальных моделей сложных динамических систем и проведения вычислительных экспериментов с ними. К основным свойствам пакета можно отнести:

- поддержку технологии объектно-ориентированного моделирования (ООМ);

- удобное и адекватное описание гибридных (непрерывно-дискретных, цифровых) систем;
- обеспечение достоверности численного решения;
- обеспечение моделирования и визуализации без написания какого-либо программного кода.

Основным компонентом описания в MVS является блок, называемый устройством. Устройство является ориентированным блоком, т. е. все взаимодействия устройства с окружающим миром осуществляются только через его входы и выходы, составляющие интерфейс устройства. Все остальные устройства инкапсулированы внутри него.

В общем случае в описании устройства содержатся следующие элементы: входы, выходы, параметры конструкции, переменные состояния, поведение, локальная структура. Входы, выходы и переменные состояния являются фазовыми переменными, а вместе они образуют фазовый вектор устройства. Описание устройства всегда строится как описание класса устройств, т. е. все устройства являются потомками базового класса CDevice.

Предполагается, что поведение любого блока может быть гибридным. Для задания гибридного поведения авторами пакета выбран формализм гибридного автомата. Для визуального представления смены поведений дискретных объектов и их формального описания используется карта состояний. Как специальное ограничение гибридной карты состояний используется *карта поведений*. Карта поведений (B-chart) – это ориентированный граф, в котором узлам приписываются некоторые локальные поведения, а дугам, называемым переходами, – условия перехода от одного поведения к другому и выполняемые при этом действия. Узел, в котором система находится в каждый конкретный момент времени, называется текущим. Один из узлов должен быть помечен как начальный, он автоматически становится текущим при создании карты состояний. Смена текущего узла происходит в результате срабатывания переходов. Когда узел становится текущим, создаётся экземпляр приписанного ему локального поведения. Созданный экземпляр уничтожается, как только узел перестанет быть текущим. Именно в этом состоит отличие карты поведений от карты состояний. В карте состояний локальное поведение существует всегда, и можно вернуться в его текущее состояние.

Вычислительный эксперимент с визуальной моделью представляет собой совокупность прогонов модели от начального состояния к конечному. Прогон модели включает в себя точки останова, в которых модельное время не изменяется, и интервалы выполнения, в течение которых имитируется функционирование модели в модельном времени. Пользователь наблюдает за ходом прогона модели и интерактивно вмешивается в него, но уже в физическом времени.

7.5. Применение среды LabVIEW для исследования систем управления ИУС

National Instruments LabVIEW представляет собой высокоэффективную среду графического программирования, в которой можно создавать гибкие и масштабируемые приложения измерений, управления и тестирования с минимальными временными и денежными затратами. LabVIEW сочетает в себе гибкость традиционного языка программирования с интерактивной технологией *Экспресс ВП*, которая включает в себя автоматическое создание кода, использование помощников при конфигурировании измерений, шаблоны приложений и настраиваемые *Экспресс ВП*. Благодаря этим особенностям и новички, и эксперты могут легко и быстро создавать приложения в LabVIEW.

LabVIEW содержит следующие инструменты для задач автоматизации и обработки данных [25], [26], [27]:

- *Интерактивные измерения.* С помощью пакета NI SignalExpress можно интерактивно использовать виртуальные приборы (встраиваемые либо автономные приборы, управляемые с компьютера) для проведения необходимых измерений и анализа сигналов.

- *Автоматизированные системы испытаний.* С помощью пакета NI TestStand можно разработать структурированную последовательность испытаний, представляющих собой отдельные программы LabVIEW (либо модули других систем программирования), со сложной логикой принятия решений «тест прошел/не прошел» для управления общим ходом испытания. Кроме того, NI TestStand легко интегрируется в единые информационные системы предприятия (ERP) для предоставления результатов в базы данных либо для отслеживания испытуемых изделий через автоматизированные системы управления производством (MES).

Для проведения автоматизированных измерений LabVIEW содержит пакеты анализа, оптимизированные для различных измерительных задач:

- 1) тестирование линий связи – средства обработки и генерации сложных модулированных сигналов и усовершенствованные функции для проведения спектральных измерений, расширяющие возможности библиотек, содержащихся в базовом комплекте LabVIEW;

- 2) измерение виброакустических сигналов – модули для исследования динамических акустических сигналов с целью оценки качества звука или проведения структурных испытаний;

- 3) мониторинг состояния машин и механизмов – специализированные алгоритмы порядкового анализа вращающихся частей механизмов (вэйвлет-анализ, совместный частотно-временной анализ);

- 4) обработка изображений – средства для автоматизированного визуального контроля и приложений машинного зрения.

Для создания приложений управления LabVIEW содержит отдельный набор специализированных библиотек, дополняющих графическую платформу методами управления, функциями распределенного мониторинга и управления, АСУТП, а также возможностями управления в реальном времени.

Пакет LabVIEW позволяет реализовать:

- ПИД-регулирование;
- расширенное управление: LabVIEW содержит средства разработки алгоритмов оптимального управления на основе моделей контроллер-объект либо управление на базе принятых/выданных сигналов с учетом усовершенствованной идентификации системы. Кроме того, к LabVIEW поставляется дополнительная библиотека для непрерывного контроля работы динамических систем;
- управление передачей данных в локальных, промышленных и других видов сетей;
- повторное воспроизведение опытных сигналов;
- обширная библиотека анализа и математических функций. LabVIEW содержит сотни математических функций, охватывающих широкий спектр традиционных алгоритмов в областях математического анализа, обработки сигналов, вероятности и статистики, систем управления, представляющих собой основу любого пользовательского алгоритма;
- естественная интеграция с устройствами ввода;
- аппаратные платформы для создания систем реального времени: алгоритмы LabVIEW можно выполнять на платформах реального времени с интегрированным вводом/выводом. С помощью модульных аппаратных средств National Instruments CompactRIO и PXI можно быстро создавать прототипы встраиваемых систем, использующих процессоры, программируемые логические интегральные схемы (ПЛИС) для встроенной логики и широкий спектр оригинальных устройств ввода/вывода.

Для реализации ИУС и их моделей в LabVIEW используются стандартные функции (математические, логические, функции генерации и обработки сигналов и др.) и специальные дополнительные модули, в число которых входят:

- Control Design and Simulation Module – модуль для моделирования и проектирования систем автоматического управления;
- Datalogging and Supervisory Control Module – модуль, расширяющий возможности LabView до полноценной SCADA-системы;
- PID and Fuzzy Logic Toolkit – тулkit, содержащий виртуальные приборы для создания ПИД-регуляторов и регуляторов на основе нечеткой логики;
- Simulation Interface Toolkit – тулkit, дающий возможность создания математических моделей динамических систем;
- SoftMotion Module – модуль для управления движением.

Данные модули включают в себя генераторы сигналов, блоки построения осциллограмм и графиков, блоки интегрирования, задержки, запаздывания, блоки передаточной функции и задания модели в пространстве состояний, блоки ПИД-регуляторов, задатчик интенсивности, ограничитель, функции преобразования модели из пространства состояний в передаточную функцию и обратно, блоки анализа модели во временной и частотной областях, блоки расчета оптимальных регуляторов для модели и прочие разнообразные инструменты.

Применение перечисленных инструментов позволяет строить ИУС и модели динамических систем произвольной сложности. Разработка систем может осуществляться в пространстве состояний, с использованием передаточных функций, и на основе задания нулей и полюсов. Также данные наборы инструментов позволяют производить различного рода анализ полученных результатов моделирования, такие как обработка осциллограмм динамических процессов, производить оптимизацию моделируемой системы и др.

СПИСОК ЛИТЕРАТУРЫ

1. Тарасик, В. П. Математическое моделирование технических систем: учеб. для вузов. – Минск: ДизайнПРО, 2004. – 754 с.
2. Веников, В. А. Теория подобия и моделирования / В. А. Веников, Г. В. – М.: Высшая школа, 1984. – 268 с.
3. Советов, Б. Я. Моделирование систем / Б. Я. Советов, С. А. Яковлев – М.: Высшая школа, 1985. – 343 с.
4. Моделирование систем: учеб. для вузов / Б. Я. Советов, С. А. Яковлев. – М.: Высшая школа, 2005. – 343 с.

5. Белов, М. П. Моделирование компьютеризированных электроприводных систем – СПб.: Технолит, 2008. – 172 с.
6. Советов, Б. Я. Моделирование систем: Практикум: учеб. пособие для вузов / Б. Я. Советов, С. А. Яковлев – М.: Высшая школа, 2005. – 295 с.
7. Черняховская, Л. Р. Моделирование систем: конспект лекции / Л. Р. Черняховская – Уфа: УГАТУ, 2007. – 138 с.
8. Системный анализ и принятие решений: словарь-справочник: учеб. пособие для вузов / Под. ред. В. Н. Волковой, В. Н. Козлова. – М.: Высшая школа, 2004. – 616 с.
9. Антонов, А. В. Системный анализ. учеб. для вузов / А.В. Антонов. – М.: Высшая школа, 2006. – 454 с.
10. Советов, Б. Я. Моделирование систем / Б. Я. Советов, С. А. Яковлев – М.: Высшая школа, 1998. – 343 с.
11. Советов, Б. Я. Информационная технология / Б. Я. Советов – М.: Высшая школа, 1994. – 242 с.
12. Форрестер, Дж. Основы кибернетики предприятия / Дж. Форрестер – М.: Прогресс, 1981. – 363 с.
13. Яковлев, С. А. Эволюционные имитационные модели процессов и систем как методологическая основа интеллектуальных технологий обучения // Междунар. конф. «Современные технологии обучения»: тез. докл. – СПб., 1996. – с. 56 - 62.
14. Яковлев С. А., Шабуневич Е. В. Системное моделирование процессов управления информационными ресурсами в цифровых сетях интегрального обслуживания // Сб. науч. тр. «Вычислительная техника и новые информационные технологии». – Уфа: изд. УГАТУ, 1997. – С. 89 - 97.
15. Кандрашина, Е. Ю. Представление знаний о времени и пространстве в интеллектуальных системах / Е. Ю. Кандрашина, Л. В. Литвинцева, Д. А. Поспелов – М.: Наука, 1989. – 286 с.
16. Хожен, Г. Синергетика. Иерархии неустойчивостей в самоорганизующихся системах и устройствах / Г. Хожен – М.: Мир, 1985. – 244 с.
17. Ермаков, С. М. Математический эксперимент с моделями сложных стохастических систем / С. М. Ермаков, В. Б. Мелос – СПбГУ, 1993. – 328 с.
18. Калашников, В. В. Математические методы построения стохастических моделей обслуживания / В. В. Калашников, С. Т. Рачев – М.: Наука, 1988. – 312 с.
19. Клейнен, Дж. Статистические методы в имитационном моделировании / Дж. Клейнен – М.: Статистика, 1978. – 268 с.
20. Математическая теория планирования эксперимента / Под ред. С. М. Ермакова. – М.: Наука, 1983. – 236 с.
21. Авдеев, О.Н. Моделирование систем: учеб. пособие / О. Н. Авдеев, Л. В. Мотайленко СПб.: Изд-во СПбГТУ, 2001. – 242 с.
22. Белов, М. П. Автоматизированный электропривод типовых производственных механизмов и технологических комплексов: Учеб. для вузов / М.

П. Белов, В. А. Новиков, Л. Н. Рассудов – М.: Изд. центр «Академия», 2004. – 576 с.

23. Инжиниринг электроприводов и систем автоматизации: Учеб. пособие для студентов высш. учеб. заведений / М. П. Белов, О. И. Зементов, А. Е. Козярук и др.: Под ред. В. А. Новикова и Л. М. Чернигова. – М.: Изд. центр «Академия», 2006. – 384 с.

24. Химмельблау, Д. Прикладное нелинейное программирование / Д. Химмельблау – М.: Мир, 1975. – 546 с.

25. Корячко, В. П. Теоретические основы САПР / В. П. Корячко, В. М. Курейчик, И. П. Норенков. – М.: Энергоатомиздат, 1987. – 284 с.

26. Тревис, Дж. LabVIEW для всех / Пер. с англ. Н. А. Клушин. – М.: ДМК Пресс; Прибор Комплект, 2005. – 624 с.

27. Суранов, А. Я. LabVIEW 7: справочник по функциям. – М.: ДМК Пресс, 2005. – 346 с.

28. Методы моделирования стохастических систем управления: Учеб. пособие / В. Ю. Емельянов; Балт. гос. техн. ун-т; СПб., 2004. 168 с.

29. Герман-Галкин, С. Г. Электрические машины: Лабораторные работы на ПК / С. Г. Герман-Галкин, Г. А. Кардонов – СПб.: КОРОНА принт, 2003. – 256 с.

30. Новгородцев, А. Б. Расчет электрических цепей в MATLAB: Учеб. курс. – СПб.: Питер, 2004. – 250 с.

31. Башарин, А. В. Управление электроприводами: Учеб. пособие для вузов / А. В. Башарин, В. А. Новиков, Г. Г. Соколовский. – Л.: Энергоиздат, 1982. – 392 с.

32. Антонов, В. Н. Адаптивное управление в технических системах: Учеб. пособие / В. Н. Антонов, В. А. Терехов, И. Ю. Тюкин. – СПб.: Изд-во СПбГУ, 2001. – 186 с.

33. Дьяконов, В. П. MATLAB 6/6.1/6.5 + Simulink 4/5 в математике и моделировании. Полное руководство пользователя. – М.: СОЛОН-Пресс, 2003. – 576 с.

34. Новгородцев, А. Б. Расчет электрических цепей в MATLAB: Учеб. курс. – СПб.: Питер, 2004. – 250 с.

ПРЕДИСЛОВИЕ 3

1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ МОДЕЛИРОВАНИЯ 7

1.1. Основные понятия теории моделирования 7

1.2. Теория подобия в моделировании систем 9

1.3. Системный подход в моделировании систем 10

1.3.1. Подходы к исследованию систем 13

1.4. Использование моделирования при исследовании

и проектировании сложных систем 18

1.5. Общая характеристика проблемы моделирования систем 20

1.6. Виды моделей 26

1.7. Классификация видов моделирования систем 33

1.8. Основные свойства модели системы	40
1.9. Жизненный цикл моделируемой системы	41
Вопросы для самопроверки	42
2. МАТЕМАТИЧЕСКИЕ СХЕМЫ МОДЕЛИРОВАНИЯ СИСТЕМ	42
2.1. Математические схемы описания информационно-управляющих систем	42
2.2. Непрерывно–детерминированные модели (<i>D</i> – схемы)	45
2.3. Дискретно–детерминированные модели (<i>F</i> – схемы)	47
2.4. Дискретно – непрерывные модели	49
2.5. Дискретно – стохастические модели (<i>P</i> – схемы)	49
2.6. Непрерывно–стохастические модели (<i>Q</i> – схемы)	53
2.7. Стохастические минимаксные модели	57
2.8. Особенности моделирования инфокоммуникационных систем	58
Вопросы для самопроверки	61
3. ФОРМАЛИЗАЦИЯ И АЛГОРИТМИЗАЦИЯ ПРОЦЕССОВ ФУНКЦИОНИРОВАНИЯ СИСТЕМ	61
3.1. Этапы моделирования информационно-управляющих систем	61
3.2. Способы описания алгоритмов	67
3.2.1. Словесный способ представления алгоритмов	68
3.2.2. Графический способ представления алгоритмов	68
3.2.3. Графические способы описания алгоритмов работы информационных систем (промышленных систем)	72
Вопросы для самопроверки	74
4. ПЛАНИРОВАНИЕ МАШИННЫХ ЭКСПЕРИМЕНТОВ С МОДЕЛЯМИ СИСТЕМ	75
4.1. Стратегия эффективного планирования машинного эксперимента	75
4.2. Выбор и анализ эмпирических моделей	75
4.3. Методы планирования эксперимента на модели	79
4.4. Проверка адекватности модели	83
4.4.1. Критерий Фишера	83
4.4.2. Определение дисперсий неточности модели и ошибки эксперимента	83
4.4.3. Определение дисперсии воспроизводимости эксперимента	84
4.4.4. Проверка однородности дисперсий	84
4.5. Проверка значимости коэффициентов модели	85

4.6. Стратегическое планирование машинных экспериментов с моделями систем	85
4.7. Tактическое планирование машинных экспериментов с моделями систем	86
4.8. Планирование экспериментов при построении полной квадратичной модели	89
4.8.1. Ортогональное центральное композиционное планирование	91
4.8.2. Рототабельное композиционное планирование	94
4.8.3. Разбиение матрицы планирования 2^K на блоки	96
4.9. Критерии оптимальности планов	97
4.10. Правило автоматической остановки имитационного эксперимента	100
4.11. Принятие решений после построения модели процесса	102
Вопросы для самопроверки	102
5. МЕТОДЫ ОПРЕДЕЛЕНИЯ ХАРАКТЕРИСТИК МОДЕЛИРУЕМЫХ СИСТЕМ	102
5.1. Измеряемые характеристики моделируемых систем	102
5.2. Расчёт математического ожидания и дисперсии выходной характеристики	103
5.3. Расчёт среднего по времени значения выходной характеристики	104
5.4. Построение гистограммы для стационарной системы	104
5.5. Моделирование случайных воздействий	105
5.6. Преобразование случайных величин	106
5.7. Моделирование нормально распределённой случайной величины	108
6. ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ МОДЕЛИРОВАНИЯ	108
6.1. CASE-технологии	108
6.2. Системы моделирования электронных устройств информационно-управляющих систем	112
6.3. Моделирование полупроводниковых преобразователей	126
6.4. Программные средства для решения задач моделирования ИУС	126
6.5. MATLAB Simulink и его приложения	138
6.6. Специализированный Toolbox Sim Power Systems	140
6.6.1. Источники электрической энергии	141
6.6.2. Электротехнические элементы	142
6.6.3. Устройства силовой электроники	143
6.6.4. Измерительные и контрольные устройства	144
6.6.5. Расширенные библиотеки	144
6.6.6. Трансформаторы	144
6.7. Оптимизация параметров ИУС	145

6.8. Методика оптимизации параметров ИУС	147
6.9. Программные средства для оптимизации	149
6.9.1. MATLAB Optimization Toolbox	151
7. КОМПЬЮТЕРНЫЕ МЕТОДЫ ИССЛЕДОВАНИЯ С ВИЗУАЛИЗАЦИЕЙ ДИНАМИЧЕСКИХ ПРОЦЕССОВ В ИУС	152
7.1. Средства визуального моделирования динамических систем	152
7.2. UML как средство проектирования моделей динамических систем	153
7.3. Классы моделей и технология их реализации в пакетах визуального моделирования	155
7.4. Технология моделирования в инструментальной среде Model Vision Stadium	159
7.5. Применение среды LabVIEW для исследования систем управления ИУС	160
СПИСОК ЛИТЕРАТУРЫ	163