

Министерство науки и высшего образования Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем
в управлении и проектировании (КСУП)

В. М. Дмитриев, Т. В. Ганджа, Т. Е. Григорьева

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СИСТЕМ

Курс лекций

Томск
2020

УДК 004.94(075.8)
ББК 32.973.2-018я73
Д 534

Рецензент

А. Г. Горюнов, д-р техн. наук, руководитель Отделения
ядерно-топливного цикла Инженерной школы ядерных технологий
Томского политехнического университета

Дмитриев В. М. и др.

Д 534 Компьютерное моделирование систем : курс лекций /
В. М. Дмитриев, Т. В. Ганджа, Т. Е. Григорьева. – Томск : Эль Кон-
тент, 2020. – 260 с.

ISBN 978-5-4332-0284-9

Курс лекций «Компьютерное моделирование систем» охватывает современные методы, алгоритмы и программные средства моделирования систем с непрерывным, дискретно-событийным и гибридным поведением.

Для студентов технических специальностей, связанных с проектированием и исследованием сложных технических управляемых систем, систем массового обслуживания, а также с информационными технологиями и системами управления социально-экономическими процессами.

ISBN 978-5-4332-0284-9

© Дмитриев В. М.,
Ганджа Т. В.,
Григорьева Т. Е., 2020
© Оформление.
Эль Контент, 2020

Оглавление

Введение.....	7
1 Моделирование, понятия, классификация	
и задачи моделирования	9
1.1 Основные понятия теории моделирования	9
1.2 Классификация видов моделирования систем.....	10
1.3 СММ МАРС – инструмент компьютерного моделирования нового типа	13
2 Виды моделей.....	16
2.1 Статические и динамические модели.....	16
2.2 Непрерывные, дискретные и гибридные модели	16
2.3 Непрерывные и стохастические модели	22
2.4 Математические и имитационные модели.....	22
3 Моделирование на микроуровне	24
3.1 Краевые задачи при моделировании технических объектов	24
3.1.1 Примеры уравнений, составляющих основу моделей объектов на микроуровне	24
3.1.2 Краевые условия	25
3.1.3 Приближенные модели задач микроуровня.....	26
3.2 Метод конечных элементов	27
4 Теоретические основы метода компонентных цепей	33
4.1 Основы формализма метода компонентных цепей	33
4.2 Метод компонентных цепей как язык моделирования СТУС	33
4.3. Методы решения модели	40
5 Методы формирования моделей для схемотехнического	
моделирования во временной области	44
5.1 Критерии выбора метода моделирования	44
5.2 Метод узловых потенциалов.....	45
5.3 Модифицированный метод узловых потенциалов	46
5.4 Метод КЦ в матричной форме.....	46
5.5 Метод переменных состояния	49
6 Операторно-структурные схемы и графы систем.....	52
6.1 ОСС линейных стационарных непрерывных систем	52
6.2 Правила преобразований ОСС линейных систем.....	55
7 Операторно-структурные схемы	
линейных многомерных систем	58

7.1	Операторно-структурные схемы линейных стационарных систем.....	58
7.2	Операторно-структурные схемы линейных нестационарных непрерывных систем	60
7.3	Операторно-структурные схемы и графы нелинейных систем	60
7.4	Операторно-структурные схемы дискретных и дискретно-непрерывных систем.....	61
8	Конечные автоматы.....	64
8.1	Способы описания конечного автомата	64
8.2	Примеры конечных автоматов.....	66
8.3	Системы, предоставляющие возможность построения конечных автоматов	70
9	Гибридные динамические системы.....	71
9.1	Прыгающий мячик	71
9.2	Время.....	73
9.3	Пространство	74
10	Гибридные автоматы.....	79
10.1	Типы гибридных систем	79
10.2	Смена начальных условий	79
10.3	Зависимость от параметров.....	81
10.4	Карты состояний и гибридные автоматы	83
11	Разработка моделей дискретно-событийных систем	87
11.1	Модели дискретно-событийных систем.....	87
11.2	Stateflow 5 как программный продукт.....	94
11.3	Проектирование моделей реактивных систем	95
12	Анализ динамических цепей.....	97
12.1	Уравнения состояния электрических цепей.....	97
12.2	Алгоритм формирования уравнений состояния	100
12.3	Методы дискретного отображения. Прямое Z-преобразование цифровых сигналов.....	104
13	Анализ нелинейных систем	108
13.1	Основные понятия	108
13.2	Численное решение уравнений нелинейных резистивных цепей. Метод Ньютона – Рафсона.....	110
13.3	Дискретные схемы замещения нелинейных резистивных цепей	113

14 Моделирование дискретных систем	118
14.1 Математическое описание систем дискретного управления	118
14.2 Модели состояния линейной дискретной системы	127
15 Агрегативные модели	129
15.1 Определение дискретно-событийной системы	129
15.2 Агрегативные модели. Кусочно-линейный агрегат	130
15.3 Схема сопряжения агрегативной системы	131
15.4 Агрегативная модель укладки бетона	132
15.5 Оценка агрегативных систем как моделей сложных систем	135
16 Язык моделирования алгоритмических конструкций	137
16.1 Грамматика языка моделирования алгоритмических конструкций	137
16.2 Операторы языка МАК	139
16.3 Операнды языка МАК	139
16.4 Отображение операторов языка МАК	139
16.5 Отображение функций языка МАК в алгоритмических компонентных цепях	142
17 Системная динамика	146
17.1 Нотация и основные идеи системной динамики	146
17.2 Концепция и терминология системной динамики	147
17.3 Формализм метода компонентных цепей для интерпретации моделей системной динамики	148
Пример. Гидросистема из двух баков	151
18 Базовая модель системной динамики экономико-экологической системы	152
19 Системы массового обслуживания	161
19.1 Классификация систем массового обслуживания и их показатели эффективности	161
19.2 Моделирование системы массового обслуживания: основные параметры, граф состояний	162
19.3 Сети Петри для моделирования СМО	163
19.4 Структура сети Петри	165
19.5 Правила выполнения сетей Петри	168
20 Классификация сетей Петри	173
20.1 Автоматные сети Петри	173
20.2 Маркированные сети	173

20.3	Виды расширенных сетей Петри	179
21	Интерпретация сетей Петри в МКЦ.....	184
21.1	Основные понятия и определения	184
21.2	Разработка компонентов сетей Петри в СММ МАРС	186
21.3	Компоненты, реализованные в СММ МАРС	188
22	Метод нейронных сетей.....	195
22.1	Основные определения	195
22.2	Решаемые задачи	195
22.3	Элементы нейронных сетей	196
22.4	Архитектура нейронных сетей	198
23	Статистическое моделирование.....	203
23.1	Основные понятия	203
23.2	Реализация компонентов класса для функций распределения	206
23.3	Метод Монте-Карло	208
23.4	Построение гистограмм	209
24	Методы оптимизации в системах моделирования	212
24.1	Структурная схема оптимизации.....	212
24.2	Программы оптимизации в СММ МАРС.....	217
24.3	Пример задачи параметрического синтеза.....	219
25	Измерения и обработка результатов моделирования.....	224
25.1	Компоненты-блоки для обработки результатов моделирования .	224
25.2	Блоки обработки результатов стационарных колебательных процессов	227
26	Компьютерные модели в системах поддержки принятия решений.....	235
26.1	Системы поддержки принятия решений	235
26.2	Структура СППР	236
26.3	Подсистема моделей.....	237
27	Алгоритм функционирования программы компьютерного моделирования (PSPICE).....	243
27.1	Основные сведения	243
27.2	Функциональные возможности среды PSpice.....	245
27.3	Алгоритм функционирования SPICE	245
28	Среда многоуровневого компьютерного моделирования	251
	Список сокращений	258

Введение

Моделирование – это общепризнанное средство познания действительности, которое *состоит из двух этапов*: разработки модели и ее анализа. Первый этап связан с формализованным представлением *объекта-оригинала к формату объекта-заместителя*, то есть модели. Второй этап позволяет исследовать сложные процессы и явления на основе экспериментов не с реальной системой, а с ее моделью.

Предлагаемый материал представляет собой *комплексный набор методов моделирования*, составленный авторами при анализе современной литературы по системотехнике и компьютерному моделированию. В начале каждой лекции указаны ключевые понятия, которые раскрываются по ходу изложения. В конце лекции приведен список контрольных вопросов и литературных источников.

В начале курса рассматриваются общие теоретические вопросы предметной области системотехники и компьютерного моделирования. Приводятся сведения, необходимые для понимания методологических принципов, лежащих в основе того или иного метода анализа предметной области модели. В инженерных задачах обычно применяется *схемотехническое моделирование* для дискретных и полевых структур, в основном при проведении исследований на этапах проектирования, внедрения и эксплуатации сложных технических устройств и систем (СТУС). На этапах разработки технического и рабочего проектов систем моделирование служит для решения конкретных задач проектирования, выбора оптимального по определенному критерию и при заданных ограничениях варианта системы из множества допустимых вариантов или для синтеза сложных систем. Далее рассматриваются правила описания технических систем с помощью *операционных структурных схем (ОСС)* в пространстве состояния, дающие возможности настройки регулятора состояния по полному вектору состояния. Даются необходимые сведения *анализа непрерывных и дискретных, а также нелинейных систем*. Приводятся теоретические основы *конечных и гибридных автоматов и применение карт состояний*.

Имитационное моделирование представлено методами системной динамики, построением моделей на основе *теории массового обслуживания*, обсуждаются концепции построения моделей на основе классификации *сетей Петри и агрегатов*. Изучаются правила построения *агентных моделей*, состоящих из

гомогенных и гетерогенных агентов. Даются основы *статистического моделирования*, включающие отображение явлений и процессов с помощью случайных (стохастических) событий и их поведения, которые описываются соответствующими *вероятностными (статистическими) характеристиками*. Рассматривается одна из важных задач исследования систем и технических объектов в плане *параметрического синтеза*, в которой по известным входным воздействиям и откликам цепи, а также структуре объекта производится поиск его параметров, оптимальных в рамках определенного критерия. Приводятся сведения, необходимые для обработки результатов измерений эксперимента с помощью виртуальных приборов и блоков обработки данных. В заключительной части приводятся сведения о *системе компьютерного моделирования МАРС* в объеме, необходимом для понимания его структуры и алгоритма функционирования, а также его настройки на решаемую задачу.

Данное учебное пособие представляет собой конспект лекций для семестрового курса обучения по дисциплинам, связанным с компьютерным моделированием и системами управления для технических направлений подготовки.

1 Моделирование, понятия, классификация и задачи моделирования

1.1 Основные понятия теории моделирования

Моделирование является общепризнанным средством познания действительности. Этот процесс состоит из двух больших этапов: разработки модели и анализа разработанной модели. Моделирование позволяет исследовать суть сложных процессов или явлений с помощью экспериментов.

В области создания новых систем моделирование является средством исследования важных характеристик будущей системы на ранних стадиях ее разработки; с его помощью возможно исследовать узкие места будущей системы, оценивать производительность, стоимость, пропускную способность – все главные характеристики системы до ее создания.

С помощью моделей разрабатываются оптимальные операционные планы и расписания функционирования сложных систем. В организационных системах имитационное моделирование становится основным инструментом сравнения различных вариантов управляющих решений и поиска наиболее эффективного из них как на микро-, так и на макроэкономическом уровне.

Введем *основные понятия моделирования*.

Система – это совокупность элементов, которые принадлежат ограниченной части реального мира, являющейся объектом исследования.

Модель – это описание системы.

Теория замещения одних объектов (оригиналов) другими объектами (моделями) и исследование свойств объектов на их моделях называется *теорией моделирования*.

В научных исследованиях большую роль играют *гипотезы*, т. е. определенные предсказания, основывающиеся на опытных данных, наблюдениях, интуиции.

При формировании и проверке правильности гипотез большое значение в качестве метода суждения имеет *аналогия*.

Аналогией называется суждение о каком-либо частном сходстве двух объектов. С учетом этих понятий дадим другое понятие модели.

Модель – это объект-заместитель объекта-оригинала, отдельные свойства которого полностью или частично совпадают со свойствами оригинала, обеспечивающий изучение некоторых свойств объекта-оригинала.

Замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала с помощью объекта-модели называется *моделированием*.

Математическое моделирование – это процесс замещения оригинала *математической моделью*, обеспечивающей исследование свойств и отношений оригинала [1].

Особое значение среди математических моделей имеют *подобные*, обеспечивающие перенос данных на оригинал на основании *подобия*.

Подобие – это полная математическая аналогия при наличии пропорциональности между сходственными переменными, неизменно сохраняющаяся при всех возможных значениях этих переменных, удовлетворяющих сходственным уравнениям.

Геометрическое подобие – это подобие геометрических образов: точек, линий, поверхностей, фигур, тел. Широко распространен этот вид моделирования в архитектуре, дизайнерской работе и т. п.

Физическое подобие означает подобие физически однородных объектов. Все масштабы при этом являются безразмерными величинами. В теории и практике моделирования *подобие* имеет большее значение, чем *аналогия*. При *анalogии* двух объектов распространение свойств одного из них на другой носит характер предположения и нуждается в проверке. При *подобии* двух объектов знание поведения одного из них означает знание поведения другого.

В основе моделирования лежит теория подобия, которая утверждает, что абсолютное подобие может иметь место лишь при замене одного объекта другим, точно таким же. Поэтому стремятся к тому, чтобы модель достаточно хорошо отображала исследуемую сторону функционирования объекта. В соответствии с вышесказанным классификацию видов моделирования можно проводить по степени полноты моделей и разделить их в соответствии с этим признаком: *полные, неполные, приближенные*.

1.2 Классификация видов моделирования систем

Как уже было отмечено выше, первичная классификация моделей может быть проведена по степени полноты модели и по характеристикам процессов, протекающих в исследуемом объекте (рис. 1.1).



Рис. 1.1 – Классификация видов моделирования систем

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

Стохастическое моделирование отображает вероятные процессы и события, обусловленные случайными воздействиями.

Статическое моделирование служит для описания поведения объектов в какой-либо момент времени.

Динамическое моделирование отражает поведение объекта во времени.

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

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

Аналоговое моделирование основывается на применении аналогий различных уровней.

Макетирование мысленное применяется в случаях, когда моделируемый процесс не поддается физическому моделированию, либо в случаях, когда это предшествует другим видам моделирования.

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

В основе *языкового моделирования* лежит некоторый *тезаурус*.

Тезаурус – словарь, в котором каждому слову может соответствовать лишь единственное понятие.

Математическое моделирование – процесс установления соответствия данному реальному объекту некоторого математического объекта, называемого математической моделью, и исследование этой модели с целью получения характеристик реального объекта.

Аналитическое моделирование характеризуется тем, что процессы функционирования элементов системы записываются в виде функциональных соотношений (алгебраических, разностных, интегро-дифференциальных и т. п.) или логических условий.

Комбинированное моделирование – это разумное сочетание аналитического (на первом этапе) и имитационного (на последующих этапах) моделирования, позволяющее создавать модели более сложных систем.

Имитационное моделирование реализует модель алгоритма, воспроизводящего процесс функционирования системы во времени, причем имитируются элементарные явления с сохранением их логической структуры и связей в системе и последовательности протекания во времени.

Основные преимущества имитационного моделирования по сравнению с аналитическим – возможность моделирования сложных систем.

Реальное моделирование использует возможность исследования различных характеристик системы либо на реальном объекте, либо на его части.

Натурное моделирование – проведение исследований на реальном объекте с последующей обработкой результатов на основе теории подобия.

Производственный эксперимент и комплексные испытания обладают высокой степенью достоверности, однако по ряду ограничений не всегда возможны: невозможно исследовать критические режимы, тонкие эффекты, нереально исследовать все функциональные зависимости, и т. п.

Научный эксперимент в настоящее время завоевал широкое распространение благодаря использованию средств автоматизации и управления, а также благодаря применению разнообразных вычислительных средств обработки и визуализации информации. В научном эксперименте имеется возможность вмешательства человека в процесс проведения эксперимента и, благодаря этому, появилось новое направление – *автоматизация научных экспериментов*.

Другим видом реального моделирования является *физическое моделирование*.

Физическое моделирование отличается от натурного тем, что исследования проводятся на установках, которые сохраняют природу явлений и обладают физическим подобием [2].

1.3 СММ МАРС – инструмент компьютерного моделирования нового типа

В работах научного коллектива под руководством профессора В. М. Дмитриева разработаны системные подходы, развивающие теорию цепей общего вида (компонентных цепей) как методологическую основу алгоритмического и программного аппарата автоматизированного моделирования СТУС, а также социально-экономических систем, реализованного в рамках СММ МАРС [3].

Программное обеспечение, так или иначе используемое при исследовании СТУС, можно разделить на три группы:

- системы автоматизированного моделирования и проектирования радиоэлектронных схем (*PCAD, OrCAD, Micro-CAP, PSpice, MicroSim Design Center (PSPICE), MicroSim Design Lab, Electronic Workbench* (теперь *MultiSim*));
- системы автоматизированного моделирования виртуальных инструментов и приборов (*LabVIEW*);
- системы автоматизированного моделирования и проектирования в области машиностроения (*AvtoCAD, ProEngineer*, отечественные разработки «КомПАС», «Эйлер» и «Спрут»);
- универсальные системы моделирования (*MATLAB, Stratum, CM MАРС*);
- системы автоматизации математических вычислений (*MATCAD, MAPL, «Макрокалькулятор»*);
- системы имитационного моделирования систем управления и бизнес-процессов (*AnyLogic, AIRYS*).

Исследовать элементы из любой предметной области позволяют *универсальные системы моделирования*. В настоящее время наиболее популярной при исследовании сложных технических объектов является система *MATLAB*. Однако при использовании большинства описаний этой системы, не имеющих проблемной ориентации, у пользователей со специфической сферой профессиональных интересов при изучении системы возникают трудности. Еще большие трудности возникают у пользователей при исследовании сложных физически неоднородных систем, так как в этом случае пользователю предстоит овладеть всеми техническими ресурсами данной системы и попытаться эффективно *совместить и применить* их на практике.

Система МАРС базируется на основах *формализма метода компонентных цепей*, которые были заложены в приложении к электрическим цепям и далее последовательно развивались в различных работах применительно к механическим, электромеханическим цепям и системам автоматического управления. Система МАРС позволяет быстро создавать *исполняемую модель – виртуальный прототип разрабатываемой системы* и ее окружения, в том числе модели физически неоднородных систем и математических задач. Используя построенную модель, уже на ранней стадии разработки, в удобной и безопасной среде можно оценить, насколько удачны выбранная структура и параметры системы. Уникальные технологии, положенные в основу вычислительного ядра, делают МАРС эффективным инструментом для разработки и отладки больших и сложных систем, когда эксперименты с реальным прототипом или самой системой требуют много времени, финансовых средств или же совсем невозможны.

Контрольные вопросы

1. Из каких двух больших этапов состоит процесс моделирования?
2. Средством исследования каких характеристик технических систем служит моделирование?
3. Дайте понятия системы и модели системы.
4. Что такое математическое моделирование?
5. Дайте классификацию типов моделирования.
6. Что такое имитационное моделирование и его отличие от математического?
7. На основе какого формализма базируется система МАРС?

Литература

1. Колесов, Ю. Б. Моделирование систем. Объектно-ориентированный подход / Ю. Б. Колесов, Ю. Б. Сениченков. – СПб. : БХВ-Петербург, 2006. – 192 с.
2. Музынкин, С. Н. Моделирование динамических систем / С. Н. Музынкин, Ю. М. Родионова. – Ярославль : Верхне-Волжское книжное издательство, 1984. – 304 с.
3. Дмитриев, В. М. МАРС – среда моделирования технических устройств и систем / В. М. Дмитриев, А. В. Шутенков, Т. Н. Зайченко, Т. В. Ганджа. – Томск : В-Спектр, 2011. – 278 с.

2 Виды моделей

2.1 Статические и динамические модели

Статические модели оперируют характеристиками и объектами, не изменяющимися во времени. Они обычно имеют дело с установившимися процессами, уравнениями балансового типа, такими как уравнения материального баланса или уравнения экологического равновесия.

В *динамических моделях*, которые обычно более сложны, изменение параметров во времени является более существенным. Модели транспортной системы города, работы аэропорта являются *динамическими*. В них моделируется поведение транспортных единиц, движение цистерн заправки, уровень бензина в бензоколонке. *Моделирование динамических систем* (ДС) состоит в имитации правил перехода системы из одного состояния в другое с течением времени. *Изменение состояний в ДС во времени* – это изменение переменных системы в соответствии с законами, определяющими связи переменных и их зависимости друг от друга во времени.

В соответствии с подходом к классификации типов моделирования СТУС можно укрупненно выделить два подхода к моделированию – *информационный* и *физический*. При *информационном подходе* определяется лишь преобразование входного сигнала в выходной без учета внутренних физических законов сохранения или равновесия, определяющих или сопровождающих это преобразование. *Информационные модели* характеризуются однонаправленным прохождением информации, для которого известна функция или алгоритм преобразования входного сигнала в выходной [1].

При *физическом подходе* моделирование выполняется с учетом реальных физических законов равновесия (законов Кирхгофа) или сохранения (уравнений непрерывности).

2.2 Непрерывные, дискретные и гибридные модели

Реальные физические объекты функционируют во времени и для изучения их характеристик модели должны быть *непрерывными*. Состояние таких моделей изменяется непрерывно во времени. Это модели электротехнических систем, механических систем движения в реальных координатах, модели химического про-

изводства и т. п. Процессы движения объектов и процессы транспортировки нефти и газа по трубопроводам являются *непрерывными*. В соответствии со способом представления исследуемых объектов и применяемым математическим аппаратом различают *схемотехническое, функциональное, логическое, регистровое и структурное моделирование*. *Физический (энергетический) подход* используется при схемотехническом моделировании и моделировании отдельных компонентов, а *информационный* – при структурном и функциональном. Следует отметить, что на связях элементов схем при информационном подходе действует одна переменная, а при физическом – две (P и F), одна из которых имеет смысл потенциальной величины, например напряжения, а другая – потоковой величины, например тока.

Схемотехническое моделирование – моделирование электрических процессов в электронных устройствах, изображаемых в виде принципиальных электрических схем. *Целями моделирования* являются расчет формы и параметров сигналов тока и напряжения в разных точках схемы.

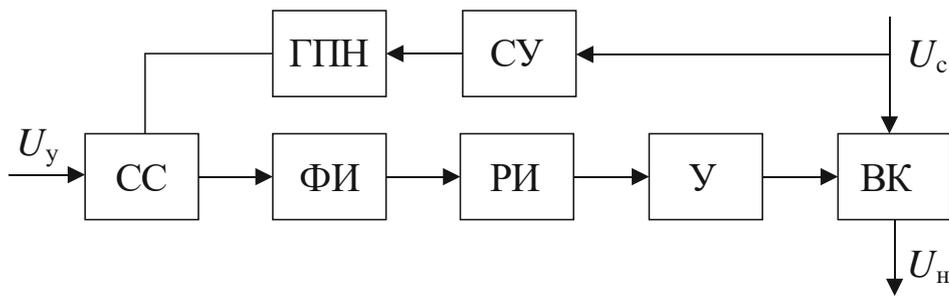
Целью исследования цепей во временной области является расчет отклика выходного сигнала $x_{\text{ВЫХ}}(t)$ либо фазовых переменных на входной сигнал $x_{\text{ВХ}}(t)$. *Классическими методами* формирования математических моделей радиоэлектронных схем являются *методы узловых потенциалов и контурных токов*. *Традиционные машинно-ориентированные методы* включают *прямые методы формирования моделей* – табличный, модифицированный узловых потенциалов, метод переменных состояния.

Исследования в частотной области базируются на определении комплексного коэффициента передачи:

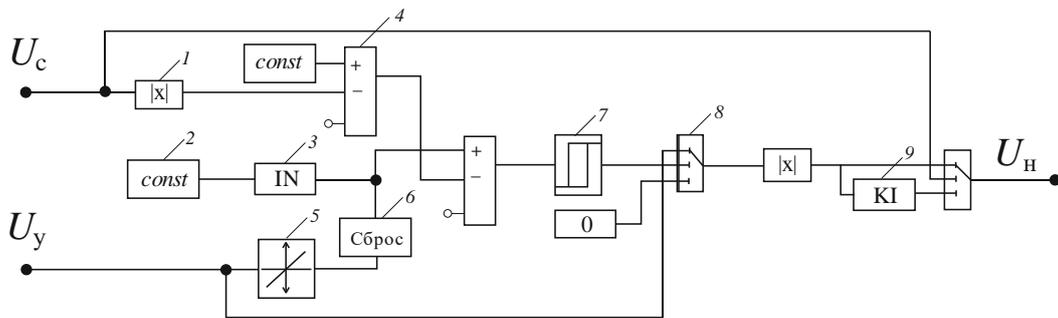
$$K(\omega) = \frac{X_{\text{ВЫХ}}}{X_{\text{ВХ}}} = K(\omega)e^{j\varphi(\omega)} = \frac{P_1(\omega) + jP_2(\omega)}{Q_1(\omega) + jQ_2(\omega)} \quad (2.1)$$

и частотных характеристик: амплитудно-частотной (АЧХ) $K(\omega) = |K(\omega)|$, фазо-частотной (ФЧХ) $\varphi(\omega) = \arg(K(\omega))$ и т. п.

Функциональное моделирование – моделирование электрических процессов в электронных устройствах, изображаемых в виде функциональных схем (рис. 2.1).



а)



б)

Рис. 2.1 – Структурная (а) и функциональная (б) схемы управляемого выпрямителя с вертикальным способом управления: а) ГПН – генератор пилообразного напряжения; СУ – синхронизирующее устройство; СС – схема сравнения; ФИ – формирователь импульсов; РИ – распределитель импульсов; У – усилители мощности; ВК – вентильный комплект; U_y – напряжение сигнала управления; U_c – напряжение сети; U_H – выходное напряжение нагрузки; б) 1 – элемент определения модуля; 2 – генератор сигнала постоянной величины; 3 – интегрирующее звено; 4 – сумматор; 5 – элемент, контролирующий момент пересечения переменной связи заданного уровня; 6 – элемент, осуществляющий сброс значения выходной переменной связи в соответствии со значением входной переменной; 7 – идеальное реле; 8 – переключатель; 9 – пропорциональное звено

Функциональное моделирование является способом моделирования радиоэлектронной аппаратуры общего назначения. *Базовыми элементами функциональных схем* являются: генераторы сигналов (независимые и управляемые), безынерционные элементы (аналого-аналоговые, аналого-цифровые, логико-аналоговые и т. п. преобразователи), инерционные линейные элементы (фильтры, усилители, выполняющие функции суммирования, интегрирования, дифференцирования и т. п.), инерционные нелинейные элементы. Характерным примером задач, решаемых *методами функционального моделирования*, является исследование поведения систем автоматического управления (САУ), функциональные схемы которых состоят из типовых звеньев – пропорциональных, дифференцирующих, интегрирующих и т. п. (рис. 2.1).

Первым допущением, характерным для функционального моделирования, является развязка отдельных блоков функциональной схемы, т. е. независимость характеристик отдельных блоков от режима работы других блоков.

Вторым допущением – однонаправленность элементов, т. е. сигнал на выходе любого элемента не влияет на сигнал на его входе. Считается, что сигнал в функциональных схемах распространяется *однонаправленно* – от входа к выходу каждого элемента.

Модель элемента функциональной схемы в общем случае задается системой уравнений:

$$\begin{aligned} X_{\text{ВЫХ}} &= F_0(X_{\text{ВХ}}, X_{\text{ВН}}); \\ X'_{\text{ВН}} &= F_1(X_{\text{ВХ}}, X_{\text{ВН}}), \end{aligned} \quad (2.2)$$

где $X_{\text{ВЫХ}}$, $X_{\text{ВХ}}$ – векторы выходных и входных переменных элемента; $X_{\text{ВН}}$, $X'_{\text{ВН}}$ – векторы внутренних переменных, характеризующих состояние элемента. Элементы векторов $X_{\text{ВХ}}$, $X_{\text{ВН}}$, относятся к моменту времени t , а векторов $X_{\text{ВЫХ}}$, $X'_{\text{ВН}}$ – к моменту времени $(t - k_j T)$, т. е. являются *задержанными*.

Математическая модель функциональной схемы – совокупность математических моделей элементов, в которых произведено отождествление фазовых переменных, относящихся к соединяемым выводам элементов (равенство переменных в узлах).

На более высоком уровне абстракции адекватными являются *модели систем*, в которых переходы систем из одного состояния в другое можно *считать мгновенными*, происходящими в *дискретные моменты времени*. Такие системы называются *дискретными*. *Примером дискретных моделей* являются процессы в дискретных электронных схемах или изменение числа клиентов банка или числа покупателей в магазине. Результаты анализа полученной дискретной модели обычно достаточны для принятия обоснованных управленческих решений.

Логическое моделирование – моделирование устройств, изображаемых в виде логической схемы (рис. 2.2), т. е. схемы на уровне простейших логических элементов (*НЕ, И, ИЛИ* и т. п.).

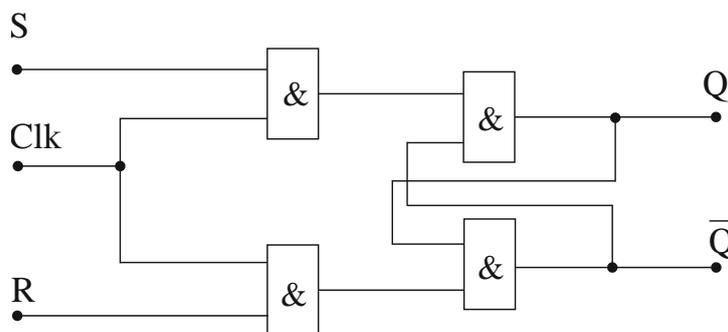


Рис. 2.2 – Логическая схема RSC-триггера, реализованная на логических элементах И

При логическом моделировании решаются задачи *проверки логики работы схем, анализа переходных процессов*, определение *надежности работы схем* в зависимости от разброса параметров комплектующих элементов. В зависимости от поставленной задачи выбирается метод моделирования. Основными *отличительными чертами методов* являются:

- способ учета времени распространения сигнала в схеме;
- способ кодирования сигналов;
- способ построения компьютерной модели;
- очередность моделирования элементов.

Методы классифицируются по различным основаниям:

- в зависимости от способа учета времени распространения сигнала методы делятся на *синхронный* (без учета задержек в элементах схемы) и *асинхронный* (с учетом задержек);
- в зависимости от способа представления сигналов – на *двоичный* и *многозначный* (троичный, пятизначный и т. д.);
- по способу организации работы программы моделирования – на *метод компиляции* и *метод интерпретации*;
- по организации очередности моделирования – *событийный* (моделирование только активированных элементов, т. е. элементов, у которых хотя бы на одном входе произошло событие – изменилась входная переменная) и *сквозной* (моделирование всех элементов).

При логическом моделировании используются функциональные модели элементов – модели, представленные в виде «черного ящика», для которого связь между входными и выходными сигналами задается в виде булевых уравнений либо таблиц истинности.

Для функционально-логического моделирования функциональных схем аналоговой электроники используются два подхода:

- электрическое макромоделирование;
- функциональное моделирование, развитое в теории автоматического управления.

Функционально-логическое моделирование функциональных схем цифровой РЭА связано с особенностями:

- 1) состояние элементов схем характеризуется фазовыми переменными одного типа, обозначающими хранимую или передаваемую информацию; физическая природа фазовой переменной (напряжение или ток) не конкретизируется;
- 2) фазовые переменные целесообразно представлять в дискретной форме, поскольку информация имеет цифровую форму. Обычно фазовые переменные могут принимать значения из множества $\{0, 1\}$;
- 3) анализ функциональных схем производится в дискретном времени. Ось времени разделяется на такты длительностью T .

На еще более высоком уровне абстракции при анализе систем также используются непрерывные модели, что характерно для *системной динамики*. Потоки машин на автострадах, эколого-экономические системы (ЭЭС) (рис. 2.3), распространение инфекции среди населения, часто удобно представлять непрерывными зависимостями, описывающими количества, степени влияния этих количеств, степени влияния одних количеств на другие. Соотношения таких переменных выражаются обычно дифференциальными уравнениями.

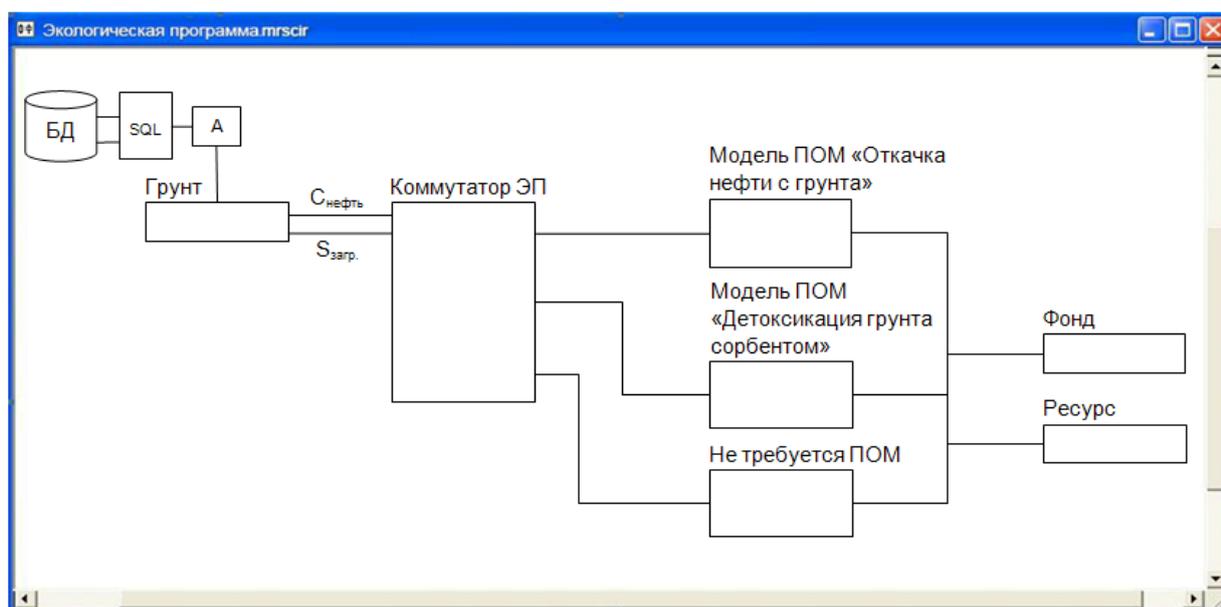


Рис. 2.3 – Фрагмент компьютерной модели ЭЭС

Во многих случаях в реальных системах присутствуют оба типа процессов. И если оба они являются существенными для анализа системы, то и в модели должны сочетаться *как непрерывные, так и дискретные*. Такие модели со смешанным типом процессов называются *гибридными*. Примером здесь может служить модель функционирования крупного банка. Поток инвестиций, получение и выдача кредитов описывается обычно набором дифференциальных уравнений (непрерывная модель). Однако в случае наступления финансовых катаклизмов (дискретное событие) система описывается уже другой непрерывной моделью.

2.3 Непрерывные и стохастические модели

При моделировании сложных реальных систем исследователь часто сталкивается с ситуациями, когда случайные воздействия играют существенную роль. *Стохастические модели* в отличие от *детерминированных*, учитывают вероятностный характер параметров моделируемого объекта. Например, в модели нефтеналивного порта не могут быть точно определены моменты прихода в порт танкеров, которые являются *случайными величинами*. Поэтому такая модель является *стохастической*. Анализ подобных моделей выполняется в ходе имитационного эксперимента при многократном прогоне модели для различных значений исходных случайных величин [2].

2.4 Математические и имитационные модели

Некоторые динамические системы можно описать в замкнутой форме, например в виде системы алгебро-дифференциальных уравнений. В некоторых случаях анализ этих соотношений можно выполнить с помощью *аналитических преобразований*. В более общих случаях применяются *численные методы анализа*. Все это образует направление *математического моделирования*. Даже если аналитические модели удастся построить, они часто являются существенно нелинейными, дополненными логическими соотношениями, а для них аналитического решения не существует.

При имитационном моделировании структура системы (блоки и связи) представлена структурой модели, а процесс функционирования подсистем, выраженный в виде уравнений и правил, связывающих переменные, имитируется на компьютере. *Имитационные модели* позволяют имитировать работу моделируемой системы фактически с любой желаемой степенью адекватности и выполнять анализ модели без проведения аналитических преобразований [3].

Существует ряд методов и языков имитационного моделирования, которые мы рассмотрим ниже.

Контрольные вопросы

1. Назовите основные отличия статических моделей от динамических.
2. Чем различаются информационный и физический подходы?
3. Назовите основные характеристики функционально-логического моделирования.
4. В чем состоят особенности дискретного и непрерывного моделирования?
5. В чем состоит принципиальное отличие имитационного моделирования от математического?

Литература

1. Музынкин, С. Н. Моделирование динамических систем / С. Н. Музынкин, Ю. М. Родионова. – Ярославль : Верхне-Волжское книжное издательство, 1984. – 304 с.
2. Колесов, Ю. Б. Моделирование систем. Объектно-ориентированный подход / Ю. Б. Колесов, Ю. Б. Сениченков. – СПб. : БХВ-Петербург, 2006. – 192 с.
3. Емельянов, А. А. Имитационное моделирование экономических процессов / А. А. Емельянов, Е. А. Власова, Р. В. Дума. – М. : Финансы и статистика, ИНФРА-М, 2009. – 416 с.

3 Моделирование на микроуровне

3.1 Краевые задачи при моделировании технических объектов

3.1.1 Примеры уравнений, составляющих основу моделей объектов на микроуровне

Первая важная задача проектирования летательного аппарата – определение прочности узлов и элементов конструкции при различных видах нагружения [1].

Напряженное состояние деталей конструкции в зависимости от геометрии исследуемого узла, вида приложенной нагрузки и свойств материала описываются в общем случае уравнениями вида

$$K_x \left(\frac{\partial^2 \varphi}{\partial x^2} \right) + K_y \left(\frac{\partial^2 \varphi}{\partial y^2} \right) + K_z \left(\frac{\partial^2 \varphi}{\partial z^2} \right) + Q = 0, \quad (3.1)$$

где x, y, z – пространственные координаты; φ – искомая непрерывная функция; K_x, K_y, K_z – коэффициенты; Q – внешнее воздействие.

В двумерном случае при $K_x = K_y = 1$ уравнение (3.1) сводится к уравнению, описывающему напряженное состояние, возникающее в поперечном сечении упругого однородного стержня под воздействием крутящего момента M :

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + 2E\Theta = 0, \quad (3.2)$$

где E – модуль сдвига материала стержня; Θ – угол закручивания на единицу длины, а функция φ – функция, связанная с напряжениями сдвига и с крутящим моментом.

$$M = 2 \int_S \varphi dS,$$

где S – площадь рассматриваемого сечения.

Вторая важная задача проектирования летательного аппарата – изучение его аэродинамических свойств. Решение этой задачи связано с исследованием процессов обтекания газом поверхностей произвольной формы. Для этого используются уравнения Навье – Стокса, которые в декартовой системе координат имеют вид:

$$\begin{aligned}
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} &= F_x - \frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right); \\
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} &= F_y - \frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right); \\
\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} &= F_z - \frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right);
\end{aligned} \tag{3.3}$$

где u , v , w – векторы скорости; F_x , F_y , F_z – проекции вектора силы на оси координат; ρ – плотность; p – давление; $\nu = \mu / \rho$, где μ – коэффициент вязкости.

Третья задача проектирования летательных аппаратов – расчет тепловых режимов работы деталей и узлов конструкции. Одним из основных аспектов задачи является определение температурных полей, имеющих место в конструкциях, связанное с вычислением количества теплоты, подводимой к телу или отводимой от него.

Температурное поле в сплошной среде описывается уравнением теплопроводности. Последнее может быть получено из уравнения (3.1), если под функцией φ понимать температуру T , а под коэффициентом K – коэффициент теплопроводности λ . Так, в двухмерном случае стационарное уравнение теплопроводности имеет вид:

$$\lambda_x \partial^2 T / \partial x^2 + \lambda_y \partial^2 T / \partial y^2 + Q = 0, \tag{3.4}$$

где Q – источник теплоты внутри тела, λ_x и λ_y – коэффициенты теплопроводности по соответствующим направлениям.

Сформулированные задачи являются типичными для многих областей техники. Так, задачу исследования механических напряжений, возникающих в конструкциях, необходимо решать при проектировании мостов, арок, опор электропередачи и т. д. Рост быстроходности и удельной мощности тепловых двигателей вызывает необходимость тщательного исследования проблем механической прочности и тепловых режимов работы их деталей. Аналогичные проблемы возникают в автомобиле- и турбостроении.

3.1.2 Краевые условия

Уравнения (3.1)–(3.4) имеют множество решений. Для получения единственного решения необходимо задавать краевые условия, а в случае нестационарных – значения этих же функций в начальный момент времени – *начальные условия*. Исходное дифференциальное уравнение в частных производных вместе

с краевыми условиями и представляет собой *математическую модель* исследуемого объекта.

На границе рассматриваемой области можно задать:

- 1) значение искомой функции;
- 2) значения производных по пространственным координатам от искомой функции;
- 3) уравнение баланса потоков.

Эти показатели называют *граничными условиями первого, второго и третьего рода* соответственно.

Для уравнений теплопроводности чаще задают граничные условия первого и третьего рода. Если на границе области имеет место конвективный теплообмен, то граничное условие третьего рода имеет вид:

$$\lambda_x \partial T / \partial x + \lambda_y \partial T / \partial y + \alpha(T - T_0) = 0, \quad (3.5)$$

где α – коэффициент теплообмена; T_0 – температура окружающей среды. Если на границе задан поток теплоты (q), то граничное условие будет выглядеть так:

$$\lambda_x \partial T / \partial x + \lambda_y \partial T / \partial y + q = 0, \quad (3.6)$$

где поток q считается положительным, если теплота отводится от рассматриваемого объекта.

Граничные условия для уравнений Навье – Стокса могут быть, например, первого рода, причем на границе необходимо задавать значения компонент вектора скорости, плотность и давление.

3.1.3 Приближенные модели задач микроуровня

Точное решение краевых задач удается получить лишь для немногих частных случаев. Поэтому общий способ их решения заключается в использовании различных приближенных моделей. Наиболее широкое распространение получили *модели на основе метода сеток*.

Применение метода сводится к аппроксимации искомой непрерывной функции совокупностью приближенных значений, рассчитанных в некоторых точках области – *узлах*. Совокупность узлов, соединенных определенным образом, образует *сетку*. Сетка и является *дискретной моделью области определения искомой функции*.

Применение метода сеток позволяет свести дифференциальную краевую задачу к системе нелинейных в общем случае алгебраических уравнений относительно неизвестных узловых значений функций.

В общем случае алгоритм метода сеток сводится к выполнению следующих этапов:

Этап 1. Дискретизация – замена непрерывных переменных x , y и z их конечным множеством в узлах выбранной расчетной сетки.

Этап 2. Алгебраизация – замена частных производных отношениями конечных разностей, в результате которой получается система алгебраических уравнений.

Этап 3. Решение полученной системы уравнений.

Метод сеток реализован в ряде специализированных пакетов прикладных программ, в которых все этапы метода выполняются автоматически. В системах автоматизированного проектирования (САПР) используются два метода сеток: *метод конечных элементов* и *метод конечных разностей*. Эти методы отличаются на этапах 1 и 2, а на этапе 3 они идентичны.

3.2 Метод конечных элементов

В настоящее время метод конечных элементов (МКЭ) является одним из наиболее популярных методов решения краевых задач в различных областях деятельности [2]. Начав развиваться как метод решения задач строительной механики, МКЭ быстро завоевал такие сферы инженерной деятельности, как проектирование самолетов и автомобилей, космических ракет, турбин и электродвигателей.

К *основным преимуществам МКЭ* относят доступность, простоту понимания и применимость для задач с произвольной формой области решения, возможность создания на основе метода МКЭ высококачественных компьютерных программ.

В МКЭ исходная область определения функции разбивается с помощью сетки, в общем случае неравномерной, на отдельные подобласти – *конечные элементы*. Искомая непрерывная функция аппроксимируется кусочно-непрерывной функцией, определенной на множестве конечных элементов. При аппроксимации чаще всего используются полиномы, которые подбираются так, чтобы обеспечить непрерывность искомой функции в узлах на границах элементов.

Пример использования МКЭ для расчета одномерного температурного поля в однородном стержне. Пусть имеется стержень длиной L и площадью поперечного сечения S (рис. 3.1). Один конец стержня жестко закреплен и к нему подводится тепловой поток q заданной интенсивности. На свободном конце

стержня происходит конвективный теплообмен с внешней средой. Известны коэффициент теплообмена α и температура окружающей среды T_0 . Вдоль боковой поверхности стержень теплоизолирован.

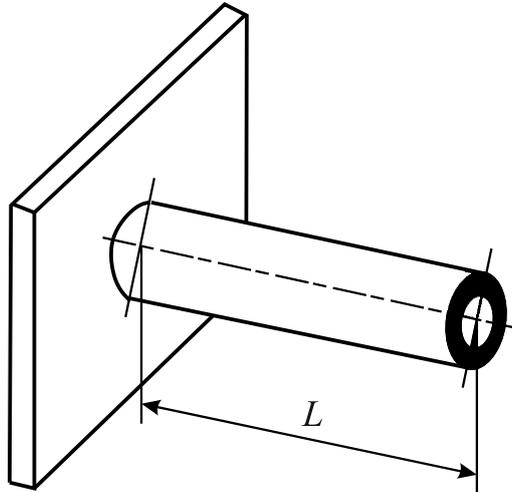


Рис. 3.1 – Однородный стержень

Температурное поле в одномерном стержне описывается уравнением теплопроводности, которое в одномерном приближении имеет вид:

$$\left(\lambda_x \partial^2 T / \partial x^2\right) = 0. \quad (3.7)$$

Краевые условия определяются уравнениями:

$$\begin{aligned} \lambda_x (\partial T / \partial x) + q, \text{ при } x = 0; \\ \lambda_x \partial T / \partial x + \alpha (T - T_0) = 0, \text{ при } x = L. \end{aligned} \quad (3.8)$$

Искомое температурное поле является *непрерывной функцией* координаты x . В МКЭ стержень разбивается произвольным образом на конечные элементы, которые в данном случае являются отрезками неравной длины. На каждом элементе непрерывная функция $T(x)$ аппроксимируется некоторой линейной зависимостью, как показано на рисунке 3.2 (в скобках указаны номера элементов). Аппроксимирующая кусочно-линейная функция определяется через узловые значения $T_1 - T_6$, которые в общем случае сначала неизвестны и подлежат определению в МКЭ.

Аналогичный подход может быть и в случае двух- и трехмерных областей искомой функции.

Для *двухмерных областей* наиболее часто используются элементы в форме треугольников и четырехугольников.

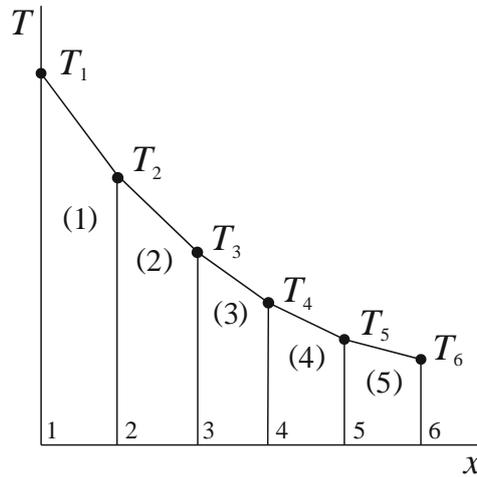


Рис. 3.2 – Расчет одномерного температурного поля

Для трехмерных областей наиболее употребимы элементы в форме тетраэдра или параллелепипеда, которые, также как и двумерные области, могут иметь прямолинейные и криволинейные границы.

В общем случае алгоритм МКЭ состоит из четырех этапов.

Этап 1. Выделение конечных элементов (разбиение заданной области на конечные элементы).

Этап 2. Определение аппроксимирующей функции для каждого элемента. На данном этапе значение непрерывной функции φ^e в произвольной точке его конечного элемента аппроксимируется полиномом

$$\varphi^e = A^e R + A_0, \quad (3.9)$$

где A^e – вектор-строка коэффициентов полинома; A_0 – свободный член; $R = (x, y, z)$ – вектор координат в рассматриваемой точке. Задача этапа далее заключается в определении вектора A^e и свободного члена A_0 . Для этого коэффициенты полинома выражают через вектор Φ^e узловых значений функции и координаты узлов

$$\varphi^e = N^e \Phi^e, \quad (3.10)$$

где N^e – матрица-строка, элементы которой называют функциями формы конечного элемента. Функции формы легко вычисляются в каждой точке конкретного элемента через координаты самой точки и координаты узлов элемента.

Этап 3. Объединение конечных элементов в ансамбль. На этом этапе уравнения (3.9), относящиеся к отдельным элементам, объединяются в ансамбль, т. е. в систему алгебраических уравнений:

$$\varphi = N\Phi. \quad (3.11)$$

Система (3.11) является моделью искомой непрерывной функции.

Этап 4. Определение вектора узловых значений функции. В общем случае вектор Φ в (3.11) вначале неизвестен. Его определение – наиболее сложная процедура в МКЭ, для нее строятся специальные алгоритмы.

Выделение конечных элементов. Разбиение области на элементы обычно начинают от ее границы с целью ее точной аппроксимации, затем производится разбиение внутренних областей. Сначала область разбивают на достаточно крупные подобласти, по границам изменения свойств материала, геометрии или приложенной нагрузки. Затем каждая подобласть разбивается на элементы. На рисунке 3.3 приведен пример разбиения двумерной области произвольной формы на треугольные конечные элементы с криволинейными границами.

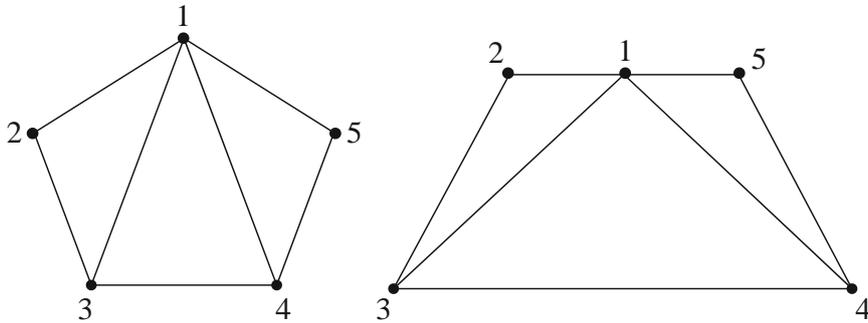


Рис. 3.3 – Пример разбиения на треугольные конечные элементы

Нумерация узлов элементов – следующая процедура этапа выделения конечных элементов. Матрица коэффициентов по методу МКЭ – сильно разреженная матрица ленточной структуры. Ненулевые элементы такой матрицы располагаются параллельно главной диагонали, и ширина полосы характеризуются числом L (наибольшая разность между номерами ненулевых элементов в строке).

Под числом степеней свободы понимают количество неизвестных функций, определяемых в каждом узле. Так, например, для 2-мерных задач гидравлики в каждом узле определяют три переменные: давление и составляющие скорости по осям X и Y .

При нумерации узлов предпочтителен способ, обеспечивающий минимальную разность между номерами узлов в каждом отдельном элементе. Если максимальную разность между номерами узлов для отдельного элемента обозначить N , а число степеней M , то ширина полосы

$$L = (N + 1)M.$$

При построении алгоритмов МКЭ стремятся к уменьшению числа L .

При описании области, разбитой на конечные элементы (КЭ), необходимо задавать:

- тип КЭ;
- порядковый номер КЭ;
- номера узлов и координаты узлов;
- информацию о соединении КЭ;
- значение физических параметров объекта в пределах КЭ.

Алгоритм подготовки топологической информации (препроцессор) состоит из трех этапов.

Этап 1. Нанесение на заданную область некоторого множества узлов.

Этап 2. Формирование узловых связей для заполнения области конечными элементами «наилучшей формы».

Этап 3. Нумерация узлов, минимизирующая ширину полосы L . В основу разбиения области произвольной формы на треугольные конечные элементы может быть положен следующий алгоритм:

- определение граничных узлов области;
- построение регулярной сетки с тем же числом узлов, что и в заданной области;
- использование полученной схемы соединения узлов для области произвольной формы.

На рисунке 3.3 приведен пример разбиения на треугольные конечные элементы области с пятью граничными узлами.

Определение аппроксимирующей функции элементов

Эту процедуру можно выполнить один раз для типичного элемента безотносительно к его топологическому положению в ней. Полученная функция используется для всех элементов области того же вида. Благодаря этой важной особенности МКЭ элементы с однажды определенными функциями легко включаются в библиотеку элементов программного комплекса и далее применяются для решения краевых задач.

Контрольные вопросы

1. С чем связана первая важная задача проектирования летательного аппарата?
2. В чем состоит вторая важная задача проектирования летательного аппарата?

3. Каким уравнением описывается температурное поле в сплошной среде?
4. Для каких задач проектирования летательного аппарата используются уравнения Навье – Стокса?
5. Какую модель реализует метод сеток?
6. Сформулируйте основные этапы метода МКЭ.

Литература

1. Кулон, Ж. Л. САПР в электротехнике / Ж. Л. Кулон, Ж. К. Сабонна-дьер. – М. : Мир, 1988. – 205 с.
2. Эндрюс, Дж. Математическое моделирование / Дж. Эндрюс, Мак-Лоун. – М. : Мир, 1979. – 277 с.

4 Теоретические основы метода компонентных цепей

4.1 Основы формализма метода компонентных цепей

Основы формализма метода компонентных цепей (МКЦ) были заложены в [1] и далее последовательно развивались в работах [2–4]. В тезисном выражении основные характеристики данного формализма можно представить в виде следующих составляющих:

- МКЦ – это объектно-ориентированный язык для моделирования сложных и физически неоднородных систем с энергетическими и информационными потоками в связях;
- компоненты могут иметь различную физическую природу (электроника, мехатроника, робототехника, автомобилестроение...);
- исследуемый объект представляется в форме компонентной цепи, модель которой строится из моделей независимых компонентов;
- модель компонента формируется с учетом четырех основных аспектов – топологического, физического, математического (логического) и геометрического. В результате автоматически строится система алгебро-дифференциальных (и/или логических) уравнений в обыкновенных или частных производных. Можно строить модели компонентов с логическими условиями и алгоритмическими блоками;
- для объектов с функционально обособленными подсистемами введено понятие структуры – подцепи, допускающей автономное решение. Здесь четко разделяются непрерывные (уравнения) и дискретные (алгоритмы) процессы;
- форма уравнений компонентной цепи и ее топологическая структура может меняться в зависимости от поведения переменных или наступления определенных событий.

4.2 Метод компонентных цепей как язык моделирования СТЭС

Рассмотрим неоднородные по своей физической природе СТЭС, исследование которых необходимо проводить на компонентном, макрокомпонентном и метауровнях, что требует разработки средств их формализованного представления для схемного моделирования. В качестве методической основы формализо-

ванного представления и моделирования неоднородных СТУС используется метод компонентных цепей, адаптированный В. М. Дмитриевым применительно к механическим и электромеханическим системам в работах [3, 4].

МКЦ относится к классу методов с полным координатным базисом. В отличие от табличного метода, ориентированного на моделирование радиоэлектронных схем, МКЦ допускает декомпозицию исходной системы на компоненты различной физической природы. Первым шагом к моделированию устройства является создание формализованного представления и машинно-ориентированного объекта-заместителя для реального элемента либо физэффекта, отражающего с требуемой степенью точности необходимые и достаточные аспекты элементов и явлений в СТУС. В методе КЦ таким объектом-заместителем объекта-оригинала является *компонент*, что и определило название метода [2].

Выделим в общем случае *четыре взаимосвязанных аспекта* рассмотрения каждого реального физического объекта (элемента реального устройства, прибора) либо явления (физического эффекта), которые должна отражать их модель на этапе функционального проектирования:

- *топологический аспект* – схемное изображение элемента либо физэффекта на принципиальной схеме, установление соответствия между связями и физическими переменными, действующими на связях;
- *физический аспект* – физические процессы, протекающие в элементе, степень их детализации; физические переменные (координатный базис), участвующие в описании физического процесса при различных степенях его детализации;
- *математический аспект* – математическое описание процессов в элементе и законов его функционирования;
- *графо-геометрический аспект* – пространственное изображение элемента, графическое отображение результатов.

Будем рассматривать аспект как совокупность признаков (предикатов): топологических, физических, математических и геометрических.

Центральным понятием МКЦ является понятие компонента как модели объективной реальности (элемента, физэффекта). С аспектами отдельного элемента и физэффекта связаны *четыре аспекта компонента* (рис. 4.1), а компонентные цепи исследуемого устройства наследуют все аспекты составляющих их компонентов.

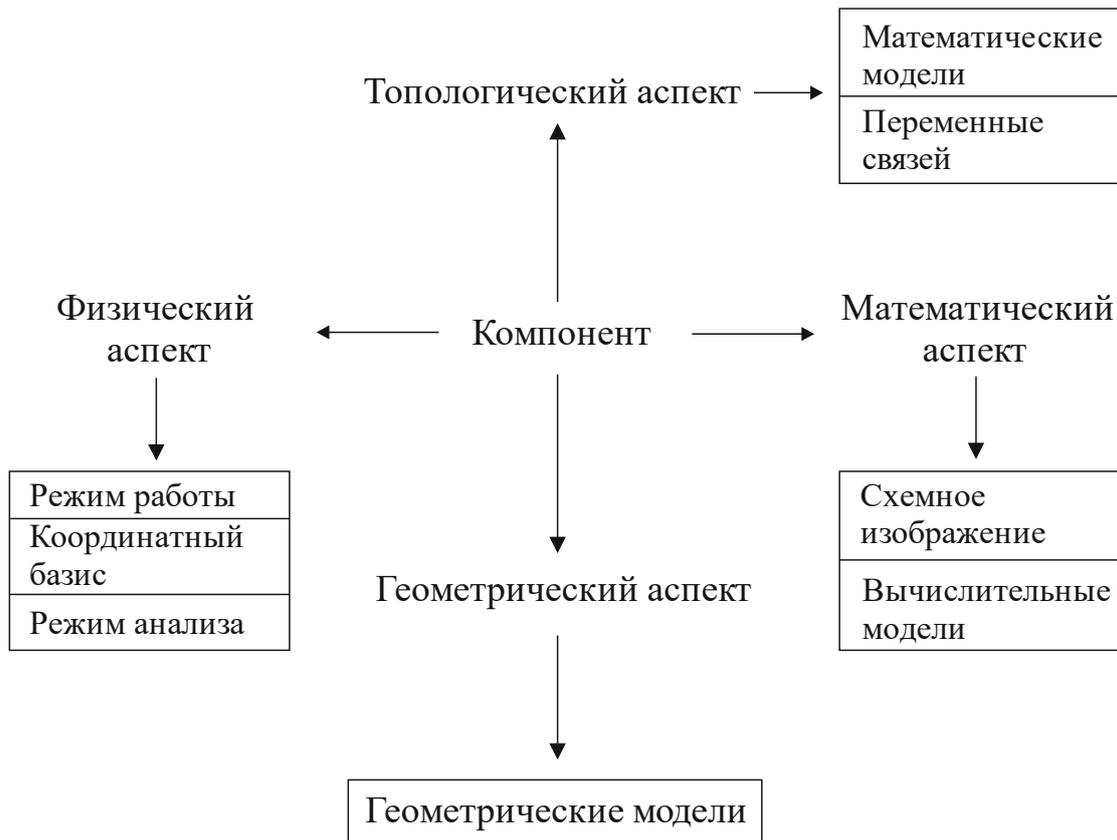


Рис. 4.1 – Аспекты компонента

Топологический аспект. Базовыми и наиболее общими категориями МКЦ являются *понятия компонента и компонентной цепи*. Данный метод ориентирован на моделирование элементов с сосредоточенными параметрами, а также для эквивалентного представления цепей и элементов с распределенными параметрами. Основной структурной сущностью МКЦ является компонент K с произвольным числом связей S_j , $j = \overline{1, k}$, где k – число связей (рис. 4.2, а), т. е. многополюсник. Связи компонента в общем случае являются *ориентированными*.

Компонентной цепью (КЦ) является произвольная совокупность компонентов, связи которых, именуемые ветвями цепи, объединены в общих точках, именуемых узлами цепи (рис. 4.2, б):

$$C_k = K \cap B \cap N, \quad (4.1)$$

где K – множество компонентов; N , B – множества узлов и связей (ветвей) КЦ. Во множестве компонентов K можно выделить следующие типы компонентов:

- K_s – компоненты-источники энергии или сигналов;
- K_z – компоненты-измерители;
- K_p – компоненты-преобразователи энергии или сигналов;
- K_u – управляющие компоненты.

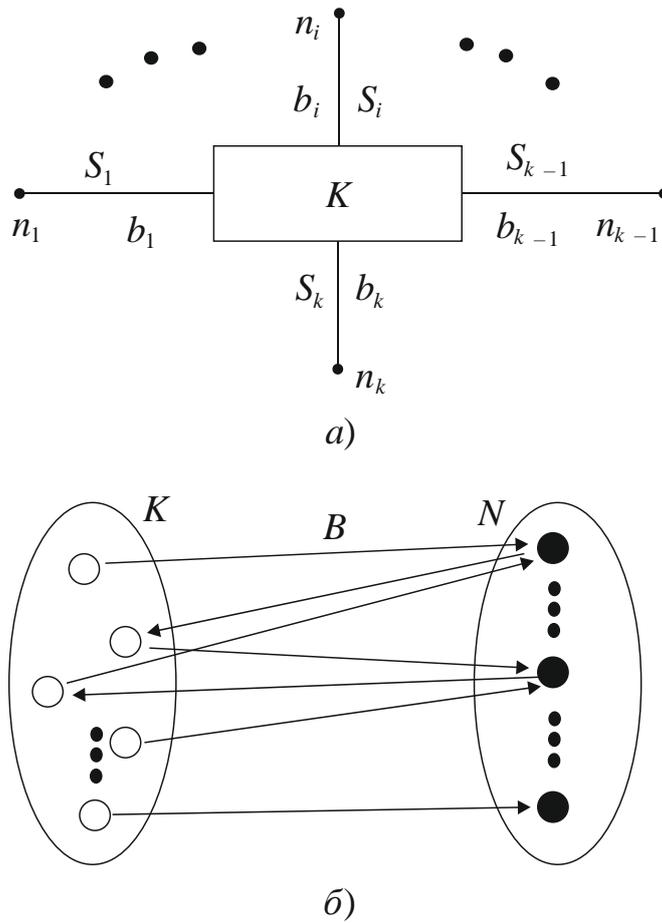


Рис. 4.2 – Компонент (а) и компонентная цепь (б)

Методом КЦ допускается использование, причем одновременное, *трех типов связей*:

- *связей энергетического типа S_e* , которым соответствует пара топологических координат (n_i, b_i) и пара дуальных переменных (V_{ni}, V_{bi}) ;
- *связей информационного типа S_i* , которым соответствует одна топологическая координата (n_i) и одна переменная связи (V_{ni}) , имеющая произвольный физический смысл, где n_i, b_i – номера узла и ветви соответственно; V_{ni}, V_{bi} – потенциальная и потоковая переменные;
- *связей векторного типа S_v* , которым соответствует вектор координат связи (n_i, b_i) и векторы потенциальных и потоковых переменных (V_{ni}, V_{bi}) .

Векторные связи удобно использовать в КЦ механических систем с многомерным (плоским или пространственным) движением при решении задач кинематики, статики и особенно динамики. Так, кинематический режим анализа ме-

ханических систем предполагает адаптацию моделей компонентов на этот режим, а связи интерпретируются как скалярные размерности 3. Переменными этих связей являются x -, y -, z -координаты точки. При динамическом режиме анализа связи тех же самых схемных элементов рассматриваются как векторные размерности 6. В процессе работы системы они автоматически расщепляются на элементарные, переменные связи которых суть x -, y -, z -координаты линейных и угловых скоростей, а также сил и моментов.

Замена нескольких связей одной векторной упрощает чертеж КЦ объекта на этапе его создания. При соответствующем выборе базового набора компонентов и их связей чертеж КЦ почти полностью (за исключением, быть может, компонентов-источников и компонентов-измерителей) идентичен естественному языку разработчика – языку принципиальных схем соответствующей предметной области.

По виду схемного изображения и количеству связей можно выделить односвязные, двухсвязные и многосвязные компоненты, по типу связей – компоненты с ориентированными, неориентированными, скалярными, элементарными и векторными связями.

При создании чертежа КЦ можно абстрагироваться от типа связей и узлов. Они однозначно определяются выбранным режимом анализа КЦ и связанными с ним математическими моделями компонентов.

В соответствии с введенной выше классификацией связей конкретизируем типы КЦ:

- $C_{\Gamma} = (K, S_{\Gamma}, N_{\Gamma})$ – графическая КЦ, представленная в графическом редакторе. Множества ее связей S_{Γ} и узлов N_{Γ} в общем случае могут включать векторные связи и узлы;
- $C_C = (K, S_C, N_C)$ – списковая КЦ со скалярными узлами и связями – преобразованная C_{Γ} -цепь, в которой векторные узлы и связи автоматически расщеплены на скалярные.

Списковая цепь C_C представляет собой информацию о каждом компоненте в виде

$$K_i (Z_1, \dots, Z_{pi}, B_1, \dots, B_{mi}, N_1, \dots, N_{ni}),$$

где K_i – имя компонента с pi внутренними параметрами и ni связями; Z_i – численные значения параметров компонента (физических, геометрических, конструктивно-технологических), являющихся атрибутами его формализованного

представления; B_1, \dots, B_{mi} – номера ветвей со знаком (ветвь, ориентированная к компоненту, отрицательна, а от компонента – положительна); N_1, \dots, N_{ni} – номера скалярных узлов (всегда положительны). Число ветвей mi у компонента может быть меньше числа его узлов ni .

В зависимости от типа компонента могут отсутствовать список параметров и список ветвей, тогда как множество узлов присутствует всегда. C_Γ -цепь в общем случае состоит из компонентов с различными типами связей. Но именно на базе C_C -цепи осуществляется автоматическое формирование модели КЦ.

Вручную задается только начало отсчета – нулевой узел. Пользователь обозначает узлом с номером $N = 0$ начало отсчета, в котором потенциальная переменная $V_0 = \text{const}$, чаще всего $V_0 = 0$.

Математический аспект. Наиболее актуальный при моделировании аспект – математический, связанный с представлением математической и вычислительной моделей компонента, а также формированием и решением вычислительной модели цепи. Следует обратить внимание, что глубокий системный анализ топологического и математического аспектов компонентов позволяет упростить формализованное представление объекта в форме КЦ за счет:

- выбора удобного топологического представления компонентов;
- автоматического формирования уравнений законов сохранения для потоковых переменных связей.

Кроме этого, МКЦ позволяет реализовать все способы формализованного представления, общепринятые при исследовании СТУС, – методы электромеханических аналогий или методов эквивалентирования, структурных графов или структурных схем.

При моделировании с каждым компонентом связано *два типа моделей* – математическая и вычислительная.

Математическая модель компонента, как и любая математическая модель, – это уравнение либо система уравнений относительно переменных связей компонента, являющиеся математическим описанием закона функционирования соответствующего элемента. Следует отметить, что изначально закон функционирования может быть представлен в различных видах – в аналитическом, таблично-графическом, в том числе в виде таблицы истинности. Для использования в МКЦ модель должна быть преобразована к аналитическому виду.

Вычислительная модель компонента – это его математическая модель, представленная в стандарте системы моделирования. Очевидно, что в разных системах моделирования, являющихся программной реализацией различных методов моделирования, формы представления одной и той же математической модели будут различны. В методе КЦ вычислительная модель каждого уравнения содержит информацию о его типе, переменных связях и соответствиях «ветвь – узел», коэффициентах при переменных и их производных, правых частях уравнений, т. е. сведения, необходимые и достаточные для формирования уравнения компонента в процессе формирования модели всей КЦ.

Базовой формой представления модели компонента является модель во временной области, это уравнение либо система уравнений относительно переменных его связей $\mathbf{V}_n^k, \mathbf{V}_b^k$, каждое из которых имеет вид:

$$f_{\text{лч}}(\mathbf{V}_n^k, \mathbf{V}_b^k, \mathbf{P}_k, t) = f_{\text{пч}}(\mathbf{V}_n^k, \mathbf{V}_b^k, \mathbf{P}_k, t), \quad (4.2)$$

где $f_{\text{лч}}, f_{\text{пч}}$ – левые и правые части уравнений модели; \mathbf{P}_k – множество параметров компонента; t – время, независимая переменная модели.

Математическая модель КЦ образуется объединением моделей компонентов вида (4.2) и уравнений топологических законов сохранения для потоковых переменных всех узлов КЦ за исключением базового.

Методом КЦ допускаются три типа уравнений относительно переменных связей компонента $\mathbf{V}^k = \mathbf{V}_n^k \cup \mathbf{V}_b^k$:

- линейные $\Phi(\mathbf{V}^k) = c$; (4.3)

- нелинейные $f(\mathbf{V}^k) = 0$; (4.4)

- дифференциальные $\frac{d\Psi(\mathbf{V}^k)}{dt} = f_1(\mathbf{V}^k, t)$; (4.5)

где $\Phi(\mathbf{V}^k) = \sum_{i=1}^{n+m} a_i V_i^k = a_1 V_{n1}^k + a_2 V_{n2}^k + \dots + a_{n+1} V_{b1}^k + \dots + a_{n+m} V_{bm}^k$ – линейная форма

относительно переменных связей \mathbf{V}^k ;

$$\frac{d\Psi(\mathbf{V}^k)}{dt} = \sum_{i=1}^{n+m} \psi_i \frac{dV_i^k}{dt} = \psi_1 \frac{dV_{n1}^k}{dt} + \psi_2 \frac{dV_{n2}^k}{dt} + \dots + \psi_{n+m} \frac{dV_{bm}^k}{dt} - \quad (4.6)$$

линейная форма относительно производных от переменных связей по времени;

$f(\mathbf{V}^k)$ – нелинейная функция; c – правая часть уравнения; $\psi_1, \psi_2, \dots, \psi_{n+m}$, –

коэффициенты при производных; n, m – количество узлов и ветвей компонента соответственно. В общем случае правая часть уравнения s и коэффициенты при производных $\psi_i, i = 1, 2, \dots, n + m$ являются функциями времени.

Уравнения вида (4.3)–(4.5) представляют собой необходимый и достаточный набор уравнений, которыми может быть описана математическая модель любого элемента с сосредоточенными параметрами, в том числе если она содержит производные высших порядков. Математические модели компонентов, не относящиеся к указанным классам, приводятся к каноническим формам путем введения дополнительных переменных и уравнений моделей на этапе разработки вычислительных моделей компонентов.

Геометрический аспект. На этапе создания КЦ компоненты соотносятся с соответствующим графическим образом. Геометрический аспект связан с интерпретацией полученных результатов моделирования и визуализацией графических образов, причем множество результатов моделирования, необходимых для визуализации, является субвектором переменных КЦ, использующегося при решении задач анализа функционирования ЭМС.

4.3. Методы решения модели

Автоматизированное моделирование на ЭВМ предполагает численное формирование и решение математических моделей устройств и систем, полученных различными методами при известных начальных условиях X_0 . Формирование машинной модели включает автоматическое выполнение следующих операций:

- *линеаризация* нелинейных уравнений – замена их линейными алгебраическими уравнениями в окрестности рабочих режимов;
- *алгебраизация* – замена линейных дифференциальных уравнений моделей линейными алгебраическими;
- *дискретизация* – замена непрерывных переменных конечным множеством их значений в заданных для исследования пространственном и временном интервалах.

При этом обязательными операциями является дискретизация и алгебраизация с целью получения системы алгебраических уравнений $F(X) = 0$, которая может быть решена непосредственно (методами простых итераций или релакса-

ции). Линеаризация нелинейных уравнений обеспечивает преобразование исходной системы уравнений в систему линейных алгебраических уравнений (СЛАУ) вида

$$AX=B, \quad (4.7)$$

где A – матрица системы, B – вектор свободных членов.

Решением СЛАУ является вектор фазовых переменных при дискретных значениях независимой переменной ξ (времени, частоты) на интервале анализа $\xi_i \in [\xi_0, \xi_k]$:

$$X = X_0, X_1, \dots, X_k = X(\xi_0), X(\xi_1), \dots, X(\xi_k). \quad (4.8)$$

Основной метод решения нелинейных алгебраических и трансцендентных уравнений вида $f(x) = 0$ в системах схемотехнического моделирования – линеаризация уравнения методом Ньютона:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad (4.9)$$

где x_n, x_{n+1} – значение переменной на n -й и $(n+1)$ -й итерациях.

Дифференциальные уравнения, представленные в нормальной форме Коши вида

$$\frac{dx(t)}{dt} = f(x, t) \quad (4.10)$$

алгебраизуются явными, неявными и комбинированными явно-неявными одношаговыми и многошаговыми методами интегрирования, которые подробно рассмотрены в работах [1, 2] и др.

Недостатком явных методов является ограничение на величину шага интегрирования, порядок которого для сохранения устойчивости вычислений не должен превышать порядка минимальной постоянной времени схемы. *Хорошими свойствами устойчивости* при решении жестких (плохо обусловленных) систем уравнений (для физических систем с большим разбросом постоянных времени) обладают *неявные разностные методы*, но они требуют решения системы алгебраических уравнений, в общем случае включающей нелинейные уравнения.

Решение СЛАУ осуществляется *прямыми методами* (Гаусса, Жордана, LU-разложения, правило Крамера) либо *итерационными* (методы простой итерации, Якоби, Гаусса – Зейделя, релаксационные). *Итерационные (приближенные) методы* находят применение в задачах сверхвысокой размерности с числом неизвестных от нескольких тысяч до миллионов. *Основными методами решения*

СЛАУ в системах схемотехнического моделирования являются метод Гаусса и его разновидность – метод LU-разложения [2].

В реальных задачах А-матрица СЛАУ имеет особенности (разреженность, специальный ленточно-диагональный и блочно-диагональный вид), учет которых (упорядочивание уравнений, хранения матриц в упакованном (без нулевых элементов) виде позволяет повысить быстродействие прямых методов.

Для расчета статических режимов используются: метод установления, прямой метод, метод движущейся области сходимости, метод оптимизации [1, 2].

В методе установления, или методе расчета статики через динамику, статический режим рассматривается как состояние схемы, к которому она стремится асимптотически при затухании переходных процессов, возникающих в схеме при подаче питающего напряжения. Данный подход не требует реализации специальных алгоритмов моделирования. Недостатком метода является значительное время расчета.

В прямом методе математическая модель схемы в статическом режиме вида $\Phi(\mathbf{X}) = 0$ получают из основной модели (4.6), полагая в ней $\dot{\mathbf{X}} = 0$. Решение осуществляется методом Ньютона.

Метод оптимизации основан на эквивалентности результатов решения математической модели прямым методом и минимизации какого-либо функционала, зависящего от $\Phi(\mathbf{X})$. Данный подход требует наличия в составе системы моделирования программ оптимизации. Метод оптимизации не нашел широкого применения при расчете статических режимов, так как в большинстве случаев он уступает методу Ньютона по быстродействию и не обеспечивает надежной сходимости.

Контрольные вопросы

1. К какому классу языков моделирования относится метод компонентных цепей?
2. Какой объект в МКЦ является основным?
3. Назовите 4 основных аспекта, на которых строится модель компонента.
4. Какие типы компонентов присутствуют в обобщенной компонентной цепи?
5. В чем состоят отличия списковой КЦ от графической КЦ?
6. Какие операции присутствуют при формировании машинной модели из исходной аналитической?

7. Какие методы интегрирования следует признать более устойчивыми – явные или неявные?

Литература

1. Дмитриев, В. М. MAPC – среда моделирования технических устройств и систем / В. М. Дмитриев, А. В. Шутенков, Т. Н. Зайченко, Т. В. Ганджа. – Томск : В-Спектр, 2011. – 278 с.
2. Дмитриев, В. М. Автоматизация моделирования промышленных роботов / В. М. Дмитриев, Л. А. Арайс, А. В. Шутенков. – М. : Машиностроение, 1995. – 304 с.
3. Дмитриев, В. М. Система визуализации и управления вычислительным экспериментом в среде многоуровневого моделирования MAPC / В. М. Дмитриев, Т. В. Ганджа, Т. Ю. Коротина // Доклады ТУСУРа. – 2010. – № 1–2(21) – С. 149–155.
4. Дмитриев, В. М. Среда многоуровневого компьютерного моделирования химико-технологических систем / В. М. Дмитриев, Т. В. Ганджа. – Томск : Изд-во Том. ун-та, 2017. – 332 с.

5 Методы формирования моделей для схемотехнического моделирования во временной области

5.1 Критерии выбора метода моделирования

Выбор аналитического или численного метода решения задачи определяется уровнем функциональной сложности проектируемого объекта. Теоретические основы моделирования СТЭС: теоретические основы электротехники, электроники, электромеханики и автоматики, теория электропривода и устройств преобразовательной техники, теория автоматического управления – находятся в постоянном развитии, а методы проектирования СТЭС развиваются и совершенствуются.

В общем случае математическая модель схемы при схемотехническом моделировании имеет вид системы обыкновенных дифференциальных уравнений:

$$\Phi(\mathbf{X}, \dot{\mathbf{X}}, t) = 0, \quad (5.1)$$

где \mathbf{X} – вектор фазовых переменных, достаточных для определения состояния моделируемого объекта, размерностью n ; $\dot{\mathbf{X}} = d\mathbf{X} / dt$ – вектор производных фазовых переменных по времени, имеющий $l \leq n$ ненулевых элементов; Φ – вектор-функция.

Способы формирования математических моделей определяются:

- топологическими законами сохранения;
- координатным базисом, т. е. тем, какие из фазовых переменных (переменных, характеризующих поведение объекта) выбраны в качестве базисных координат и входят в уравнения его математической модели;
- все остальные фазовые переменные рассчитываются через базисные.

От выбранного координатного базиса зависят простота алгоритма формирования модели объекта, размер модели (соответственно скорость, точность расчета и максимальная размерность рассчитываемой схемы), ограничения на допустимые в схеме типы зависимых источников. Каждый из методов формирования математической модели (табличный, узловых потенциалов, контурных токов, переменных состояния) имеет свои достоинства и недостатки.

Координатный базис *табличного метода* включает токи и напряжения ветвей, а также узловые потенциалы, отсчитанные от заземленного (базового) узла

электрической цепи. В работе [1] отмечается, что из табличного метода можно получить другие методы формирования системы уравнений модели. Так, если модель характеризовать только потенциалами узлов, отсчитанными от базисного, то получим модель, соответствующую методу узловых потенциалов. Если оставить в качестве фазовых переменных только токи хорд, то получим модель по методу контурных токов.

При составлении уравнений *модели по методу узловых потенциалов* цепь должна содержать только пассивные элементы и источники тока. Все источники напряжения должны быть преобразованы в источники тока по теореме об эквивалентном генераторе (*теорема Тевенена*). После решения системы уравнений модели дополнительно рассчитываются напряжения, действующие на пассивных элементах, напряжения на источниках и токи ветвей через напряжения ветвей и проводимости.

При моделировании *по методу контурных токов* все источники тока преобразуются в источники напряжения. Токи, протекающие через пассивные элементы, и токи, протекающие через источники напряжения, находятся после решения основной системы уравнений модели. В методах узловых потенциалов и контурных токов необходимо различать ветви с пассивными элементами и источниками. *Достоинством табличного метода* является отсутствие необходимости различать ветви с пассивными элементами и источниками, произвольная нумерация ветвей и матрицы инцидентий графа электрической модели. Однако из-за большой размерности системы уравнений метод не получил широкого распространения.

Основными машинно-ориентированными методами формирования моделей являются *модифицированные методы: узловых потенциалов и табличный*. Метод контурных токов удобен для ручного расчета простых цепей. При реализации программ компьютерного моделирования он не нашел широкого применения из-за сложности алгоритмов для проверки планарности цепи и автоматического выбора контуров.

5.2 Метод узловых потенциалов

Метод узловых потенциалов характеризуется моделью, где в общем виде для цепи, имеющей $n + 1$ узел, эти уравнения имеют вид:

$$\begin{bmatrix} g_{11} & g_{12} & L & g_{1n} \\ g_{21} & g_{22} & L & g_{2n} \\ M & M & L & M \\ g_{n1} & g_{n2} & L & g_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ M \\ V_n \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \\ M \\ J_n \end{bmatrix}.$$

В более компактном виде:

$$[G][V]=[J]. \quad (5.2)$$

Здесь $[V]$ – вектор узловых напряжений. Квадратную матрицу коэффициентов $[G]$ называют *матрицей узловых проводимостей*, а вектор правых частей – *вектором узловых токов*.

Оптимизация метода узловых потенциалов и табличного метода связана с исключением из координатного базиса переменных, которые могут быть определены через фазовые переменные. В модифицированном методе узловых потенциалов, использующем одиночный граф, фазовыми переменными являются узловые потенциалы и токи через элементы, которые невозможно описать с помощью проводимостей, а также токи источников напряжения и токи ветвей, которые необходимо определить из решения.

5.3 Модифицированный метод узловых потенциалов

Расширенные узловые уравнения имеют форму:

$$\begin{bmatrix} [Y] & [M] \\ [N] & [Z] \end{bmatrix} \begin{bmatrix} [V] \\ [I] \end{bmatrix} = \begin{bmatrix} [J] \\ [E] \end{bmatrix}. \quad (5.3)$$

Матрица коэффициентов в системе уравнений (5.3) разбита на четыре субматрицы. Субматрица $[Y]$ размера $(n_y - 1)(n_y - 1)$ является *матрицей узловых проводимостей* регулярной части цепи. Субматрица $[M]$ содержит коэффициенты компонентных уравнений, матрица $[N]$ учитывает токи нерегулярных элементов в уравнениях по первому закону Кирхгофа, а матрица $[Z]$ является матрицей сопротивлений регулярной части цепи.

5.4 Метод КЦ в матричной форме

Методом КЦ предусматривается построение функциональных моделей КЦ во временной и в частотной областях для линейных непрерывных схем.

Модель КЦ со связями энергетического типа во временной области. Существует матрично-топологическая интерпретация МКЦ, которая предполагает

предварительную алгебраизацию и линеаризацию уравнений моделей компонентов [2].

Построение математической модели КЦ заключается в объединении моделей компонентов и узловых топологических законов сохранения – первого закона Кирхгофа (закон равенства нулю суммы потоковых переменных связей, ведущих в один и тот же узел).

Матричная модель компонента во временной области имеет вид:

$$[\mathbf{A}^k] \cdot \begin{bmatrix} [\mathbf{V}_n^k] \\ [\mathbf{V}_b^k] \end{bmatrix} = [\mathbf{W}^k], \quad (5.4)$$

где $[\mathbf{A}^k]$, $[\mathbf{W}^k]$ – компонентная матрица и вектор правых частей матричной модели компонента с учетом линеаризации нелинейных и алгебраизации и линеаризации дифференциальных уравнений модели компонента.

Модель КЦ имеет вид:

$$\begin{cases} V_{n0} = 0; \\ \begin{bmatrix} [\mathbf{A}] \\ 0 | [\mathbf{T}] \end{bmatrix} \cdot \begin{bmatrix} [\mathbf{V}_N] \\ [\mathbf{V}_B] \end{bmatrix} = \begin{bmatrix} [\mathbf{W}] \\ 0 \end{bmatrix}, \end{cases} \quad (5.5)$$

где $V_{n0} = 0$ уравнение базового узла n_0 ; $[\mathbf{T}]$ – матрица инциденций; $[\mathbf{V}_N]$, $[\mathbf{V}_B]$ – векторы потенциальных и потоковых переменных КЦ; $[\mathbf{A}] = \begin{bmatrix} [\mathbf{A}^1] \\ [\mathbf{A}^2] \\ \dots \\ [\mathbf{A}^q] \end{bmatrix}^T$ – общая компонентная матрица; $[\mathbf{W}] = \begin{bmatrix} [\mathbf{W}^1] \\ [\mathbf{W}^2] \\ \dots \\ [\mathbf{W}^q] \end{bmatrix}^T$ – вектор правых частей; $[\mathbf{A}^j]$, $[\mathbf{W}^j]$, $j = \overline{1, q}$ – компонентные матрицы и векторы правых частей компонентов; q – количество компонентов КЦ.

Решением модели КЦ является вектор ее переменных связей (потенциальных и потоковых переменных) $[\mathbf{V}] = \begin{bmatrix} [\mathbf{V}_N] \\ [\mathbf{V}_B] \end{bmatrix}^T$.

Модель КЦ со связями информационного типа во временной области. В КЦ со связями информационного типа отсутствуют потоковые переменные и $[\mathbf{V}] = [\mathbf{V}_N]$. Поэтому модель КЦ имеет вид:

$$[\mathbf{A}] \cdot [\mathbf{V}_N] = [\mathbf{W}] \quad (5.6)$$

Для решения моделей КЦ (5.4) и (5.5) применяются программы решения линейных алгебраических уравнений. *Решением модели КЦ* является вектор ее переменных связей $[\mathbf{V}] = \left[[\mathbf{V}_N] [\mathbf{V}_B] \right]^T$.

Модель КЦ в частотной области. Формирование модели непрерывной линейной КЦ в частотной области базируется на традиционном переходе к комплексной форме представления. Модель КЦ формируется в полном координатном базисе действительных и мнимых составляющих потенциальных и потоковых переменных связей КЦ. Размерности всех матриц в моделях (5.4), (5.5) и вектора решения КЦ удваиваются. Таким образом, модель КЦ со связями энергетического типа в частотной области имеет вид:

$$\begin{bmatrix} [\mathbf{A}'_{MN}] & 0 & 0 & [\mathbf{A}'_{MB}] \\ 0 & [\mathbf{A}'_{MB}] & [\mathbf{A}''_{MN}] & 0 \\ 0 & [\mathbf{T}] & 0 & 0 \\ 0 & 0 & 0 & [\mathbf{T}] \end{bmatrix} \cdot \begin{bmatrix} [\mathbf{V}'_N] \\ [\mathbf{V}'_B] \\ [\mathbf{V}''_N] \\ [\mathbf{V}''_B] \end{bmatrix} = \begin{bmatrix} [\mathbf{W}'_M] \\ [\mathbf{W}''_M] \\ 0 \\ 0 \end{bmatrix},$$

а со связями информационного типа:

$$\begin{bmatrix} [\mathbf{A}'_{MN}] & 0 \\ 0 & [\mathbf{A}''_{MN}] \end{bmatrix} \cdot \begin{bmatrix} [\mathbf{V}'_N] \\ [\mathbf{V}''_N] \end{bmatrix} = \begin{bmatrix} [\mathbf{W}'_M] \\ [\mathbf{W}''_M] \end{bmatrix},$$

где $[\mathbf{V}'_N], [\mathbf{V}'_B]$ – действительные составляющие потенциальных и потоковых переменных связей КЦ; $[\mathbf{V}''_N], [\mathbf{V}''_B]$ – мнимые составляющие потенциальных и потоковых переменных связей; $[\mathbf{A}'_{MN}], [\mathbf{A}''_{MN}], [\mathbf{A}'_{MB}], [\mathbf{A}''_{MB}]$ – компонентные матрицы моделей уравнений, записанных относительно действительных и мнимых составляющих потенциальных и потоковых переменных соответственно; $[\mathbf{W}'_M], [\mathbf{W}''_M]$ – матрицы векторов правых частей уравнений моделей компонентов.

Решением модели КЦ является совокупность действительных и мнимых составляющих потенциальных и потоковых переменных:

$$[\mathbf{V}] = \left[[\mathbf{V}'_N] \quad [\mathbf{V}'_B] \quad [\mathbf{V}''_N] \quad [\mathbf{V}''_B] \right]^T.$$

После автоматической алгебраизации и линеаризации дифференциальных и нелинейных модельных уравнений модель КЦ в общем случае имеет вид:

$$[\Phi][\mathbf{V}] = [\mathbf{W}], \quad (5.7)$$

где $[\Phi]$ – матрица коэффициентов; $[\mathbf{V}]$ – вектор-столбец решения КЦ; $[\mathbf{W}]$ – вектор-столбец правых частей.

Решения линеаризованной модели КЦ (5.7) осуществляются программами решения линейных алгебраических уравнений.

5.5 Метод переменных состояния

Метод переменных состояния предназначен для получения математической модели схемы в виде системы обыкновенных дифференциальных уравнений в нормальной форме Коши.

Уравнения переменных состояния удобно записывать в матричной форме:

$$[X(t)] = [A][X(t)] + [B][U(t)]. \quad (5.8)$$

Здесь $[X]$ – вектор переменных состояния, $[\dot{X}]$ – вектор производных, $[U(t)]$ – вектор входных воздействий.

Квадратную матрицу $[A]$ называют *матрицей параметров схемы*, а $[B]$ – *матрицей параметров входного воздействия*.

Такая форма предпочтительна при применении явных методов численного интегрирования, поэтому метод переменных состояния был наиболее распространен в программах анализа первого поколения, где преобладало использование явного интегрирования. За вектор переменных состояния могут быть приняты узловые потенциалы, контурные токи либо другие переменные. При моделировании переходных процессов в радиоэлектронных устройствах за переменные состояния рекомендуется выбирать напряжения на всех конденсаторах и токи всех индуктивностей.

Основные недостатки метода переменных состояния – сложность формирования математической модели и ориентация на явные методы решения обыкновенных дифференциальных уравнений, неустойчивость и неэффективность при расчете схем с большим разбросом постоянных времени из-за ограничений на шаг расчета.

В работе И. Влаха и К. Сингхала [3] анализируются методы автоматического формирования уравнений и даются рекомендации по их применению. *Критерием оценки* является размерность матрицы матричного уравнения модели цепи и плотность их заполнения (отношение числа ненулевых элементов к общему числу элементов матрицы). Отмечается, что матрицы, полученные табличным методом, велики даже для небольших цепей. Они очень разрежены и требуют использования методов решения, учитывающих разреженность. *Модифицированный узловой метод* приводит к более компактной матрице и не требует

учета разреженности. Для задач, связанных с машинным расчетом, рекомендуются узловые методы.

Вместе с тем в работе [4] отмечается, что реализованные ими программы (ПА4, ПА6, ПА7, ПА8, ПА9, разработанные на основе методов переменных состояния, узлового и модифицированного узлового, имеют ограничения в части реализации моделей отдельных элементов и функциональных зависимостей. Поэтому ими вводится в рассмотрение обобщенный координатный базис и набор базовых элементов. *Обобщенный координатный базис включает* напряжения и токи на всех двухполюсных элементах, потенциалы всех узлов, кроме базового, производные от напряжений на емкостях и производные от токов на индуктивностях. *Набор базовых элементов состоит* из базовых двухполюсников (зависимые источники тока и напряжения, емкость, сопротивление, индуктивность, проводимость) и функционального многополюсника, с помощью которого становится возможным получать модели сложных элементов технических объектов. Данный подход находится на стадии разработки и планируется для реализации в программе моделирования ПА10. Пока самым эффективным оказался метод компонентных цепей [5].

Контрольные вопросы

1. Чем определяется выбор аналитического и численного методов решения задачи?
2. Какие компоненты может содержать цепь при составлении уравнений модели по методу узловых потенциалов?
3. Какие компоненты может содержать цепь при составлении уравнений модели по методу контурных токов?
4. Для чего предназначен метод переменных состояния?
5. Какой смысл имеют матрицы метода переменных состояния?

Литература

1. САПР: Системы автоматизированного проектирования : учеб. пособие для техн. вузов / под ред. И. П. Норенкова. – М. : Высш. шк., 1988. – 143 с.
2. Левицкий, В. П. Обзор основных программных средств для моделирования математических задач / В. П. Левицкий, Н. В. Колодницкий // САПР и графика. – 1999. – № 10. – С. 56–65.
3. Влах, И. Машинные методы анализа и проектирования электронных схем / И. Влах, К. Синхгал. – М. : Радио и связь, 1988. – 560 с.

4. Норенков, И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры / И. П. Норенков, В. Б. Маничев. – М. : Высш. шк., 1983. – 272 с.
5. Дмитриев, В. М. МАРС – среда моделирования технических устройств и систем / В. М. Дмитриев, А. В. Шутенков, Т. Н. Зайченко, Т. В. Ганджа. – Томск : В-Спектр, 2011. – 278 с.

6 Операторно-структурные схемы и графы систем

При решении задач анализа и синтеза систем широко используют структурные отображения математической модели – операторно-структурные схемы (ОСС), описывающие процессы в этих системах. Это позволяет более наглядно представлять и определять взаимосвязи между входными, внутренними и выходными переменными систем, решать задачи их анализа и модификации.

6.1 ОСС линейных стационарных непрерывных систем

В основе графических изображений, описывающих системы математических моделей, лежит понятие оператора и операторного изображения. Напомним, что *оператором* в математике принято называть математический символ, отражающий преобразование одного множества переменных во множество других переменных. Например, взаимосвязь

$$y(p) = W(p)x(p) \quad (6.1)$$

изображения $y(p)$ выходного сигнала $y(t)$ одномерной по входу-выходу линейной системы с изображением $x(p)$ входного сигнала $x(t)$ этой системы через оператор – передаточную функцию $W(p)$. Это изображение можно представить графически в виде схемы (рис. 6.1):

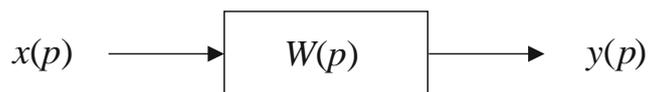


Рис. 6.1 – Графическое представление изображения

Операторно-структурные схемы линейных стационарных непрерывных систем (ЛСН-систем) состоят из следующих *типовых элементов*: звеньев, сумматоров, узлов и линий связи.

Каждое *звено* в ОСС отображается прямоугольником с направленными (причинно-следственными) *линиями связи* (связями) с указанием входных, внутренних и выходных переменных (сигналов). *Известные примеры звеньев*: интегрирующее, дифференцирующее, апериодическое, колебательное и т. д.

Сумматоры в ОСС отображаются так, как представлено на рисунке 6.3 (кружок, разделенный на секторы). Подводимые сигналы отражаются линиями со стрелками к сумматору, а выходные – линиями со стрелками от сумматора.

Зачернением сектора сумматора учитывается знак подводимого к сумматору сигнала. Плюс (+) – светлый сегмент, минус (–) – темный сегмент входа стрелки обратной связи (ОС).

Модель сумматора отражает алгебраическую сумму вида

$$y = \sum_i \lambda_i x_i, \quad (6.2)$$

где $\lambda_i = 1$ или -1 для входящих и выходящих связей соответственно.

Имеется модель и многовходового (больше двух) сумматора.

Узлы в ОСС изображаются точками на линиях связей согласно рисунку 6.2 и используются для отражения разветвлений для передачи одних и тех же сигналов. При этом *сигналы*, подходящие и отходящие от узла, равны между собой (в отличие от токов в узлах электрических цепей).



Рис. 6.2 – Узел с линиями связи

С учетом введенных выше понятий основных элементов ОСС графическим отображением операторного уравнения

$$y = W_1(p)x_1 - W_2(p)x_2 \quad (6.3)$$

будет ОСС, представленная на рисунке 6.3.

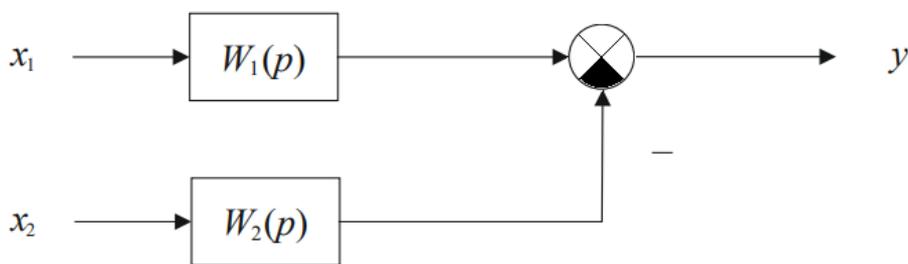


Рис. 6.3 – Графическое отображение операторного уравнения

Отображение математического уравнения в виде ОСС не всегда является однозначным. В частности, даже для такого простого уравнения как

$$A(p)y(p) = B(p)x(p) \quad (6.4)$$

может быть поставлено в соответствие несколько ОСС. Две из них представлены на рисунке 6.4.

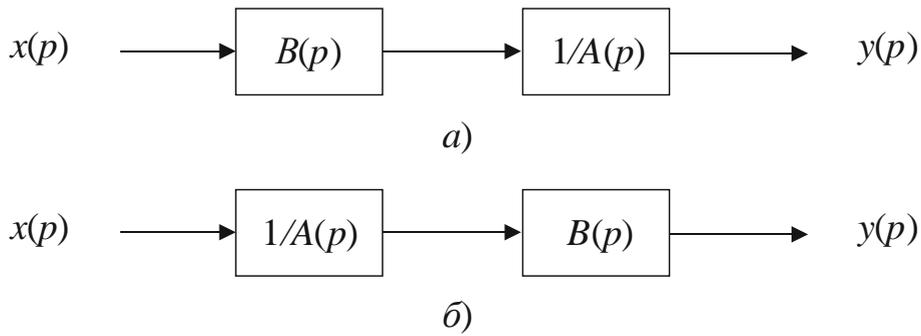


Рис. 6.4 – Варианты ОСС для одного уравнения

Данный факт объясняется тем, что *операторные уравнения преобразуются по правилам алгебры и могут быть записаны разным образом.*

Общая схема алгоритма для представления математических моделей динамики ЛСН-системы содержит следующие действия:

- 1) перевести в операторную форму исходную модель системы, записанную в форме интегро-дифференциальных и алгебраических уравнений, применяя преобразование по Лапласу;
- 2) построить ОСС для каждого из полученных уравнений модели с учетом причинно-следственных связей у звеньев;
- 3) линии, соответствующие одним и тем же переменным системы в схемах, соответствующих отдельным уравнениям, соединяют в узлы. Тем самым фиксируется взаимосвязь отдельных уравнений математической модели системы.

Пример 1. В качестве примера рассмотрим и построим ООС для следующей упрощенной математической модели электродвигателя постоянного тока с независимым возбуждением:

$$\begin{aligned}
 J \frac{dw}{dt} &= M_d(t) - M_c(t) \\
 u(t) &= IR + L \frac{di}{dt} + E(t), \\
 M_d(t) &= M_c(t) \\
 E(t) &= c_e w(t)
 \end{aligned}
 \tag{6.5}$$

где w – угловая скорость вращения ротора; J – приведенный к валу двигателя момент инерции; M_d – электромагнитный момент двигателя; M_c – момент сопротивления на его валу.

Запишем эти уравнения в операторной форме:

$$\begin{aligned} Jpw(p) - Jw(0) &= M_d(p) - M_c(p) \\ u(p) &= I(p)R + LpI(p) - LI(0) + E(p) \\ M_d(p) &= M_c(p) \\ E(p) &= c_e w(p) \end{aligned} \quad (6.6)$$

Первые два из этих уравнений перепишем в иной форме, где явно выражены выходные переменные и получим окончательно:

$$\begin{aligned} w(p) &= \frac{1}{Jp} [Jw(0) + M_d(p) - M_c(p)] \\ I(p) &= \frac{1}{R + Lp} [u(p) + LI(0) - E(p)] \\ M_d(p) &= M_c(p) \\ E(p) &= c_e w(p) \end{aligned} \quad (6.7)$$

На основе полученных уравнений построим ОСС (рис. 6.5). В ней в качестве выходного сигнала рассматривается скорость вращения w , хотя в качестве выходной может быть выбрана и любая другая. Отметим также, что часто на практике начальные условия в системе нулевые.

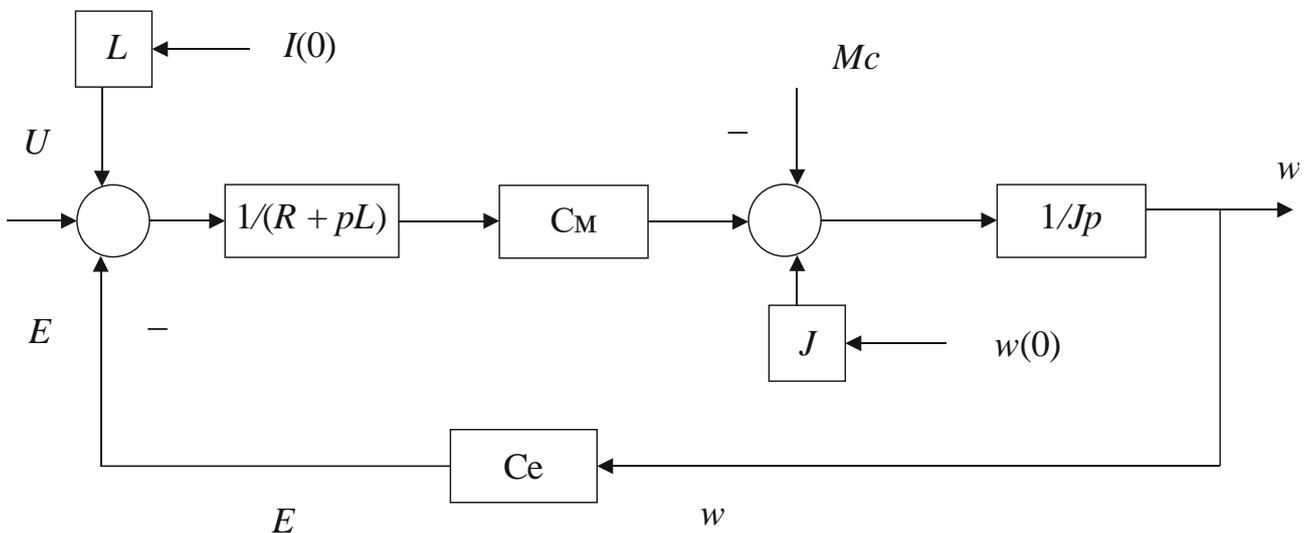


Рис. 6.5 – ОСС для упрощенной математической модели электродвигателя постоянного тока

6.2 Правила преобразований ОСС линейных систем

Преобразования ОСС производятся с целью их агрегирования, трансформации или декомпозиции.

Агрегирование ОСС сводится к замене соединений двух и большего числа звеньев в одно более сложное, но эквивалентное им по вход-выходным характеристикам. Это позволяет упростить ОСС системы и находить передаточные функции (ПФ) между интересующими сигналами.

Трансформация ОСС обычно осуществляется для более удобного восприятия или упрощения схемы. При этом число входящих в ОСС звеньев и сумматоров может и не уменьшаться. Простейшим видом трансформации ОСС может быть другое расположение элементов, но с сохранением всех существующих связей.

Декомпозиция ОСС преследует противоположные агрегированию цели и сводится к замене звеньев со сложными ПФ эквивалентными им по математическому описанию соединениями из звеньев с более простыми ПФ.

Правила агрегирования в ОСС формулируются для типовых соединений звеньев – последовательного (рис. 6.6, а), параллельного (рис. 6.6, б) и соединения с обратной связью (ОС) (рис. 6.6, в).

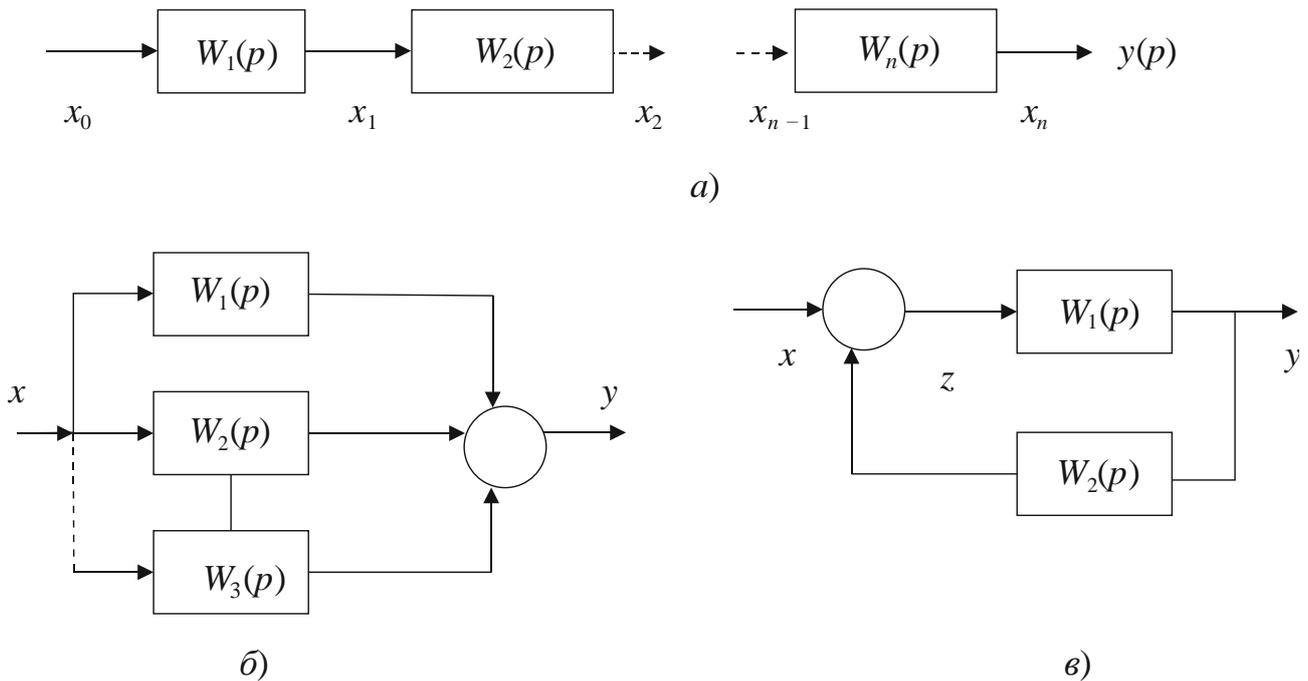


Рис. 6.6 – Правила агрегирования в ОСС

На рис 6.6, а представлено последовательное соединение звеньев. ПФ звена, эквивалентного этому соединению, находится как произведение ПФ всех входящих в данное соединение звеньев, т. е.

$$y = W_1(p) W_2(p) \dots W_n(p) x. \quad (6.8)$$

При параллельном соединении звеньев ПФ эквивалентного этому соединению звена вычисляется как алгебраическая сумма ПФ звеньев, входящих в это соединение:

$$W_{\text{экв}}(p) = \sum_{i=1}^n W_i(p). \quad (6.9)$$

Знак ПФ $W_i(p)$ каждого звена при алгебраическом суммировании в (6.9) определяется с тем знаком, с которым поступает на сумматор выходной сигнал с данного звена.

Для соединения с ОС (рис. 6.6, в) ПФ эквивалентного звена:

$$W_{\text{экв}}(p) = \frac{W_1(p)}{1 \pm W_1(p)W_2(p)}. \quad (6.10)$$

Знак «+» здесь соответствует отрицательной ОС, а знак «-» ставится, когда эта связь положительная.

Наряду с этими правилами при преобразовании операторно-структурных схем используются также правила, обеспечивающие трансформацию ОСС [1]. Сюда прежде всего относятся операции переноса сумматоров и узлов через звенья.

Перенос сумматора через звено проиллюстрирован на рисунке 6.7, а, б.

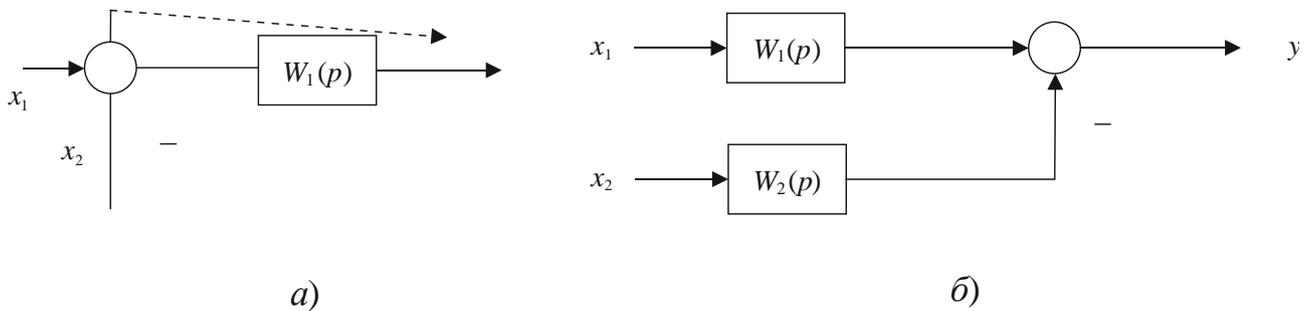


Рис. 6.7 – Перенос сумматора через звено

7 Операторно-структурные схемы линейных многомерных систем

7.1 Операторно-структурные схемы линейных стационарных систем

Для линейных стационарных и многомерных по входу-выходу систем их описание имеет вид типовой математической модели «вход – состояние – выход»:

$$\begin{aligned} \sigma x(t) &= Ax(t) + Bu(t) + Ef(t) \\ y(t) &= Cx(t) + Du(t) + Gf(t) \end{aligned} \quad (7.1)$$

Ее запись в операторной форме при нулевых начальных условиях и с использованием непрерывного преобразования Лапласа для непрерывных систем или же с использованием z -преобразования для дискретных по t систем имеет вид:

$$\begin{aligned} (qI - A)x(q) &= Bu(q) + Ef(q) \\ y(q) &= Cx(q) + Gf(q) \end{aligned}, \quad (7.2)$$

где q означает оператор *непрерывного* преобразования p в случае непрерывных систем или оператор z -преобразования для *дискретных* по t систем, а I – единичная диагональная матрица той же размерности, что и вектор x системы.

Структурный аналог модели приведен на рисунке 7.1. Такой класс структурных отображений называют *структурными схемами многомерных систем*, представленных в векторно-матричной форме. Далее будем называть их *операторными структурными схемами векторно-матричного типа*.

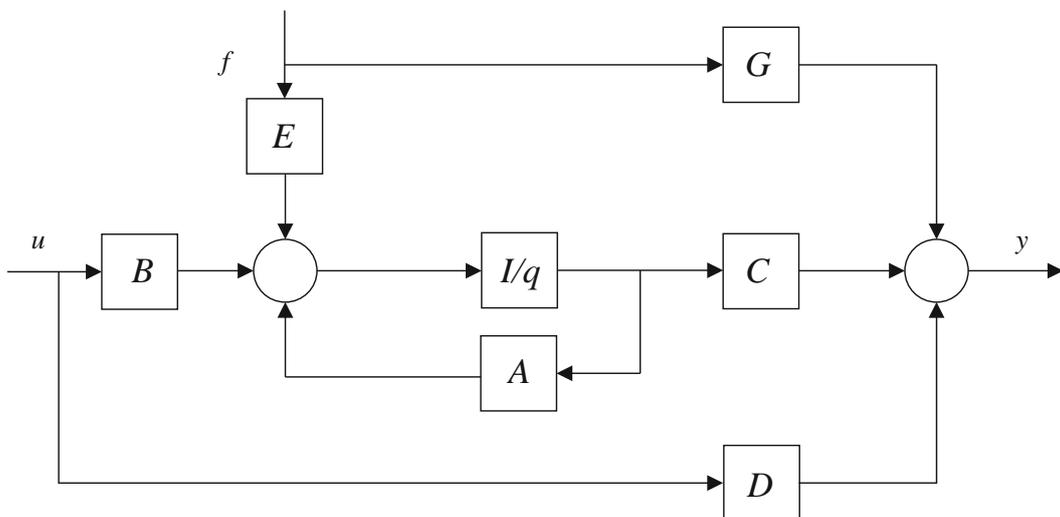


Рис. 7.1 – Структурный аналог модели

В общем случае все связи входящих в схему элементов являются *многомерными*, то есть по ним циркулируют векторные переменные и лишь в частных случаях – скалярные. Каждое из звеньев такой схемы характеризуется передаточной матрицей – A, B, C, D, E, G исходных математических моделей систем. Передаточная матрица охваченного ОС звена на рисунке 7.1 имеет диагональный вид и соответствует объединению n независимых вход-выходных интеграторов.

При преобразованиях операторно-структурных схем векторно-матричного типа используются правила того же типа, что и для обычных ОСС со скалярными переменными (сигналами). Фактически по тем же правилам производятся переносы сумматоров и узлов через звенья и относительно друг друга, объединение параллельно включенных звеньев, только вместо ПФ в таких случаях используются передаточные матрицы (ПМ) звеньев. При последовательной свертке звеньев порядок записи ПМ звеньев в ПМ эквивалентного этому соединению звена должен быть строго фиксированным. В частности, если использовать для такого соединения те же, что и ранее, но заменить оператор p на обобщающий непрерывные и дискретные системы оператор, тогда следует принимать

$$W_{\text{экв}} = W_n(q) W_{n-1}(q) \dots W_2(q) W_1(q), \quad (7.3)$$

то есть записывать в ПМ эквивалентного звена ПМ звеньев в обратном порядке от конца их соединения к началу, что обеспечивает соблюдение допустимых соотношений между размерностями ПМ и векторов переменных в связях звеньев.

Для соединения с ОС в ОСС векторно-матричного типа, соответствующего рисунку 7.1, ПМ, эквивалентного такому соединению звена, определяется как на рисунке 7.2.

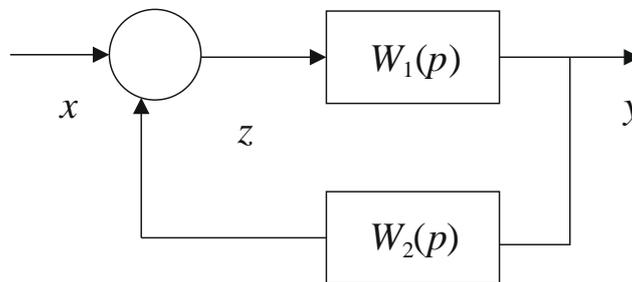


Рис. 7.2 – Звено с ОС

$$W_{\text{экв}} = [I \pm W_1(q) W_2(q)]^{-1} W_1(q). \quad (7.4)$$

При скалярном характере переменных связей соотношение (7.4) сводится к формуле (6.10) из лекции 6.

7.2 Операторно-структурные схемы линейных нестационарных непрерывных систем

Математическая модель (ММ) таких многомерных по входу-выходу систем может быть представлена в виде

$$A(p, t)y(t) = B(p, t)x(t), \quad (7.5)$$

где $x \in R^k$; $y \in R^i$ – соответственно вход-выходные множества, а $A(p, t)$ и $B(p, t)$ – полиномиальные относительно символа p матрицы. Коэффициенты входящих в эти матрицы полиномов меняются с течением времени t .

Графическое изображение математической модели типа (7.5) по сравнению с моделями линейных стационарных непрерывных систем требует дополнительных определений и интерпретаций. Это обусловлено тем, что преобразование уравнений в операторную форму связано с преобразованием сверток функций (*мультипликативных составляющих*) типа $c(t) \frac{d^r z}{dt^r}$, которые имеют сложные формулы по правилам непрерывного преобразования Лапласа. Это приводит к крайне сложным структурным изображениям вида (7.5) даже для одномерных систем.

По этой причине при анализе и синтезе систем рассматриваемого класса ОСС используют, как правило, лишь в простых случаях, когда математическая модель системы имеет сравнительно малое число параметров. При этом исходная математическая модель преобразуется по Лапласу в совокупность операторных уравнений без учета нестационарности параметров. Далее по полученным уравнениям строится ОСС и в ней в соответствующих ПФ звеньев указываются зависимости от времени этих параметров.

7.3 Операторно-структурные схемы и графы нелинейных систем

К данному классу систем относят такие, в которые входят типовые нелинейности (релейные и гистерезисные характеристики). Здесь *отличие состоит* в том, что в итоговых операторно-структурных графах появляются звенья (дуги графа) отображающие нелинейные компоненты. Линейные составляющие ММ систем отображаются звеньями или дугами, описываемыми ПФ.

Основная особенность в преобразовании таких схем заключается в невозможности их сверток до единственного звена, описываемого ПФ или ПМ, или же одного нелинейного звена с нелинейной вектор-функцией или скалярной нелинейной функцией. Для систем рассматриваемого класса обычно в структурных

преобразованиях ограничиваются агрегированием (сверткой) по отдельности линейных и нелинейных частей ОСС, причем лишь в тех частях ОСС, в которых присутствуют звенья только одного из этих типов. В итоге максимально агрегированная ОСС нелинейной системы может состоять из нескольких чередующихся между собой линейных и нелинейных звеньев [2]. На рисунке 7.3 приведен пример такой ОСС, где выделены участки, подлежащие эквивалентированию.

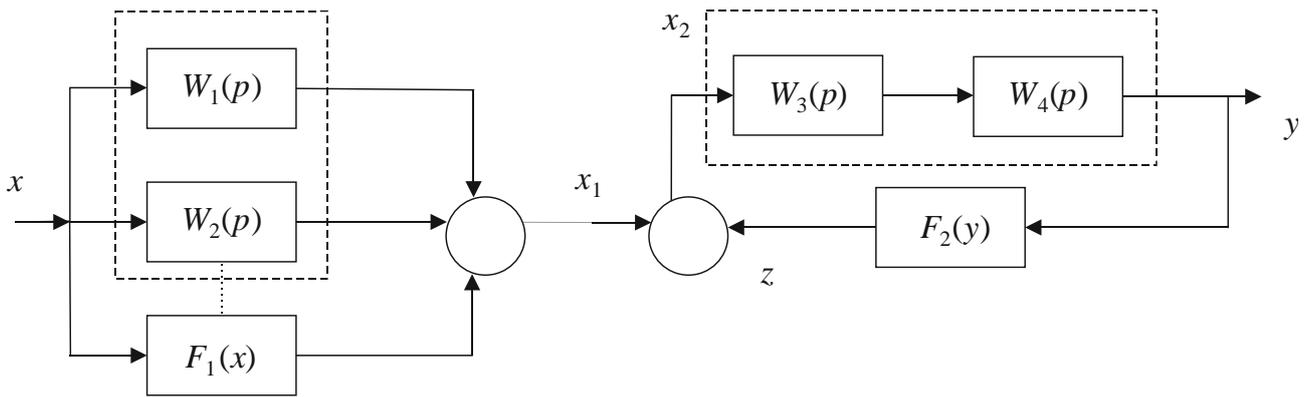


Рис. 7.3 – ОСС, где выделены участки, подлежащие эквивалентированию

Здесь выделены участки ОСС с параллельными ПФ: $W_1(p)$ и $W_2(p)$ – и последовательным с ПФ $W_3(p)$ и $W_4(p)$ соединением линейных звеньев, которые могут быть заменены своими эквивалентами с ПФ $W_{12}(p)$ и $W_{34}(p)$ соответственно.

7.4 Операторно-структурные схемы дискретных и дискретно-непрерывных систем

ОСС широко используются и для представления математической модели дискретных систем [3]. Исходные математические модели систем этого класса могут быть *двух типов*:

- содержащие только разностные уравнения;
- содержащие дополнительно к разностным уравнениям и уравнения с непрерывными координатными переменными.

Системы первого типа характеризуются только переменными, квантованными по уровню и по времени. Для того чтобы представить эти системы в форме ОСС применяют дискретное преобразование Лапласа.

Обобщенный дискретный регулятор (ОДР)

Как и аналоговый, ОДР можно представить *тремя составляющими* – контур упреждения по опорному значению, контур ОС по выходу процесса и контур упреждения по возмущению (рис. 7.4).

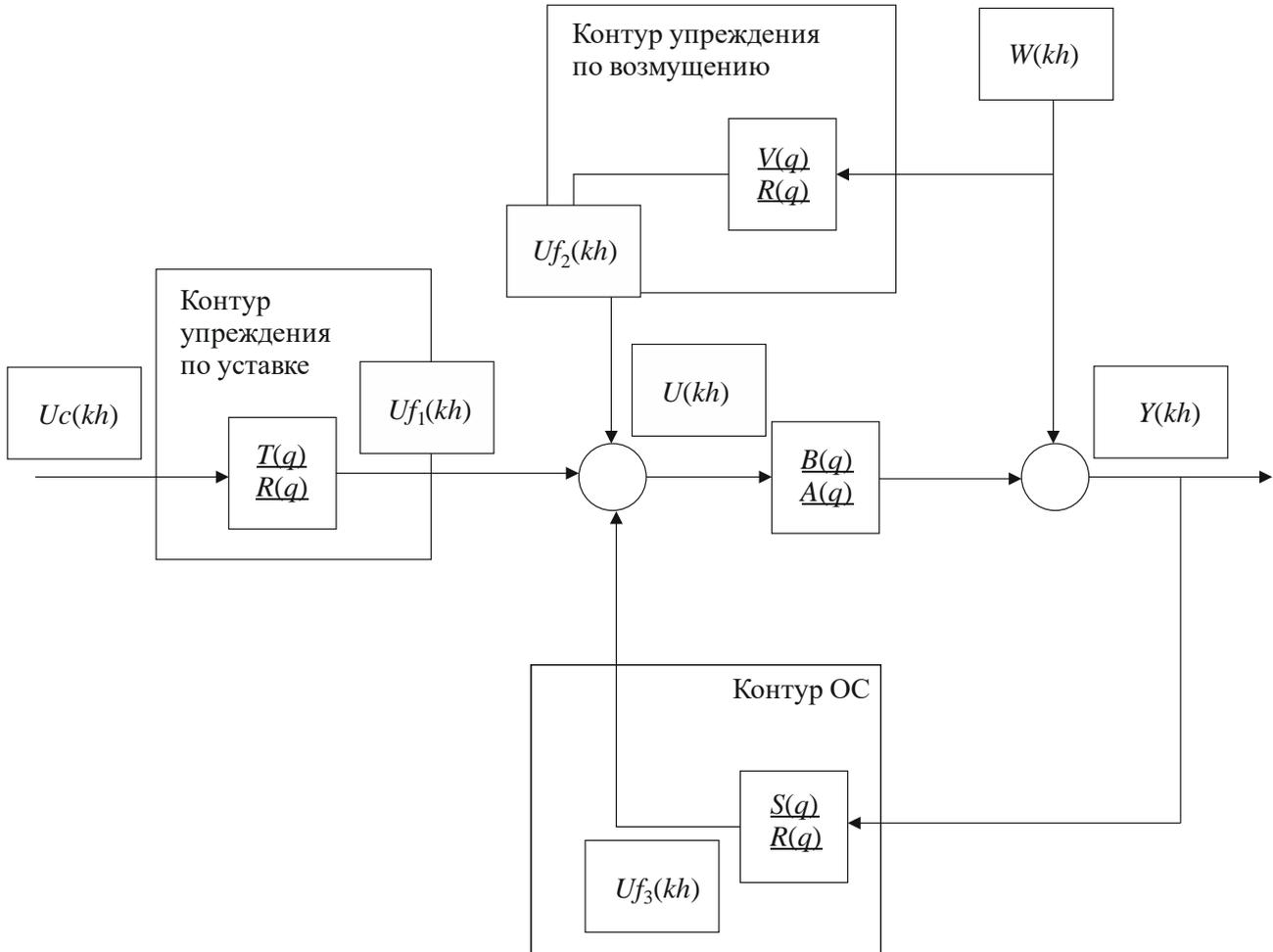


Рис. 7.4 – Обобщенный дискретный регулятор

Уравнения регулятора:

$$U(kh) = Uf_1(kh) - Uf_3(kh) - Uf_2(kh) = \\ = (T(q) / R(q))Uc(kh) - (S(q)R(q))Y(kh) - (V(q)R(q))W(kh).$$

Контрольные вопросы

1. Что понимается под ОСС?
2. Назовите основные типы звеньев ОСС.
3. Как по исходной математической модели системы построить ее ОСС?
4. Каковы основные правила преобразований ОСС?
5. Каково структурное отображение математической модели многомерной по «входу-выходу» системы?

6. Чем отличаются ОСС нелинейных систем? Каковы правила их преобразований?
7. Каковы особенности ОСС дискретных систем?

Литература

1. Востриков, А. С. Теория автоматического регулирования : учеб. пособие / А. С. Востриков, Г. А. Французова. – Новосибирск : Изд-во НГТУ, 2006. – 368 с.
2. Справочник по теории автоматического управления / под ред. А. А. Красовского. – М. : Наука, 1987. – 712 с.
3. Кориков, А. М. Основы теории управления : учеб. пособие для вузов / А. М. Кориков. – Томск : Изд-во НТЛ, 2002. – 391 с.

8 Конечные автоматы

8.1 Способы описания конечного автомата

В контактных и логических схемах значения выходных переменных определяется только комбинацией переменных на входах в данный момент времени, поэтому их называют *комбинационными схемами*. В более общем случае выходные переменные зависят от значений входных переменных не только в данный момент времени, но и от их предыдущих значений или значения выходных переменных определяется последовательностью значений входных. Схемы с такими свойствами называются *последовательными*. Если входные и выходные переменные принимают значения из конечных алфавитов, то такие схемы называются *конечными автоматами (КА)* [1]. Если зависимости между входными и выходными переменными выражаются булевыми функциями, то конечные автоматы определяются двоичным структурным алфавитом. В реальных условиях сигналы представляются непрерывными функциями времени, поэтому для надежного различения сигналов требуется, чтобы новые значения на входах появлялись после окончания переходных процессов, связанных с предыдущими значениями.

При рассмотрении логической структуры автоматов обычно отвлекаются от существа динамических процессов и считают, что переменные изменяются мгновенно, в некотором интервале времени, называемом *тактами* (Δt).

Тактовые моменты $t_{v+1} = t_v + \Delta t$ определяются синхронизирующими сигналами.

Дискретное автоматное время: t_v ($v = 1$), в котором его значение зависит не от физического времени, а от номера такта v .

Выходные функции последовательных схем в качестве своих аргументов кроме входных переменных обязательно содержат совокупность промежуточных переменных S_1, S_2, \dots, S_k , характеризующих состояние схемы (системы).

Набор всех возможных состояний, присущих данной схеме, – *множество состояний*. Таким образом, состояние конечного автомата в любой тактовый момент характеризуется значениями такой совокупности переменных, которая вместе с заданными значениями входных переменных позволяет определить выходные переменные в данный тактовый момент и состояние в следующий момент.

Поэтому автоматы должны обладать способностью сохранить предыдущее состояние до следующего такта, в связи с чем их называют *автоматом с памятью*. В качестве памяти широко используются элементы задержки, на выходах которых повторяются входные воздействия со сдвигом во времени на интервал между тактами Δt , например триггер.

Обычно конечный автомат представляют в виде *пятерки параметров* $M = (A, S, Z, \delta, q)$, где

- $A = \{a_0, a_1, \dots, a_n\}$ – конечное множество входных символов (входной алфавит) автомата, из которого формируются строки, считываемые автоматом;
- S – множество внутренних состояний;
- Z – допустимый список выходных символов (выходной алфавит);
- δ – заданное отображение множества $S \times A$ во множество S (функция переходов автомата в следующее состояние), $\delta: S \times A \rightarrow S$;
- $q: S \times A \rightarrow Z$ – функция выхода.

Тем самым конечный автомат математически описывается *тремя множествами и двумя функциями*. Действие его состоит в том, что он считывает последовательность входных символов (программу) и затем выдает последовательность выходных символов [2].

Конечный автомат, находящийся сначала в состоянии s_j , считывает по одному символу входной строки, начиная с a_k (программа). Считанный символ переводит автомат в новое состояние из Z в соответствии с функцией переходов. Если по завершении считывания входного слова (цепочки символов) автомат оказывается в одном из допускающих состояний, то слово *принимается* автоматом. В этом случае говорят, что оно принадлежит языку данного автомата. В противном случае слово *отвергается*.

Дадим наглядный алгоритм работы КА. Будем предполагать, что программа записана на входном массиве. КА считывает с него входные символы один за другим. По прочтении каждого входного символа выдается выходной символ на выходной ленте и автомат переходит в следующее состояние прежде чем считать следующий символ программы.

8.2 Примеры конечных автоматов

Пример 1. Рассмотрим следующий конкретный КА, где $M = (A, S, Z, \delta, q)$. Входной алфавит $A = \{0, 1\}$, выходной алфавит $Z = \{0, 1\}$, три внутренних состояния $S = \{s_0, s_1, s_2\}$, функции перехода и выхода задаются предписаниями (табл. 8.1).

Таблица 8.1 – Функции перехода и выхода

Δ			Q		
$(s_0, 0)$	\rightarrow	s_1	$(s_0, 0)$	\rightarrow	0
$(s_0, 1)$	\rightarrow	s_0	$(s_0, 1)$	\rightarrow	1
$(s_1, 0)$	\rightarrow	s_2	$(s_1, 0)$	\rightarrow	1
$(s_1, 1)$	\rightarrow	s_1	$(s_1, 1)$	\rightarrow	0
$(s_2, 0)$	\rightarrow	s_0	$(s_2, 0)$	\rightarrow	1
$(s_2, 1)$	\rightarrow	s_2	$(s_2, 1)$	\rightarrow	0

Подадим на вход последовательность 0 1 0 1. Если КА находился в состоянии s_0 , то он, считав первый символ 0, перейдет в состояние s_1 и выдаст 0. Считав затем 1, он останется в состоянии s_1 и выдаст 0. Считав следующий символ 0, он перейдет в состояние s_2 и выдаст 1. Считав последнюю 1, КА закончит работу в состоянии s_2 с выходом 0. Таким образом, КА преобразовал вход 0101 в выходную последовательность 0010.

Есть 2 удобных способа описать этот КА. Прежде всего можно построить помеченный ориентированный граф, называемый диаграммой состояний. Этот граф представлен на рисунке 8.1.

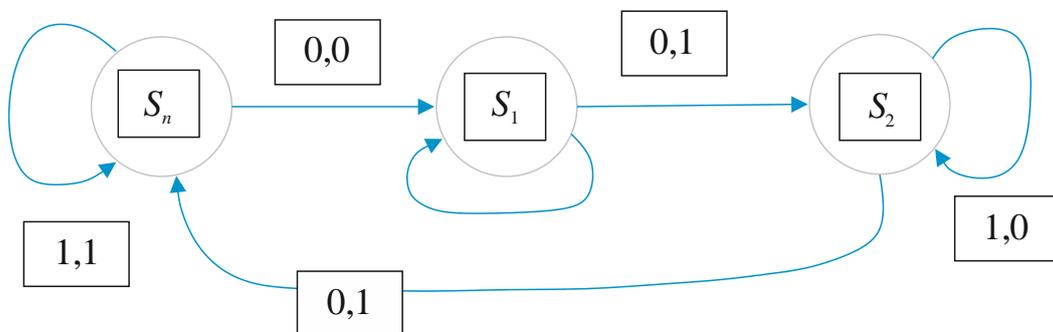


Рис. 8.1 – Диаграмма состояний

Вершины этого графа помечены символами, обозначающими внутренние состояния. Каждое ребро помечено парой символов a и z , где a – входной символ, вызывающий переход в следующее состояние, отвечающее этому ребру, а z – выходной символ, который автомат выпечатывает.

Второй способ описания – таблица состояний. Из рисунка 8.1, соответствующего нашему КА, ясно, что это просто табличное представление функций δ и q (табл. 8.2).

Таблица 8.2 – Табличное представление функций δ и q

Текущее состояние	Следующее состояние		Выход			
	δ	0	1	q	0	1
s_0		s_1	s_0		0	1
s_1		s_2	s_1		1	0
s_2		s_0	s_2		1	0

Оба способа имеют свои *преимущества и недостатки*. Таблица обычно удобна при вычислениях, а диаграмма нагляднее. Например, по диаграмме легко обнаружить состояния, недостижимые из других состояний. На рисунке 8.2 показана диаграмма состояний КА, у которого состояние s_1 недостижимо, если автомат начинает из состояния s_0 или s_2 .

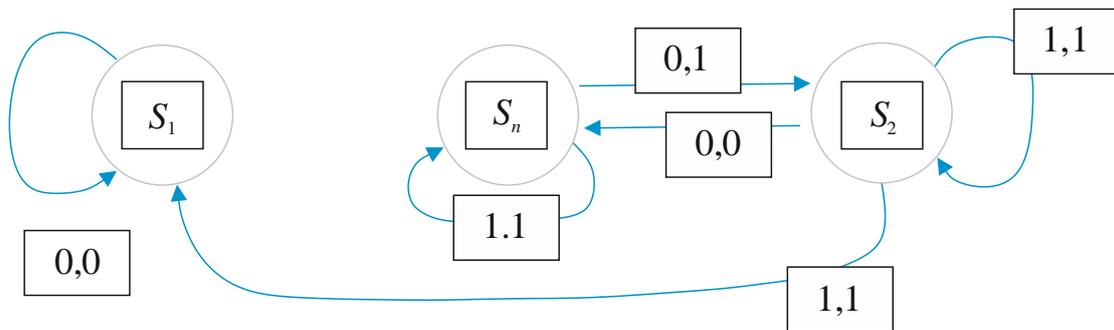


Рис. 8.2 – Диаграмма с недостижимыми состояниями

Пример 2. Автомат с двумя состояниями, изображенный на рисунке 8.3, есть автомат для проверки четности. Автомат считывает входную последовательность из нулей и единиц, и его состояние в любой момент времени совпадает с начальным, скажем s_0 , если число считанных к этому моменту единиц четно, и равно s_1 , если число считанных единиц нечетно (табл. 8.3). Выходная последовательность совпадает со входной.

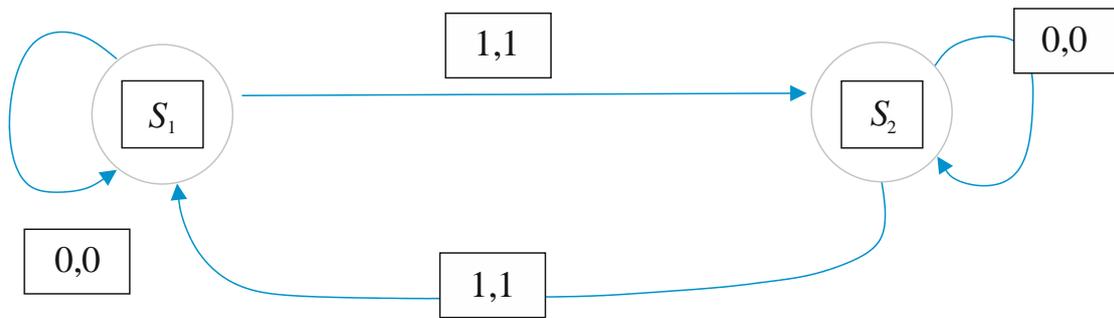


Рис. 8.3 – Автомат для проверки четности

Таблица 8.3 – Табличное представление функций δ и q

Текущее состояние	Следующее состояние		Выход			
	δ	0	1	q	0	1
s_0		S_0	S_1		0	1
s_1		S_1	S_0		0	1

Пример 3. Автомат, изображенный на рисунке 8.4, проверяет четность и вы печатывает EVEN (четный) или ODD (нечетный) в ответ на запрос, который соответствует входному символу. Считав A , автомат вы печатывает EVEN, если число ранее считанных единиц было четно, и ODD – если нечетно. Например, входная последовательность 0110A1110A будет переработана в 0110 1110 ODD.

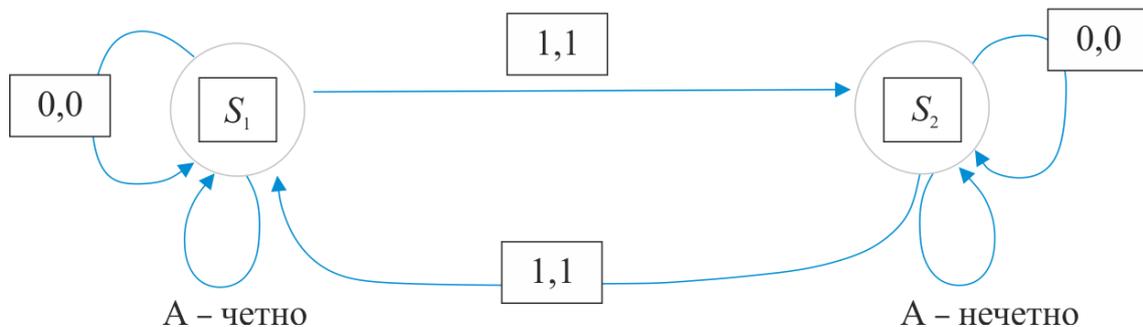


Рис. 8.4 – Автомат, вы печатывающий информацию о четности

Пример 4. Рассмотрим задачу о разгрузке машин на складах. Будем полагать, что время разгрузки товара будет прямо пропорционально количеству выгруженного на складе товара. Пусть складов будет три, как показано на рисунке 8.5. Количество товара на машине обозначим через A_c . В исходном состоянии на складах хранится A_1, A_2, A_3 единиц данного товара при максимально возможном количестве $A_{\max} = 100$.

При въезде машин на склад определяется количество выгружаемого на складе товара и длительность разгрузки. Предполагается, что обычно на складе остается хоть часть товара, и значит, время разгрузки отлично от нуля. Два предельных случая: на машине нет товара и склад переполнен – приводят к нулевому времени разгрузки. Для диспетчера, следящего за разгрузкой товара, это означает, что машина проезжает склад, не останавливаясь. Предельные случаи не нарушают работу карты поведения, превращая ее в классический КА.

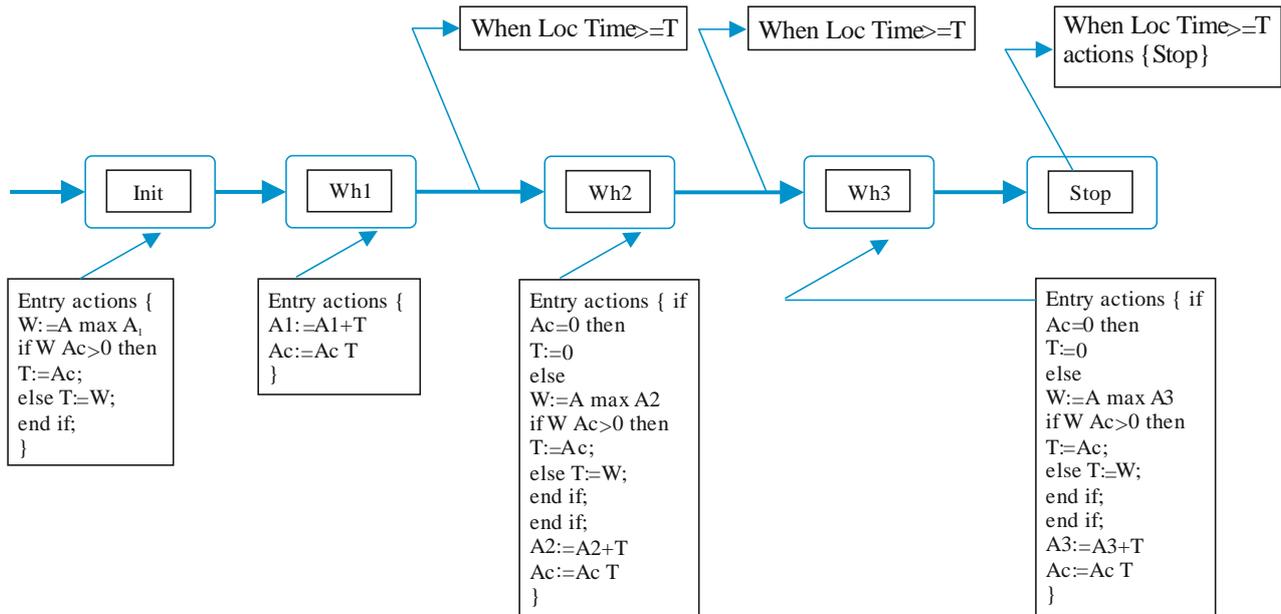


Рис. 8.5 – Карта состояний, реализующая алгоритм разгрузки автомашин на складах

При произвольных начальных условиях может реализоваться ситуация, когда машина будет разгружаться на всех трех складах, например, при $A_1 = 98$, $A_2 = 97$, $A_3 = 1$, со временами разгрузки $T = 2, 3, 5$ единиц модельного времени. Но может случиться, что первые два склада будут заполнены: при $A_1 = 100$, $A_2 = 100$, а третий пустой $A_3 = 0$, тогда машина мгновенно минует два первых склада и разгрузка произойдет на третьем складе. В этом случае наша схема для первых двух состояний ведет себя как классический КА, мгновенно переходя из одного состояния в другое.

Приведенные выше способы задания конечного автомата являются наиболее употребляемыми, поэтому не вызывает удивления факт их использования в системах, предоставляющих возможность моделирования (в том числе и визуального) алгоритмов.

8.3 Системы, предоставляющие возможность построения конечных автоматов

Из крупных систем, использующих в своей основе теорию конечных автоматов, наиболее известными являются: *Simulink/Stateflow*, подсистема среды *Matlab*; проект *JFLAP*, предназначенный для обучения; реализующее язык *SFC* дополнение интегрированной среды разработки программ контроллеров *SIMATIC PCS 7*. Итак, *SIMATIC PCS 7* (полное название: *SIEMENS SIMATIC Process Control System 7*) – набор программ для создания автоматизированных систем управления технологическими процессами, полностью отвечающих требованиям концепции «полностью интегрированная автоматизация» (англ. *Totally Integrated Automation*), разработанных и внедряемых компанией *SIEMENS*. В этом программном продукте для нас интерес представляет только одно дополнение, реализующее язык *SFC* (*Sequential Function Chart*). *SFC* – графический язык, предназначенный для написания программ последовательного управления технологическим процессом, описывающий его в форме, близкой к диаграмме состояний.

Контрольные вопросы

1. Дайте определение КА.
2. Какие множества и функции характеризуют КА?
3. Как называется помеченный ориентированный граф для описания КА?
4. Какие состояния описывает орграф КА?
5. Характеризуйте взаимосвязь КА и карты состояний.

Литература

1. Бенькович, Е. Практическое моделирование динамических систем / Е. Бенькович, Ю. Колесов, Ю. Сениченков. – СПб. : БХВ-Петербург, 2002. – 464 с.
2. Бирхгоф, Г. Ф. Современная прикладная алгебра / Г. Ф. Бирхгоф, Т. Барти. – М. : Мир, 1976. – 399 с.

9 Гибридные динамические системы

Термин «динамическая система» первоначально отождествлялся с автономной системой обыкновенных дифференциальных уравнений. Сейчас такими являются любые системы, чье поведение зависит от времени. Среди динамических систем особо выделяют *гибридные системы*, простейшие примеры которых кратко рассмотрим ниже.

9.1 Прыгающий мячик

Будем изучать движение шарика массы m , брошенного с начальной скоростью V_0 под углом α к горизонтальной плоскости в плоскопараллельном поле тяготения [1]. После броска шарик взлетит вверх, затем вниз, ударяется о плоскость и отскакивает вверх. При абсолютно упругом ударе траектория шарика будет выглядеть так, как показано на рисунке 9.1. Здесь y – высота полета, x – дальность.

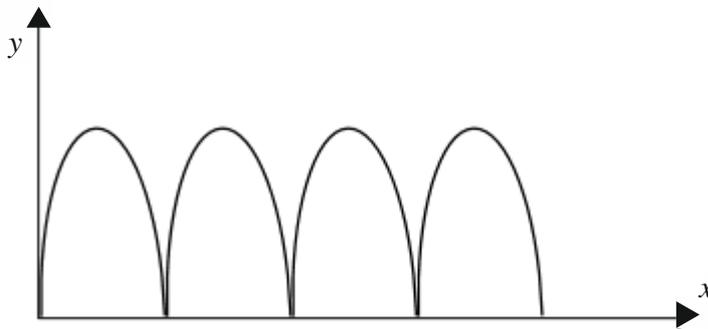


Рис. 9.1 – Траектория шарика при упругом ударе

Из этого рисунка хорошо видны две чередующиеся фазы – полет и отскок. В фазе полета поведение шарика описывается системой уравнений:

$$\begin{cases} \frac{dx}{dt} = V_x \\ \frac{dy}{dt} = V_y \\ \frac{dV_y}{dt} = -g \end{cases} \quad (9.1)$$

Здесь $[x, y]$ – дальность и высота полета; $[V_x, V_y]$ – горизонтальная и вертикальная составляющие скорости; g – ускорение силы тяжести. Здесь можно рассматривать *отскок* как мгновенное дискретное действие, в результате которого происходит изменение знака вертикальной составляющей скорости на противоположный. Момент отскока определяется при помощи выполнения неравенств вида $y \leq 0, V_y < 0$.

Алгоритм смены фаз движения шарика схематически показан на рисунке 9.2. В начальный момент $t = 0, x = 0, y = 0, V_x = V \cos \alpha, V_y = V \sin \alpha$, шарик находится в *состоянии «полет»* и движется согласно системе (9.1). В некоторый момент времени t^* , который определяется логическим предикатом, полет заканчивается и уравнения (9.1) перестают работать, шарик мгновенно меняет направление движения ($V_y = -V_y$) и полет продолжается, но уже с новыми начальными условиями.

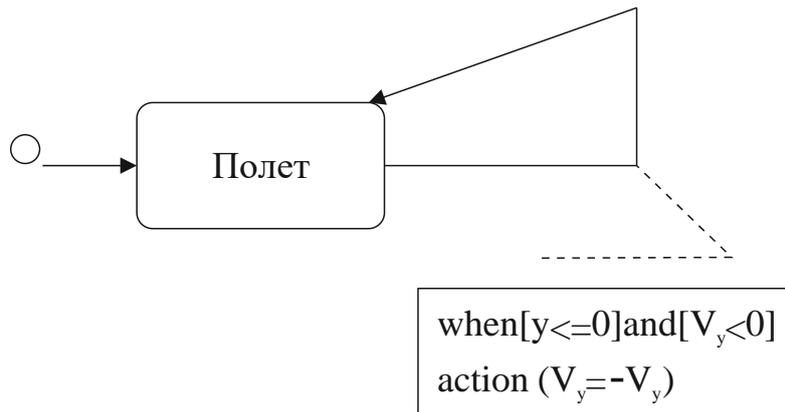


Рис. 9.2 – Алгоритм смены фаз движения шарика

Кружок со стрелкой, указывающий на состояние «полет» говорит о том, что это *начальное состояние*. Стрелка, исходящая из кружка и возвращающаяся в него, показывает, что состояние «полет» периодически повторяется. Стрелке предписаны условия выхода из текущего состояния и мгновенные действия по переходу в новое состояние.

Описание поведения таких систем включает в себя не только *описание непрерывного действия*, но и *описание дискретных действий*, указывающее, когда наступают изменения и какие действия следует выполнить.

Поведение гибридной системы можно рассматривать как *склею* непрерывных поведений в виде *кусочно-непрерывной функции*. Место склейки (момент времени t^*) называют *временной щелью*. Ее особенности иллюстрирует график изменения вертикальной составляющей скорости (рис. 9.3).

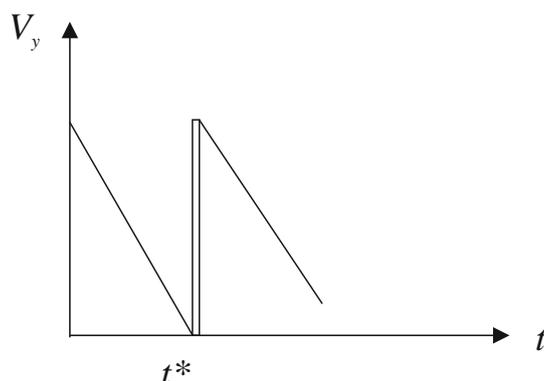


Рис. 9.3 – Изменения вертикальной составляющей скорости шарика

Переменная V_y в один и тот же момент времени t^* смены поведения имеет два значения – отрицательное и положительное, то есть имеем дело с *многозначными функциями*. Это следствие нашего допущения о том, что процесс отскока имеет *нулевую длительность*.

Как видим, *гибридные* или *событийно-управляемые* системы являются обобщением классических динамических систем, у которых в различных областях физического пространства меняются *уравнения, размерность, состав фазового вектора*. При их описании приходится использовать как традиционные непрерывные модели (системы дифференциальных уравнений), так и классические дискретные (конечные) автоматы, опирающиеся на различные модели времени. Для такого совмещения необходимо рассмотреть *новую модель времени*.

9.2 Время

При рассмотрении реальных физических процессов часто вместо непрерывного вводят *дискретное время*. Под *дискретным временем* понимают упорядоченную, неограниченно возрастающую последовательность вещественных чисел $(t_0, t_1, \dots, t_k, \dots)$, а чаще всего множество целых чисел.

Дискретное время является независимой переменной в разностных уравнениях вида $z_{i+1} = F(z_i, \dots, z_{i-k})$, $z_{i+1} = z(i+1)$; $t_i = i$.

Здесь искомая функция $z(t)$ определена только в точках сетки t_i , поэтому она и получила название *сеточной*. Используются также сеточные функции, временные точки которых могут быть расположены неравномерно (переменный шаг).

Дискретное время естественным образом появляется там, где наблюдается разделение поведения на *быстрые* – дискретные действия, и *медленные* – непрерывные и длительные действия. В этом случае при описании совместной системы «объект – регулятор» возникают одновременно дифференциальные уравнения (уравнения объекта):

$$\frac{dx}{dt} = f(x(t), u_n) \quad (9.2)$$

и разностные уравнения (уравнения регулятора):

$$\begin{aligned} u_{n+1} &= G(x(t_n), u_n); \\ u_n &= u(t_n), t_{n+1} = t_n + h. \end{aligned} \quad (9.3)$$

На рисунке 9.4 представлена система, состоящая из ПД-регулятора (CU) и объекта управления (CO).

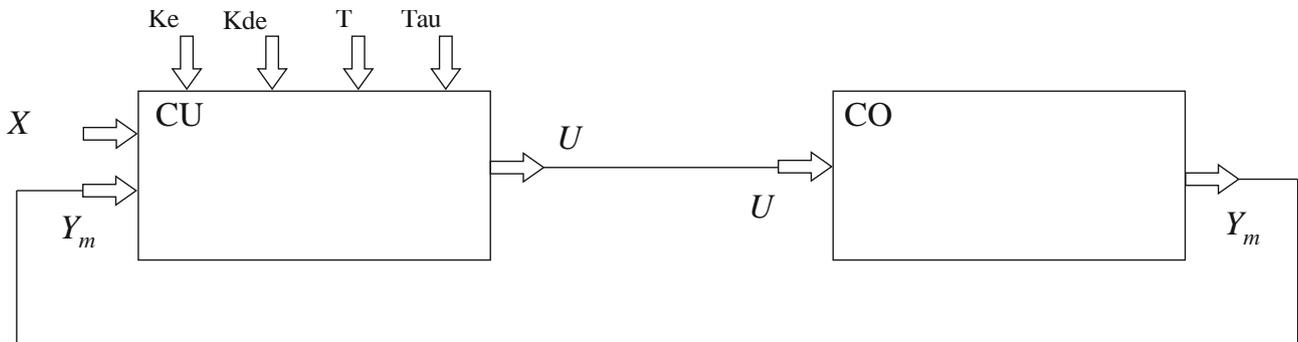


Рис. 9.4 – Система, состоящая из ПД-регулятора (CU) и объекта управления (CO)

Поведение объекта управления описывается уравнениями типа (9.2). Здесь Y – непрерывная координата объекта; Y_m – ее квантованное значение; U – управляющий сигнал; Tau – задержка в выдаче сигнала управления.

9.3 Пространство

В ньютоновских законах движения наряду с независимой переменной «время» присутствуют пространственные координаты (x, y, z) движущейся точки. Четверка (t, x, y, z) определяет положение тела в пространстве и времени. Можно следить за положением точки $x(t), y(t), z(t)$ как функцией времени,

графики которых в отдельности будем далее называть *временными диаграммами*. Можно также следить за положением точки $A: (x_a(t), y_a(t), z_a(t))$ в осях x, y, z . Тогда обычно говорят о фазовых диаграммах. На следующих рисунках показано, как выглядит решение дифференциального уравнения, представленное в виде временной (рис. 9.5) и фазовой (рис. 9.6) диаграмм.

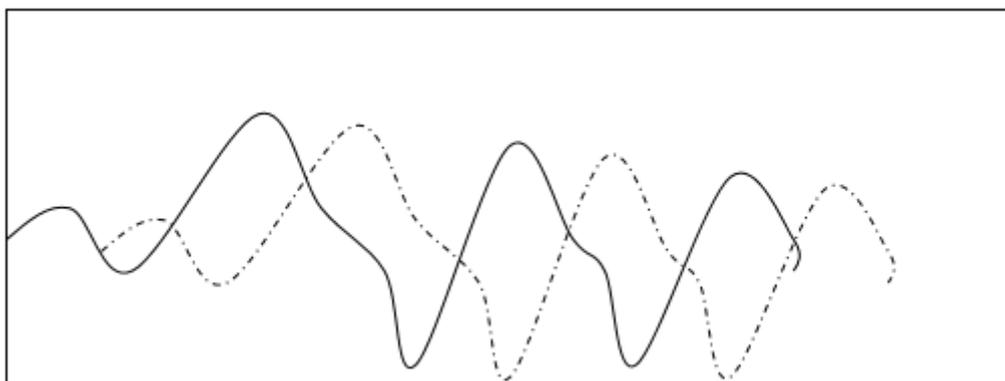


Рис. 9.5 – Временная диаграмма

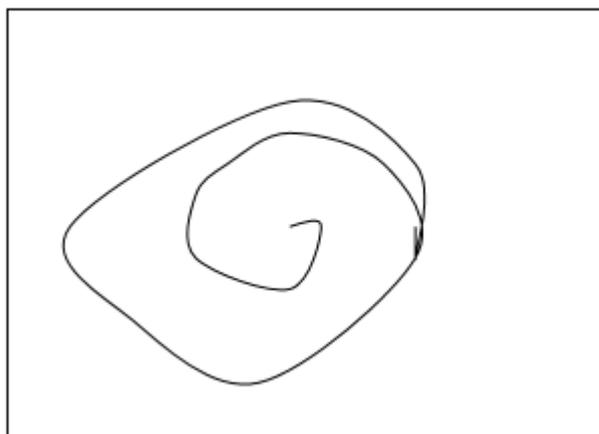


Рис. 9.6 – Фазовая диаграмма

Обобщая понятия физического пространства и фазовых переменных x, y, z можно говорить о фазовом векторе $X \in R^n$ вещественного пространства размерности n . В общем случае *фазовый вектор* $X = \{x_1, x_2, \dots, x_n\}$ – это множество независимых фазовых координат, каждая из которых имеет свое множество значений $x_i \in D$, соответственно *фазовая траектория* – это след точки $(x_1(t), x_2(t), \dots, x_n(t))$ в пространстве $D_1 \times D_2 \times \dots \times D_n$.

Множества значений функций D_i могут быть конечными, например функция

$$\text{sign}(t) = \begin{cases} 1, & t > 0; \\ 0, & t = 0 \\ -1 & t < 0 \end{cases},$$

оставаясь вещественной, принимает только три значения: $\{-1, 0, 1\}$. Это *дискретная, или вещественная кусочно-постоянная, функция*. Примером *дискретной функции* может служить также и функция, которую можно использовать для моделирования маяка:

$$\text{light}(t) = \begin{cases} \text{red}, & t = 1, 3, 5, \dots; \\ \text{green}, & t = 2, 4, 6, \dots \\ \text{none}, & t \neq 1, 2, 3, 4, \dots \end{cases}.$$

На рисунке 9.7 показано, как можно реализовать эту кусочно-линейную функцию в системе моделирования.

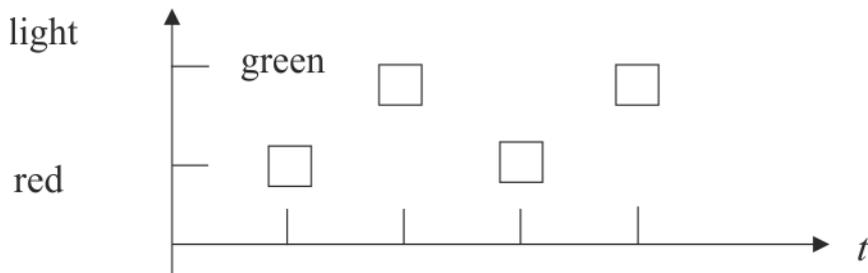


Рис. 9.7 – Кусочно-линейная функция

События

Поведение динамической системы с фазовым вектором $x(t)$ 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 говорит о том, что произошло, или характеризует явление.

В случае если пространственные координаты и время непрерывны, *временная траектория* $(t, x(t))$ определяет бесконечное, несчетное множество событий на промежутке $[0, T]$ и на множестве $[x_{\min}, x_{\max}]$ всех наблюдавшихся явлений.

В дискретном случае обычно говорят не о всем множестве событий, а лишь об *отдельных (особых) событиях*. Если речь идет о кусочно-постоянных функциях (КПФ), то из-за мгновенного перехода опознавать изменения можно, следя не за *фронтом* сигнала, а за его *уровнем*.

Использование только второй координаты события для КПФ породило название «*дискретные события*» и еще один способ описания этих функций. Их изображают в виде графа, приписывая каждому узлу одно значение функции, например периодической функции *light*, описывающей функционирование светофора:

$$\begin{aligned} \text{light}(i + T) &= \text{light}(i), T = 3, i = 1, 2, 3. \\ \text{light}(i) &= \begin{cases} \text{green}, & i = 1; \\ \text{yellow}, & i = 2 \\ \text{red}, & i = 3. \end{cases} \end{aligned}$$

соответствует граф, изображенный на рисунке 9.8.

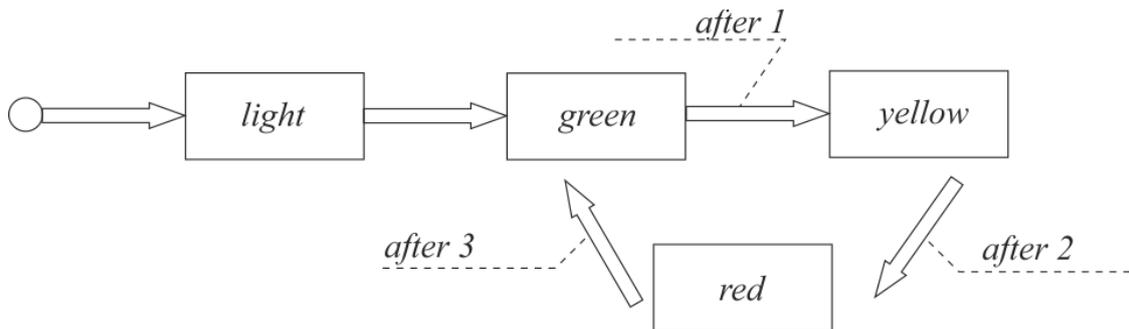


Рис. 9.8 – Граф, описывающей функционирование светофора

С событиями тесно связаны еще два понятия: *сигнал* и *сообщение*. Когда не интересует суть события, а только факт того, что данное событие произошло, ему можно дать имя и связать с этим именем *сеточную функцию*, принимающую, например, значение *true* в момент, когда происходит указанное событие. Такую функцию принято называть *сигналом* – $\text{signal} := \text{signal}((t^*, x(t^*)) \rightarrow \text{true})$.

На рисунке 9.9 показано, как можно моделировать маяк, используя сеточные функции *signal*. Это реализовано с помощью двух специальных переменных типа *red* и *green* типа *signal*, определенных только в точках 1, 2, 3...

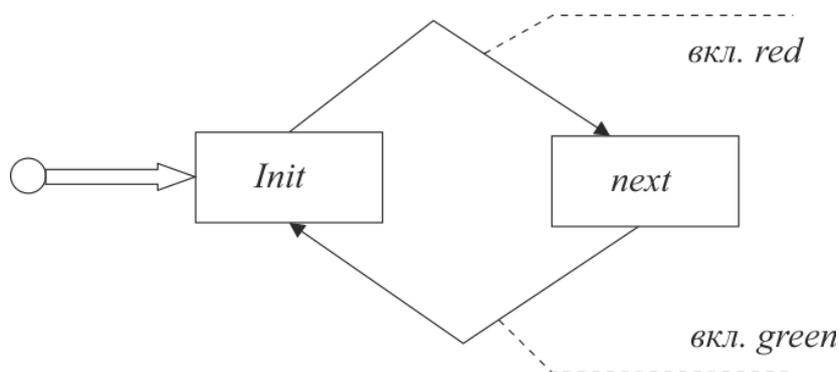


Рис. 9.9 – Моделирование маяка с использованием сеточных функций

Контрольные вопросы

1. Какие системы называются гибридными?
2. Какие типы поведения характеризуют гибридные системы?
3. Какие имеются фазы движения у скачущего шарика?
4. Как можно рассматривать отскок в полете шарика?
5. Какое время используется при рассмотрении реальных физических процессов?
6. Как характеризуются временные диаграммы?
7. Что представляют собой фазовые диаграммы?
8. Что принято называть событием?
9. Поясните такие понятия, как сигнал и сообщение.

Литература

1. Бенькович, Е. Практическое моделирование динамических систем / Е. Бенькович, Ю. Колесов, Ю. Сениченков. – СПб. : БХВ-Петербург, 2002. – 464 с.

10 Гибридные автоматы

10.1 Типы гибридных систем

Теперь можно перейти к построению гибридных автоматов. Используемый для этого формализм должен содержать средства для описания непрерывного поведения и механизм смены состояний. Здесь предлагается *объединить язык дифференциальных уравнений и язык конечных автоматов в единый формализм*, позволяющий описать смену поведения в различных областях фазового пространства.

Из рассмотренных примеров можно выделить три типа гибридных систем:

- 1) сохраняющие единое описание во всем пространстве и меняющие только *начальные условия* при переходе из области в область;
- 2) меняющие *значения параметров* в правых частях в зависимости от наступления событий;
- 3) меняющие *правые части и число уравнений* в момент, когда происходит смена поведения.

10.2 Смена начальных условий

Возьмем для примера систему, которая описывается линейными дифференциальными уравнениями (ДУ) второго порядка с постоянной матрицей. Ее *фазовым пространством* являются квадранты фазовой плоскости или ячейки. Собственные значения у этой системы вещественны и имеют разные знаки.

Свяжем состояния гибридной системы с ячейками фазового пространства. Зная их границы, действуем следующим образом:

- 1) выберем начальные условия в одной из них;
- 2) зафиксируем время или кривую внутри ячейки, по достижению которых решение останавливается;
- 3) выбираются новые начальные условия, но уже в другой ячейке;
- 4) повторяем этот процесс по определенным правилам, получаем последовательность решений ДС, соответствующую решению гибридной (ГДС).

У построенной ГДС будет столько состояний, сколько ячеек последовательно будем проходить, а система уравнений для каждого состояния будет одной и той же. Условием перехода из состояния в состояние будет либо *заданное*

время, либо достижение неких граничных значений, а мгновенными действиями – смена начальные условия по алгоритму, зависящему от конкретной ячейки.

Пример линейной системы с особой точкой типа «седло»

В качестве примера выберем систему линейных ДУ на плоскости R^2 с особой точкой типа «седло»:

$$\frac{dx}{dt} = \Lambda x; \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}; \lambda_1, \lambda_2 \geq 0;$$

$$-a \leq x_1 \leq a; -b \leq x_2 \leq b; a, b > 0;$$

и соответствующими подобластями:

$$1 \rightarrow x_1 > 0; x_2 > 0;$$

$$2 \rightarrow x_1 < 0; x_2 < 0;$$

$$3 \rightarrow x_1 < 0; x_2 > 0;$$

$$4 \rightarrow x_1 > 0; x_2 < 0;$$

В начальный момент времени

$$1 \rightarrow 2: x_0 = [-x_2, -b]^T;$$

$$2 \rightarrow 3: x_0 = [x_2, b]^T;$$

$$3 \rightarrow 4: x_0 = [x_2, -b]^T;$$

$$4 \rightarrow 1: x_0 = [-x_2, b]^T,$$

где x_2 – значение второй компоненты вектора x , при переходе границы.

Время t^* – нахождение системы в конкретной подобласти при выбранной системе координат определяется только начальным значением первой компоненты векторов решения и равно

$$t^* = \frac{1}{\lambda_1} \ln \frac{a}{x_i(0)}.$$

При этом вторая компонента при достижении новой границы в момент t^* будет равна

$$x_2(t^*) = \exp(-\lambda_2 t^*) b.$$

Эта простая система демонстрирует уже довольно сложное поведение. В зависимости от значений параметров: a, b, x_{i0}, λ, T – время обхода подобластей $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ может стремиться либо к бесконечности, либо к нулю. Вырождение системы в чисто дискретную происходит при $T = 0$ (движение по углам прямоугольника, ограничивающего область задания гибридной системы). На

рисунке 10.1 показаны пример фазового портрета рассмотренной системы и зависимость решения от времени. На рисунке 10.1, а хорошо видны ячейки, совпадающие с квадрантами фазовой плоскости. Движение внутри каждой ячейки начинается со своих начальных условий и заканчивается, когда изображающая точка достигает заданных границ. На рисунке 10.1, б приведены решения как функции времени.

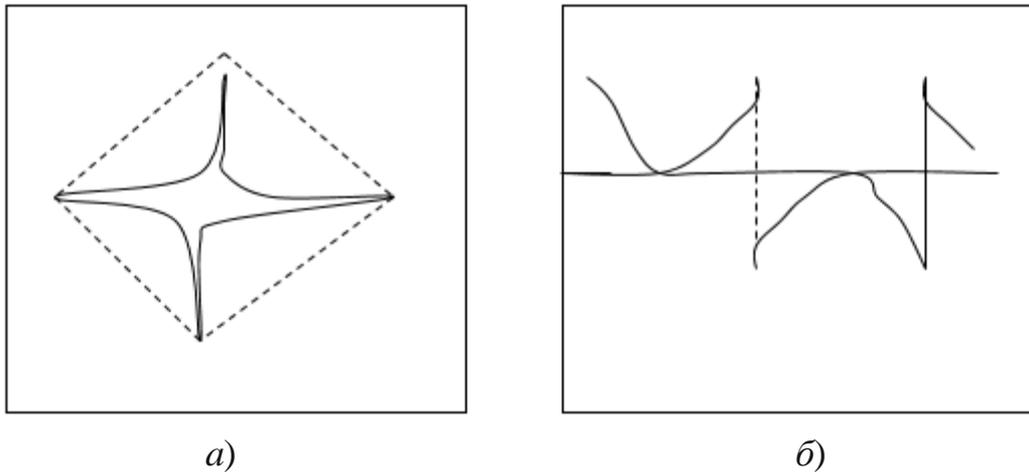


Рис. 10.1 – Решения системы как функции времени

10.3 Зависимость от параметров

Рассмотрим электрическую цепь (рис. 10.2), у которой в зависимости от положения ключа может меняться значение сопротивления.

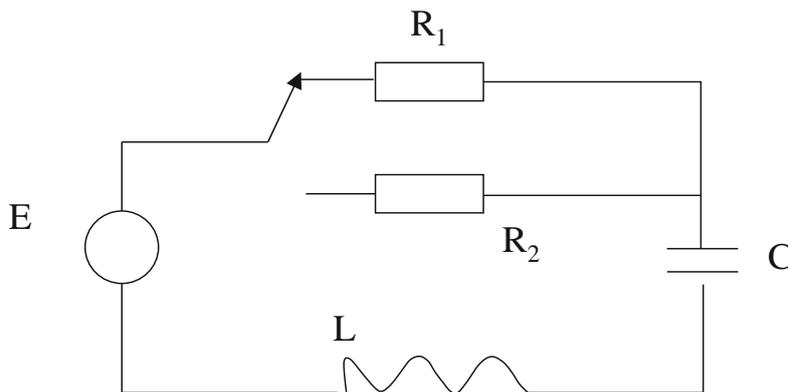


Рис. 10.2 – Электрическая цепь, у которой меняется значение сопротивления

Если R^* меняется периодически $R^*(t) = R^*(t + T)$, например, при $T = 2$

$$R^* = \begin{cases} R_1 & t \in [0, 1] \\ R_2 & t \in [1, 2] \end{cases},$$

то мы обеспечим периодическую смену характера поведения.

Поведение этой схемы описывается системой уравнений:

$$L \frac{di}{dt} + R^* i + u = E;$$

$$u = \frac{1}{C} \int_0^t i d\tau;$$

$$C \frac{du}{dt} = i.$$

Преобразуем последнюю систему уравнений, дополнив новым дифференциальным уравнением:

$$\frac{di}{dt} = \frac{1}{L} (-i(R^* + u) + E);$$

$$\frac{du}{dt} = \frac{1}{C} i;$$

$$\frac{dR^*}{dt} = 0; \quad t = 0; \quad R^* = R_1; \quad t = 1; \quad R^* = R_2, \dots$$

Последняя система уравнений отличается от предыдущей тем, что в ней периодически меняются начальные условия последнего уравнения в точках $t = 1, 2, 3$. В этих точках координаты i и u склеиваются, образуя непрерывные функции, а $R^*(t)$ меняется скачком, в результате чего появляется кусочно-непрерывная функция. Действительно, любую систему ДУ с кусочно-постоянным параметром P можно записать в виде:

$$\frac{dx}{dt} = f(x, P, t)$$

$$\frac{dP}{dt} = 0 \quad \{P(0) = c_1 \text{ в области } V_1 \quad .$$

$$\{(P(0) = c_2 \text{ в области } V_2$$

Изменение правой части ДУ возникает во многих практических задачах. В механических системах это часто связано с законом изменения возмущающей силы. *Пример:* колебания маятника, подверженного воздействию различных периодических функций:

$$\frac{d^2 x}{dt^2} + a \frac{dx}{dt} + f(x) = 0$$

$$f(x) = \left\{ \begin{array}{l} b_1 \sin(t) \text{ нпу} \quad 0 < x \leq \pi; \\ -b_2 \cos(t) \text{ нпу} \quad -\pi < x < 0. \end{array} \right\}$$

Перейдем к новой системе:

$$\begin{aligned} \frac{d^2x}{dt^2} + a \frac{dx}{dt} + k_1 f_1(x) + k_2 f_2(x) &= 0 \\ \frac{dk_1}{dt} &= 0; \\ \frac{dk_2}{dt} &= 0, \end{aligned}$$

у которой в точках $t = \pm j\pi$, $j = 0, 1, 2, 3\mathbb{K}$ меняются только начальные условия, а коэффициенты k_1 и k_2 меняются скачком в соответствующих областях.

10.4 Карты состояний и гибридные автоматы

Для описания поведения систем с переменными коэффициентами при неизменной форме уравнений хорошо подходит формализм Харела [1], называемый *картами состояний* (*statchart*).

Рассмотрим динамическую систему:

$$\frac{dx}{dt} = f(x), \quad x \in D, \quad D = D_1 \cup D_2$$

с правой частью, представимой в форме

$$\begin{aligned} f(x) &= k_1 f_1(x) + k_2 f_2(x) \\ k_1(x) &= \begin{cases} 1, & x \in D_1; \\ 0, & x \in D_2 \end{cases} \\ k_2(x) &= \begin{cases} 0, & x \in D_1; \\ 1, & x \in D_2 \end{cases} \end{aligned} .$$

Так как коэффициенты $k_1(x)$ и $k_2(x)$ – кусочно-постоянные функции, то их можно представить в форме конечного автомата с двумя состояниями: D_1 и D_2 . Выходные символы $\{k_1(x) = 1 \text{ и } k_2(x) = 0\}$ для состояния D_1 и $\{k_1(x) = 0 \text{ и } k_2(x) = 1\}$ для состояния D_2 , а входными символами являются два сигнала, говорящие о достижении этих областей. Состояние D_1 помечено как начальное.

Автомат, изображенный на рисунке 10.3, несколько отличается от классического конечного автомата. Он анализирует поступающие на него значения переменной x , входного сигнала и меняет свое текущее состояние, когда выполняются условия, приписанные дугам. В совокупности с приведенным ДУ карта со-

стояний (КС) дает полное описание непрерывно-дискретной (гибридной) системы. Можно сказать, что КС управляет значениями коэффициентов правой части ДУ и меняет характер непрерывного поведения.

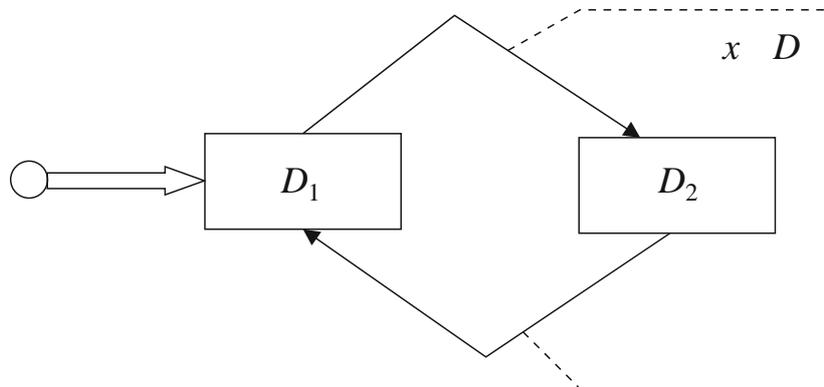


Рис. 10.3 – Карта состояний

Рассмотрим более сложную функцию $y = f(x)$, называемую *петлей гистерезиса* (рис. 10.4).

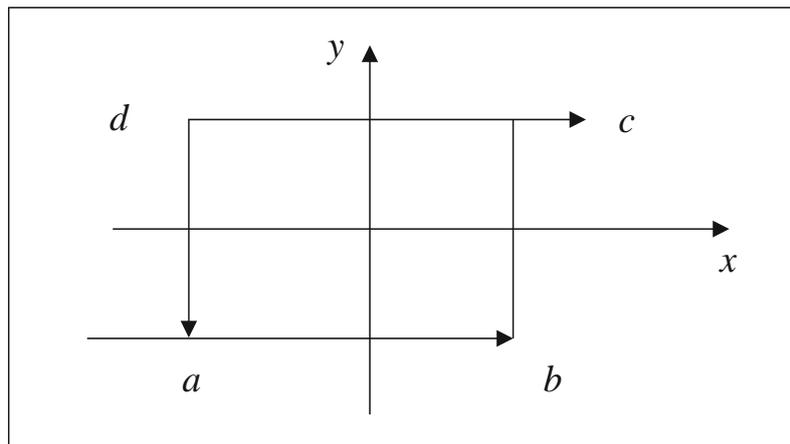


Рис. 10.4 – Петля гистерезиса

Особенность этой функции заключается в том, что она не является *однозначной*. Однако ее легко превратить в однозначную, если учитывать предысторию обращения. Так, если новое значение аргумента оказалось на участке неоднозначности, а предыдущее значение функции было отрицательным, то реализуется одно поведение (участок $a-b$), а если положительным – другое (участок $c-d$). На рисунке 10.4 это обозначено стрелками, указывающими направление обхода участка неоднозначности. Ее вертикальные участки соответствуют разрывам первого рода, а на горизонтальных участках $-(a-b)$ и $(a-b)$ существует два значения функции для каждого значения аргумента.

Рассмотрим карту (рис. 10.5), имеющую два состояния, соответствующие положительным и отрицательным значениям функций (P) и (N).

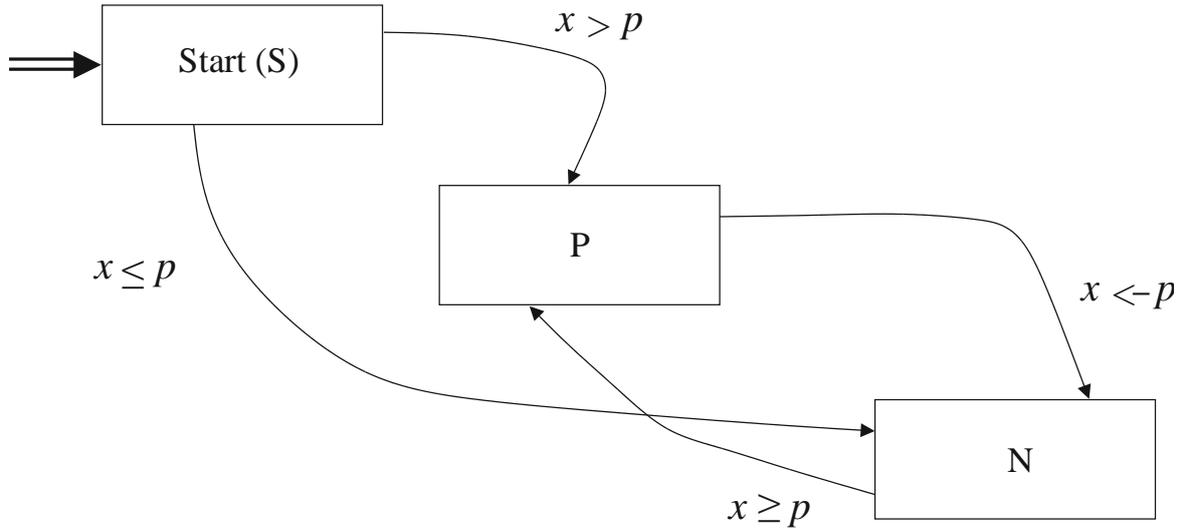


Рис. 10.5 – Карта, имеющая два состояния

В карте состояний по аналогии с графами есть множество узлов графа (прямоугольники $Start, P, N$), которые соединяются дугами $(S, P), (S, N), (P, N), (N, P)$. Карта состояний функционирует в непрерывном времени и, по существу, становится элементом описания динамической системы.

В заключение дадим определение гибридного автомата.

Гибридным автоматом $H_{sc} = \{S, SC, Q_0, F, Pr, Alg\}$ (обобщенной картой состояний) над множеством переменных $S = \{S^C, S^D\}$ называется граф $SC = \{Q, E\}$ с множеством узлов (состояний) $Q = \{q_1, q_2, \dots, q_n\}$ и множеством ориентированных дуг $E = \{(q_i, q_j), i, j = 1, n\}$ и функция $F: TxS \rightarrow S^c$, определяющая характер изменения непрерывных переменных во времени, в зависимости от значений дискретных переменных. Каждому узлу графа поставлена в соответствие совокупность дискретных переменных, принимающих постоянное значение в данном состоянии, а каждой дуге – пара, состоящая из предиката и алгоритма $(pr_i, alg_j), i = 1, k; j = 1, m$. Множество Q_0 определяет множество начальных состояний [2].

Контрольные вопросы

1. Назовите три типа гибридных систем.
2. Поясните, как связаны состояния гибридной системы с ячейками фазового пространства.
3. Что называется, картами состояний (*statechart*)?
4. Какова аналогия между картами состояний и графами?
5. Дайте определение гибридному автомату.

Литература

1. Harel, D. Statecharts: a visual formalism for complex systems // Science of Comp. Progr. – 1987. – Vol. 8, № 3. – P. 231–274.
2. Бенькович, Е. Практическое моделирование динамических систем / Е. Бенькович, Ю. Колесов, Ю. Сениченков. – СПб. : БХВ-Петербург, 2002. – 464 с.

11 Разработка моделей дискретно-событийных систем

11.1 Модели дискретно-событийных систем

Системы называются *дискретно-событийными*, если изменения переменных состояния в них происходят только в явно-определенные моменты времени или под влиянием явно-определенных событий. Находясь в некотором состоянии, дискретная система сохраняет его до наступления очередного события, после которого переменные системы изменяются скачком. Так, на примере банка, состояние системы изменяется, когда новый клиент входит в банк или освобождается кассир, а это уже можно считать мгновенными событиями, сопровождаемыми изменением состояния системы [1].

Stateflow – инструмент для численного моделирования систем, характеризующихся *сложным поведением* [2]. К числу таких систем относятся гибридные системы. *Примерами гибридных систем* могут служить системы управления, используемые в промышленности (автоматизированные технологические процессы), в быту (сложные бытовые приборы), в военной области (высокотехнологичные виды вооружений), в сфере космонавтики, транспорта и связи. Все эти системы состоят из аналоговых и дискретных компонентов. Поэтому *гибридные системы* – это системы со сложным взаимодействием дискретной и непрерывной динамики. Они характеризуются не только непрерывным изменением состояния системы, но и скачкообразными вариациями в соответствии с логикой работы управляющей подсистемы, роль которой, как правило, выполняет то или иное вычислительное устройство (*конечный автомат*). В том случае, когда логика работы управляющей подсистемы является *жесткой*, а внешние условия относительно стабильны, говорят о *трансформационных системах* (рис. 11.1). Для таких систем фазы получения информации, ее обработки и выдачи выходных сигналов четко разграничены. На момент обращения к системе все входные сигналы определены. Сигналы на выходах образуются после некоторого периода вычислений. Вычисления производятся по *некоторому алгоритму, трансформирующему (преобразующему) входной набор данных в выходной*.

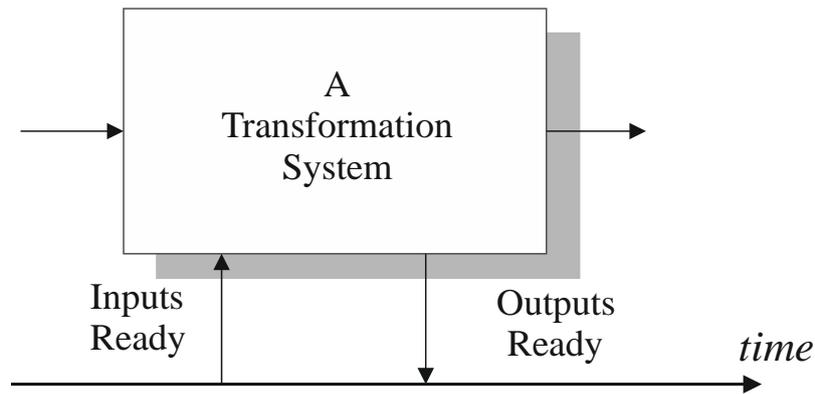


Рис. 11.1 – Трансформационная система

В противном случае систему относят к классу *управляемых событиями, или реактивных*. *Реактивная* – это такая динамическая система, которая воспринимает внешние дискретные воздействия и отвечает своими реакциями на эти воздействия (рис. 11.2). Причем реакции системы различны и сами зависят как от воздействий, так и от состояния, в котором система находится. *Основное отличие реактивных систем от трансформационных* – в принципиальной непредсказуемости моментов поступления тех или иных воздействий. Эта непредсказуемость – следствие изменчивости условий, в которых такие системы работают.

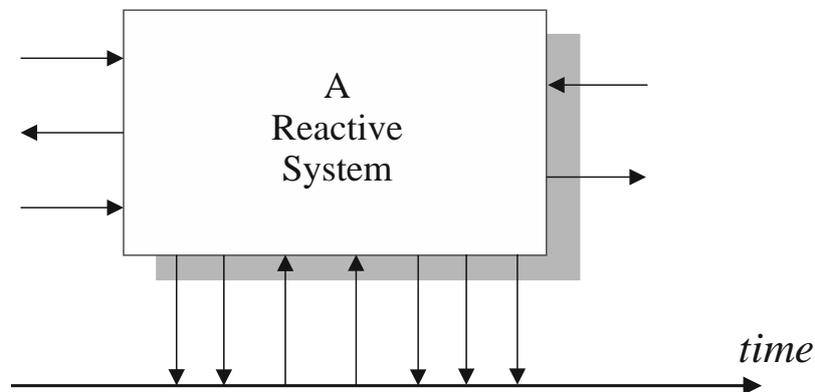


Рис. 11.2 – Реактивная система

Пример простой реактивной системы – контроллер светофора, управляемого пешеходами. Его входы никогда не приобретут законченного вида – сигналы на них поступают постоянно и в неизвестной заранее последовательности. Последние десятилетия характеризуются повсеместным распространением реактивных систем. Количество таких систем в мире превышает по некоторым оценкам число 10^{10} . Неудивительно, что в соответствии с изменением окружающего нас мира меняются и подходы к его анализу. Моделирование физики технологических процессов (непрерывная составляющая поведения системы) дополняется

моделированием логики работы управляющих ими устройств (дискретная компонента). *Математический аппарат описания* в данном случае – это система уравнений, но не дифференциальных, а дифференциально-алгебро-логических, для которых *отсутствуют стройная теория и единый подход*. Так же обстоит дело и с наглядностью. Визуализация протекания физических процессов обеспечивалась графиками изменения во времени тех или иных величин. Попытка такого графического представления процессов в реактивных системах может закончиться неудачно. *Основными причинами* этого являются многократное возрастание количества отображаемых величин и отсутствие на графиках информации о причинно-следственных связях между изменяющимися переменными состояниями.

В настоящее время для моделирования дискретной динамики реактивных систем широко используется предложенный Д. Харелом [2] *визуальный формализм* – Statechart (диаграммы состояний и переходов). Основные *неграфические компоненты* таких диаграмм – это событие и действие, основные *графические компоненты* – состояние и переход.

Событие – нечто, происходящее вне рассматриваемой системы, возможно требуя некоторых ответных действий. События могут быть вызваны поступлением некоторых данных или некоторых задающих сигналов со стороны человека или некоторой другой части системы. События считаются мгновенными (для выбранного уровня абстрагирования). Они образуют входной поток.

Действия – это реакции моделируемой системы на события. Подобно событиям действия принято считать мгновенными. Они образуют выходной поток.

Состояние – условия, в которых моделируемая система пребывает некоторое время, в течение которого она ведет себя одинаковым образом. В диаграмме переходов состояния представлены прямоугольными полями со скругленными углами.

Переход – изменение состояния, обычно вызываемое некоторым значительным событием. Как правило, состояние соответствует промежутку времени между двумя такими событиями. Переходы *показываются в диаграммах переходов линиями со стрелками*, указывающими направление перехода. Каждому переходу могут быть сопоставлены условия, при выполнении которых переход осуществляется. С каждым переходом и каждым состоянием могут быть соотнесены некоторые действия. Действия могут дополнительно обозначаться как действия,

выполняемые *однократно при входе* в состояние; действия, выполняемые *многokrатно внутри* некоторого состояния; действия, выполняемые *однократно при выходе* из состояния.

Рассмотрим алгоритм работы светофора. Кроме начального состояния в модель нужно ввести дополнительные состояния.

Начальное состояние (*go* – движение транспорта разрешено, горит зеленый), затем светофор переходит в состояние *attention* – горит мигающий зеленый. Затем светофор переходит в состояние *slow* – горит желтый. Остановка транспорта *stop* (запрет движения – горит красный) и *ready* (приготовиться к движению – горят красный и желтый). Состояние *attention* удобно представить гиперсостоянием с парой переключающихся элементарных событий: в одном горит зеленый (состояние А), в другом не горит – состояние В. Все они представлены на рисунке 11.3.

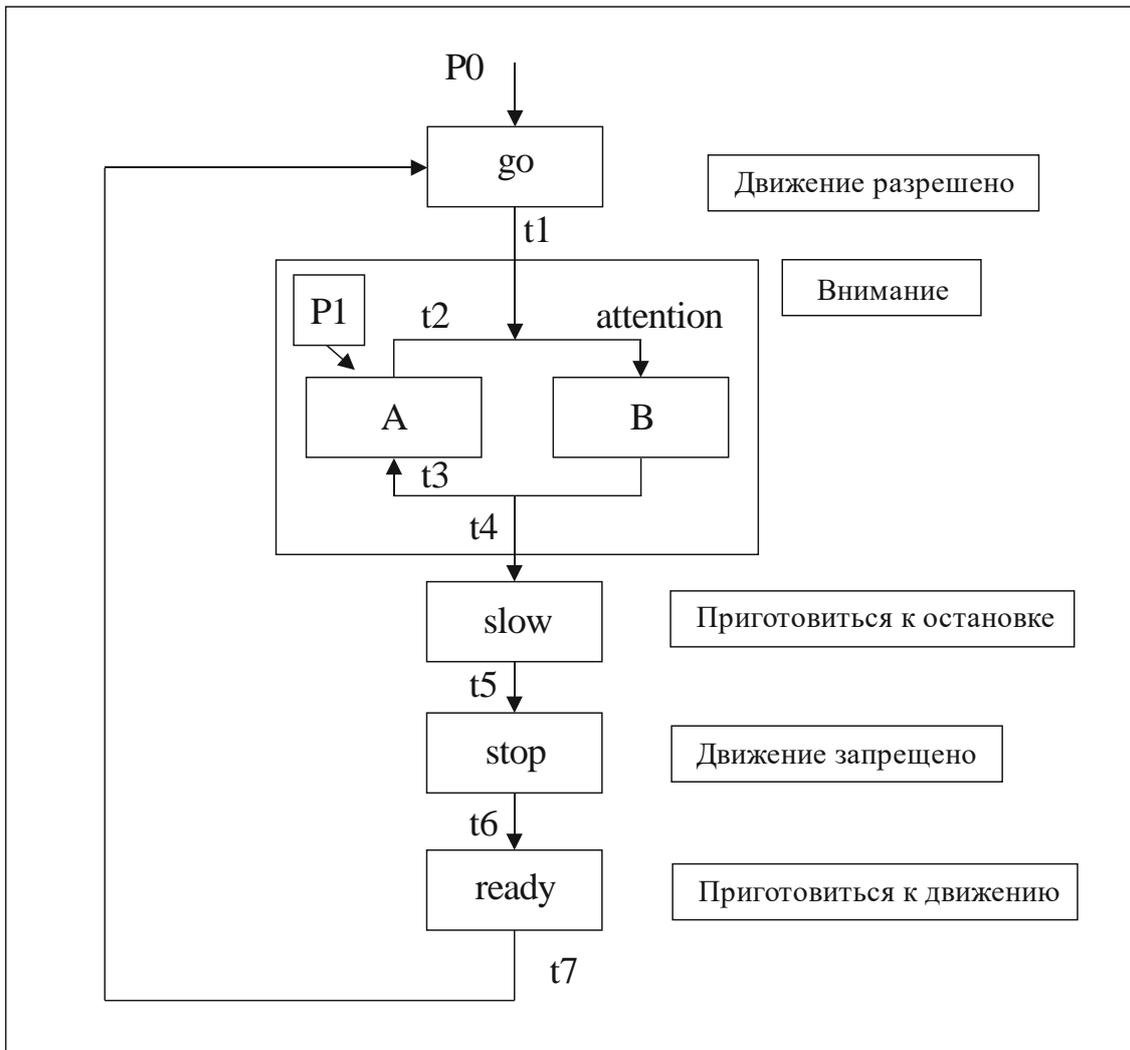


Рис. 11.3 – Стейтчарт светофора

Рассмотрим задание условий срабатывания переходов. Здесь они выполняются по таймауту, то есть по истечении интервала времени, который прошел от начала состояния.

Рассмотрим в качестве примера диаграмму состояний и переходов цифровых часов, представленную на рисунке 11.4.

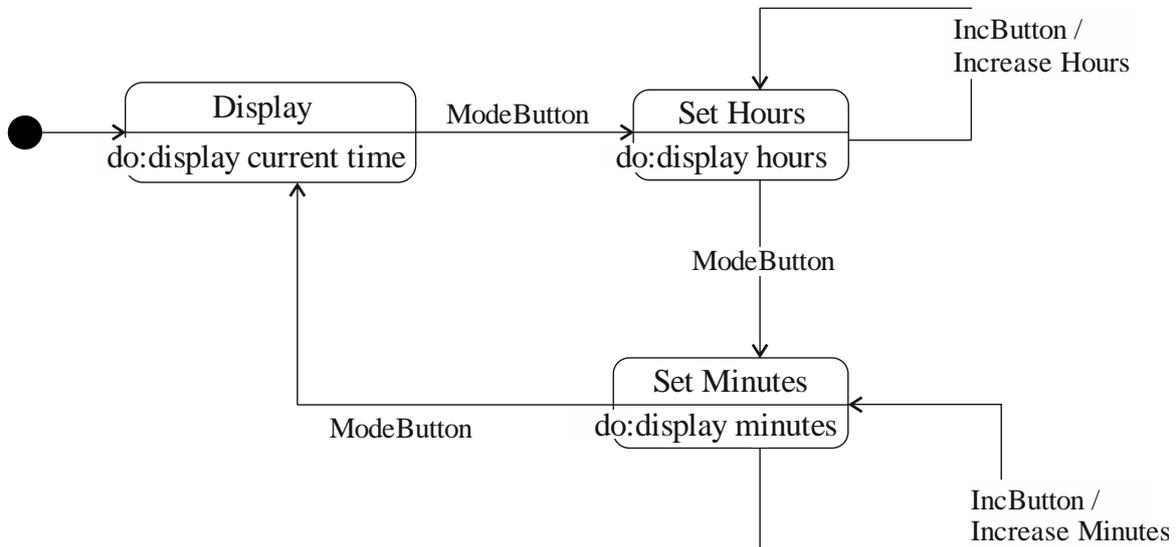


Рис. 11.4 – Диаграмма состояний и переходов цифровых часов

На часах имеются две кнопки: *ModeButton* и *IncButton* (кнопка образа действия и кнопка увеличения). Нажатие любой из них генерирует событие, которое может вызывать переход из одного состояния в другое. Имеются *три состояния*: *Display*, *Set Hours* and *Set Minutes* (дисплей, установка часов и установка минут). Состояние *дисплей* – начальное состояние (что обозначается стрелкой, направленной от блока перехода по умолчанию в виде черного круга). В состоянии *Set Hours* (установка часов) событие *ModeButton* вызывает переход к состоянию *Set Minutes* (установка минут), тогда как событие *IncButton* увеличивает текущее время (число часов), которое отображается на экране, причем это происходит без изменения состояния. Каждому состоянию часов соответствует действие, записанное ниже горизонтальной линии. Оно начинает выполняться после того, как переход в это состояние произошел.

Уже в таком виде диаграммы состояний и переходов являются достаточно выразительным языком описания реактивных систем. Для сравнения эффективности представления на языке диаграмм состояний и переходов с альтернативным (в виде автомата с конечным числом состояний) рассмотрим 10-битовый счетчик, подсчитывающий количество событий *a* и выдающий сообщение *overflow* (переполнение) после 1024-го события (рис. 11.5).

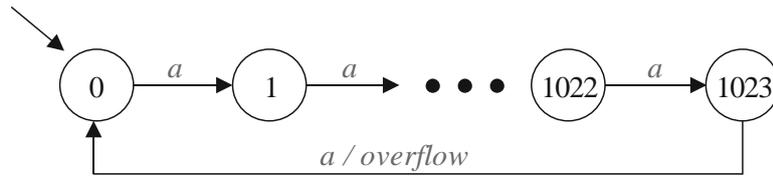
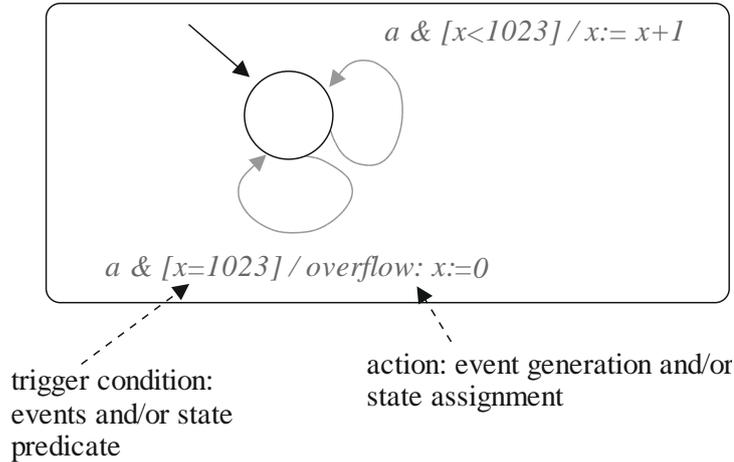
As FSM:**As Statechart:**

Рис. 11.5 – Альтернативные варианты описания 10-битового счетчика

Как видно из рисунка, представление с использованием языка диаграмм состояний и переходов Statechart компактнее примерно в 1000 раз. Однако при описании большой системы даже такое описание может оказаться слишком трудным для восприятия. Для предотвращения эффекта возрастания сложности при моделировании больших систем были предложены дальнейшие усовершенствования. Наряду с состояниями теперь могут использоваться гиперсостояния (суперсостояния), объединяющие несколько состояний, имеющих идентичную реакцию на одно и то же событие. При этом вместо изображения таких переходов в некоторое состояние из всех состояний, охватываемых гиперсостоянием, изображается только один переход из гиперсостояния в указанное состояние (обобщение переходов). Гиперсостояния теоретически могут иметь произвольную глубину вложения. Переходы из гиперсостояния связаны со всеми уровнями вложения. Гиперсостояния могут объединять ИЛИ-состояния (последовательные состояния) и И-состояния (параллельные состояния). В первом случае, перейдя в гиперсостояние, система может находиться только в одном из состояний. Во втором случае система, перейдя в гиперсостояние, будет пребывать одновременно в нескольких состояниях.

Диаграммы состояний и переходов в настоящее время широко используются для моделирования сложных систем. Достаточно упомянуть унифицированный язык моделирования (Unified Modeling Language (UML)), одним из элементов которого являются диаграммы состояний и язык Графсет, который используется при программировании логических контроллеров систем автоматизации.

Существенно повышает степень наглядности модели *использование имитации*, отображающей изменения в системе, сопровождающиеся переходами от одного состояния к другому. Построение таких имитационных моделей возможно с использованием программ Stateflow и Simulink, входящих в состав пакета MATLAB. MATLAB обеспечивает доступ к различным типам данных, высокоуровневому программированию и инструментальным средствам визуализации. Simulink поддерживает проектирование непрерывных и дискретных динамических систем в графической среде (в виде блок-схем). Stateflow-диаграммы, использующие визуальный формализм Д. Харела, включаются в Simulink-модели, чтобы обеспечить возможность моделирования процессов, управляемых событиями. Stateflow обеспечивает ясное описание поведения сложных систем, используя диаграммы состояний и переходов.

Комбинация MATLAB – Simulink – Stateflow является *мощным универсальным инструментом моделирования* реактивных систем. Дополнительная возможность следить в режиме реального времени за процессом выполнения диаграммы путем включения режима анимации делает процесс моделирования реактивных систем по-настоящему наглядным.

Stateflow – мощный графический инструмент проектирования и моделирования комплексных систем локального управления и супервизорного логического контроля. Используя *Stateflow*, можно:

- визуально моделировать комплексные реактивные системы;
- проектировать детерминированные системы супервизорного управления;
- легко изменять проект, оценивать результаты изменений и исследовать поведение системы на любой стадии проекта;
- автоматически генерировать программный код (целочисленный или с плавающей точкой) непосредственно по проекту (для этого требуется Stateflow Coder).

Преимущества интегрирования со средами MATLAB и Simulink в процессе моделирования и анализ Stateflow позволяют использовать диаграммы потоков (*flow diagram*) и диаграммы состояний и переходов (*state transition*) в одной диаграмме Stateflow. Система обозначений диаграммы потоков – логика, представленная без использования состояний. В некоторых случаях диаграммы потоков ближе логике системы, что позволяет избежать использования ненужных состояний. Система обозначений диаграммы потоков – эффективный способ представить общую структуру программного кода как конструкцию в виде условных операторов и циклов. Stateflow также обеспечивает ясное, краткое описание поведения комплексных систем, используя *теорию конечных автоматов, диаграммы потоков и диаграммы переходов состояний*. Stateflow делает описание системы (спецификацию) и проект ближе друг другу. Создавать проекты, рассматривая различные сценарии и выполняя итерации, намного проще, если при моделировании поведения системы используется Stateflow [3].

11.2 Stateflow 5 как программный продукт

В настоящее время фирмой MathWorks разработана уже пятая версия программы Stateflow. Для работы Stateflow 5 требует, чтобы были установлены следующие программные продукты:

- MATLAB[®] 6.5 (Release13);
- Simulink[®] 5;
- С или С++ compiler для генерации кода по Stateflow-диаграмме;
- Real-Time Workshop[®] Version 5 (Release 13) для генерации кода Simulink-элементов Stateflow-модели (рис. 11.6).

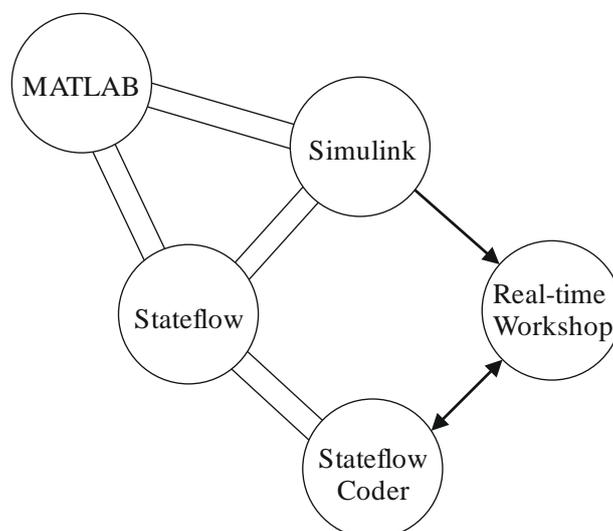


Рис. 11.6 – Программная среда Stateflow

Компоненты Stateflow

Stateflow состоит из следующих компонентов:

- stateflow graphics editor – графический редактор;
- stateflow Explorer – проводник;
- stateflow Coder – генератор объектного кода;
- stateflow Debugger – отладчик;
- stateflow Dynamic Checker – динамическое проверочное устройство.

Во время выполнения задачи обеспечивает проверку таких условий, как *заикливание и нарушение диапазона данных*.

11.3 Проектирование моделей реактивных систем

При проектировании моделей реактивных систем Stateflow используется вместе с Simulink и по желанию – с RTW (Real-Time Workshop – мастерская реального времени) под управлением MATLAB. Вы можете проектировать модель, начиная с Stateflow (управляющей) части, а позднее скомпоновать модель Simulink, можно также проектировать модель, начинающуюся с Simulink (алгоритмической) части, а позже прибавить диаграмму Stateflow. Можно также улучшить существующую модель Simulink, заменяя логические блоки Simulink на диаграммы Stateflow. Подход, который здесь используется, определяет, в какой последовательности вы разрабатываете различные части модели.

Совокупность всех Stateflow блоков в модели Simulink – это машина. При использовании Simulink вместе с Stateflow для моделирования Stateflow генерирует *S*-функцию (МEX-файл) для каждой Stateflow машины, чтобы поддерживать моделирование. Этот сгенерированный код для моделирования называется *sfun*-кодом Stateflow.

Если необходима работа модели в реальном масштабе времени, Stateflow Coder позволит сгенерировать целочисленный или с плавающей точкой код, основанный на Stateflow машине. RTW сгенерирует код для Simulink части модели и обеспечит структуру для выполнения сгенерированного кода Stateflow в реальном масштабе времени. Код, сгенерированный Stateflow Coder, включен в код, сгенерированный RTW. Вы можете получить код, сгенерированный обоими продуктами для специфической платформы. Этот сгенерированный код – специфический RTW код, который в пределах Stateflow называется *rtw кодом*.

Основные варианты использования:

- используют Stateflow вместе с Simulink для моделирования;

- используют Stateflow, Stateflow кодер, Simulink и мастерскую реального времени, чтобы генерировать целевой код для законченной модели;
- используют Stateflow и Stateflow кодер, чтобы генерировать целевой код для Stateflow машины.

Контрольные вопросы

1. Каким поведением характеризуются системы, моделируемые Stateflow?
2. Какие системы получили название гибридные системы?
3. Что такое гибридный автомат?
4. Когда говорят о трансформационных системах?
5. В чем основное отличие реактивных систем от трансформационных?
6. Каков математический аппарат описания реактивных систем?
7. Что такое событие и действие?
8. Дайте определение состоянию и переходу.
9. Определите состав программной среды Stateflow.

Литература

1. Карпов, Ю. Г. Имитационное моделирование систем: введение в моделирование с помощью AnyLogic 5 / Ю. Г. Карпов. – СПб. : БХВ-Петербург, 2005. – 390 с.
2. Рогачев, Г. Н. Stateflow 5. Руководство пользователя [Электронный ресурс]. – Режим доступа: <http://matlab.exponenta.ru/stateflow/book1> (дата обращения: 30.04.2020).
3. Новгородцев, А. Б. Расчет электрических цепей в MATLAB : учеб. курс / А. Б. Новгородцев. – СПб. : ПИТЕР, 2004. – 250 с.

12 Анализ динамических цепей

12.1 Уравнения состояния электрических цепей

Поведение электрической цепи описывается в общем случае системой дифференциальных уравнений. С помощью подстановок и замены переменных эту систему уравнений можно преобразовать к одному дифференциальному уравнению n -го порядка. Как правило, порядок уравнения равен суммарному числу индуктивных и емкостных элементов: $n = n_L + n_C$. Однако для нелинейной цепи решить дифференциальное уравнение n -го порядка трудно, а часто просто невозможно. Более рационально записывать систему дифференциальных уравнений в нормальной форме (форме Коши) [1].

Уравнения в форме Коши состоят из уравнений первого порядка, разрешенных относительно первой производной одной из переменных. В левой части каждого уравнения записывают производную, а в правой – функции переменных цепи и напряжений и токов независимых источников. Обозначим выбранные переменные x_1, x_2, \dots, x_n . Тогда система уравнений примет вид:

$$\begin{aligned} \frac{dx_1}{dt} &= f_1(x_1, x_2, \dots, x_n, \bar{u}), \\ \frac{dx_2}{dt} &= f_2(x_1, x_2, \dots, x_n, \bar{u}), \\ &L L L L L L L L L L \\ \frac{dx_n}{dt} &= f_n(x_1, x_2, \dots, x_n, \bar{u}). \end{aligned} \tag{12.1}$$

В системе уравнений (12.1) \bar{u} – вектор напряжений и токов независимых источников. Уравнения в форме (12.1) называют *уравнениями состояния*, а переменные x_1, x_2, \dots, x_n – *переменными состояния*. Для электрических цепей в качестве переменных состояния удобно выбирать напряжения емкостных элементов и токи индуктивных элементов, поскольку эти переменные определяют запас энергии в цепи.

Уравнения (12.1) получили название «*уравнения состояния*», потому что при известных начальных значениях переменных состояния $x_1(t_0), x_2(t_0), \dots, x_n(t_0)$ в некоторый момент времени t_0 мы можем определить эти переменные и в

момент $t > t_0$. Чтобы показать это, зададим малое приращение времени $\Delta t = h$. Тогда

$$x_1(t_0 + h) \approx x_1(t_0) + h \frac{dx_1(t_0)}{dt},$$

$$x_2(t_0 + h) \approx x_2(t_0) + h \frac{dx_2(t_0)}{dt},$$

L L L L L L L L L L

$$x_n(t_0 + h) \approx x_n(t_0) + h \frac{dx_n(t_0)}{dt}.$$

Далее мы можем определить значения переменных при $t = t_0 + 2h$, $t_0 + 3h$ и т. д. Таким образом, уравнения (12.1) позволяют определить состояние цепи в любой момент времени при известных начальных значениях переменных. Если известны значения переменных состояния, то легко могут быть найдены и остальные токи, и напряжения.

В качестве примера запишем уравнения состояния последовательной RLC -цепи (рис. 12.1).

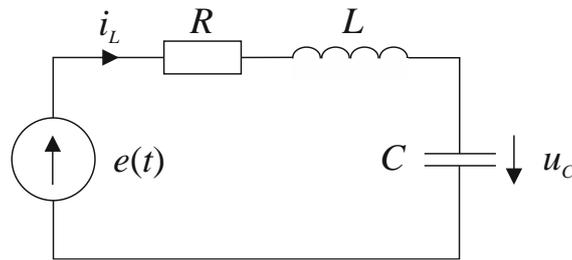


Рис. 12.1 – Последовательная RLC -цепь

В соответствии с вторым законом Кирхгофа

$$u_L(t) = -Ri_L(t) - u_C(t) + e(t). \quad (12.2)$$

Поскольку индуктивный и емкостной элементы соединены последовательно, их токи равны:

$$i_L(t) = i_C(t). \quad (12.3)$$

Разделим левую и правую части уравнения (12.2) на L , а левую и правую части уравнения (12.3) – на C . Учитывая, что $u_L = L \frac{di_L}{dt}$, $i_C = C \frac{du_C}{dt}$, получим уравнения состояния:

$$\frac{di_L}{dt} = -\frac{R}{L}i_L(t) - \frac{1}{L}u_C(t) + \frac{1}{L}e(t),$$

$$\frac{du_C}{dt} = \frac{1}{C} i_L(t).$$

Уравнения состояния удобно записывать в матричной форме:

$$\begin{bmatrix} \dot{X}(t) \end{bmatrix} = [A] \begin{bmatrix} X(t) \end{bmatrix} + [B] \begin{bmatrix} U(t) \end{bmatrix}.$$

Здесь $[X]$ – вектор переменных состояния, $\begin{bmatrix} \dot{X} \end{bmatrix}$ – вектор производных, $\begin{bmatrix} U(t) \end{bmatrix}$ – вектор входных воздействий.

Квадратную матрицу $[A]$ называют *матрицей параметров схемы*, а $[B]$ – *матрицей параметров входного воздействия*. Для рассматриваемого примера

$$[A] = \begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix},$$

$$[B] = \begin{bmatrix} 1/L \\ 0 \end{bmatrix}.$$

Зная состояние цепи $\begin{bmatrix} X(t) \end{bmatrix}$ и вектор входных воздействий $\begin{bmatrix} U(t) \end{bmatrix}$, реакцию цепи $\begin{bmatrix} Y(t) \end{bmatrix}$ (токи и напряжения ветвей) можно найти как линейную комбинацию вектора переменных состояния и вектора входных воздействий:

$$\begin{bmatrix} Y(t) \end{bmatrix} = [C] \begin{bmatrix} X(t) \end{bmatrix} + [D] \begin{bmatrix} U(t) \end{bmatrix}.$$

Матрицы $[C]$ и $[D]$ зависят от конфигурации и параметров цепи. Например, если компонентами вектора $\begin{bmatrix} Y(t) \end{bmatrix}$ являются напряжения u_L и u_R , они находятся с помощью уравнений:

$$u_L(t) = -Ri_L(t) - u_C(t) + e(t);$$

$$u_R(t) = Ri_L(t).$$

В матричной форме:

$$\begin{bmatrix} u_L(t) \\ u_R(t) \end{bmatrix} = \begin{bmatrix} -R & -1 \\ R & 0 \end{bmatrix} \begin{bmatrix} i_L(t) \\ u_C(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} e(t).$$

Существует несколько причин, по которым дифференциальные уравнения электрической цепи целесообразно представлять в форме *уравнений состояния*. Во-первых, уравнения состояния можно записать как для линейных, так и для нелинейных цепей. Во-вторых, не всегда можно получить аналитическое решение нелинейного дифференциального уравнения, а численные методы решения ориентированы на уравнения, записанные в нормальной форме (12.1). Наконец, в-третьих, матричная форма уравнений состояния не зависит от порядка цепи.

12.2 Алгоритм формирования уравнений состояния

В настоящее время разработаны эффективные компьютерные алгоритмы формирования уравнений состояния цепей любой сложности на ЭВМ. Однако в современных программах компьютерного моделирования электронных цепей такие алгоритмы не используются. Поэтому рассматривать их мы не будем. Познакомиться с этими алгоритмами можно в классическом учебнике по машинному анализу электронных схем [2].

Рассмотрим простой алгоритм формирования уравнения состояния. Он пригодится нам при рассмотрении методов численного интегрирования уравнений состояния.

Алгоритм включает следующие шаги:

- 1) анализируем цепь в начальный момент времени (при $t = 0_-$) и определяем напряжения емкостных элементов $u_C(0)$ и токи индуктивных элементов $i_L(0)$;
- 2) выделяем в анализируемой цепи индуктивные и емкостные элементы (рис. 12.2, а). Заменяем емкостные элементы источниками напряжения $u_C(t)$, а индуктивные – источниками тока $i_L(t)$ (рис. 12.2, б);

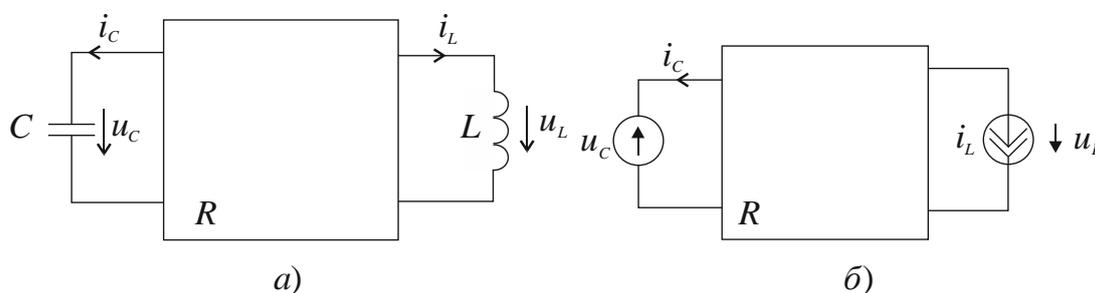


Рис. 12.2 – Эквивалентные представления для формирования уравнений

- 3) для полученной резистивной схемы замещения составляем уравнения на основании законов Кирхгофа либо с помощью любого другого метода анализа резистивных цепей;
- 4) из полученной системы уравнений путем подстановок и исключения переменных выражаем напряжения индуктивных и токи емкостных элементов как функции переменных состояния – токов индуктивностей и напряжений емкостей, а также напряжений и токов независимых источников:

$$\begin{aligned} u_L &= f_1(i_L, u_c, \bar{u}) \\ i_C &= f_2(i_L, u_c, \bar{u}) \end{aligned}$$

- 5) делим левую и правую части уравнения для u_L на L , а левую и правую части уравнения для i_C – на C . Учитывая, что $u_L = L \frac{di_L}{dt}$, $i_C = C \frac{du_C}{dt}$, получим уравнения состояния:

$$\frac{di_L}{dt} = \frac{1}{L} f_1(i_L, u_C, \bar{u});$$

$$\frac{du_C}{dt} = \frac{1}{C} f_2(i_L, u_C, \bar{u}).$$

Рассмотрим примеры составления уравнений состояния с помощью рассмотренного алгоритма.

Пример 12.1. Составить уравнения состояния для RLC -цепи второго порядка, показанной на рисунке 12.3.

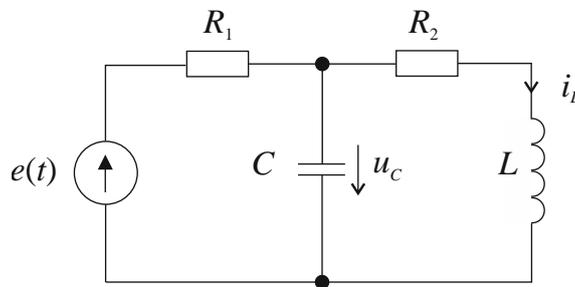


Рис. 12.3 – RLC -цепь второго порядка

Решение. Составим эквивалентную резистивную схему, заменив индуктивный элемент источником тока, а емкостный – источником напряжения (рис. 12.4). В этой схеме ток источника тока изменяется по закону $i_L(t)$, а ЭДС источника напряжения – по закону $u_C(t)$.

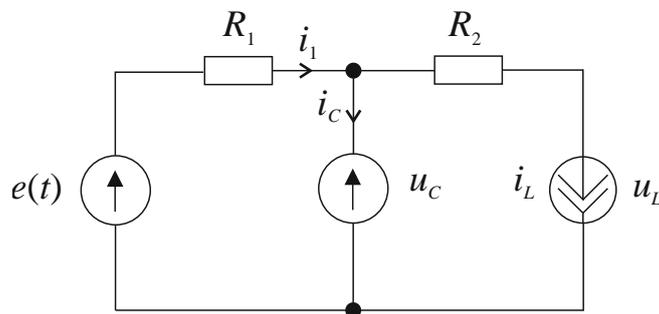


Рис. 12.4 – Эквивалентная резистивная схема

Для схемы на рисунке 12.4 запишем уравнения по законам Кирхгофа:

$$-i_1 + i_C + i_L = 0;$$

$$R_1 i_1 = e(t) - u_C;$$

$$R_2 i_L + u_L = u_C.$$

Решив эту систему уравнений относительно i_C и u_L , получим:

$$i_C = -\frac{1}{R_1} u_C - i_L + \frac{1}{R_1} e(t);$$

$$u_L = u_C - R_2 i_L.$$

Разделив левую и правую части последних равенств соответственно на C и L , получим уравнения состояния:

$$\frac{du_C}{dt} = -\frac{1}{R_1 C} u_C - \frac{1}{C} i_L + \frac{1}{R_1 C} e(t);$$

$$\frac{di_L}{dt} = \frac{1}{L} u_C - \frac{R_2}{L} i_L.$$

В матричной форме уравнения состояния анализируемой цепи имеют вид:

$$\begin{bmatrix} \frac{du_C}{dt} \\ \frac{di_L}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1 C} & -\frac{1}{C} \\ \frac{1}{L} & -\frac{R_2}{L} \end{bmatrix} \begin{bmatrix} u_C \\ i_L \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1 C} \\ 0 \end{bmatrix} e(t).$$

Пример 12.2. Записать уравнения состояния активной RC - цепи, показанной на рисунке 12.5. Операционный усилитель (ОУ) работает в линейном режиме.

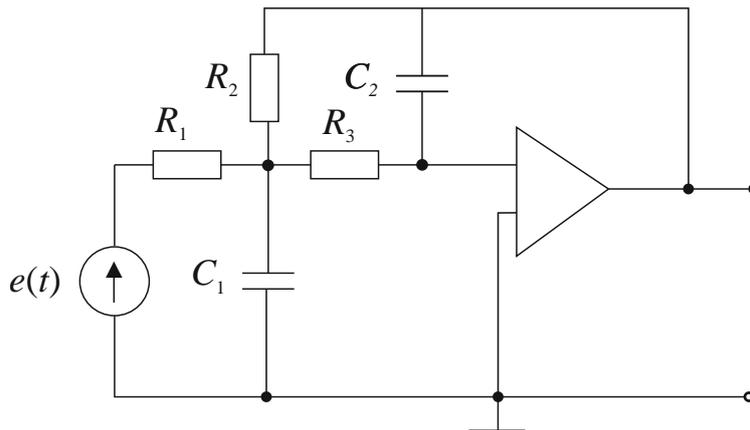


Рис. 12.5 – Операционный усилитель в линейном режиме

Решение. Составим эквивалентную схему, заменив емкостные элементы источниками напряжения. Запишем систему уравнений по законам Кирхгофа для цепи на рисунке 12.6.

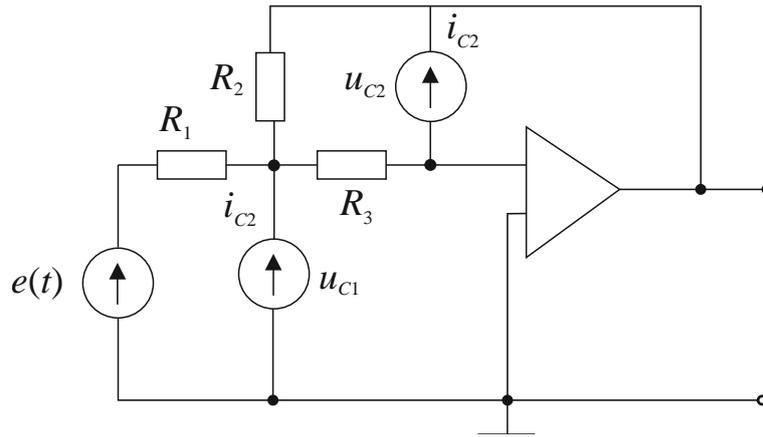


Рис. 12.6 – Эквивалентная схема ОУ

Для узла, в котором сходятся резисторы R_1, R_2, R_3 и конденсатор:

$$-i_1 - i_2 - i_3 + i_{C1} = 0. \quad (12.4, a)$$

Для узла, к которому подключен инвертирующий вход ОУ:

$$i_3 - i_{C2} = 0. \quad (12.4, б)$$

Для контура, включающего источник напряжения $e(t)$, резистор R_1 и конденсатор C_1 :

$$R_1 i_1 = e(t) - u_{C1}. \quad (12.4, в)$$

Для контура, образованного резисторами R_2, R_3 и конденсатором:

$$-R_2 i_2 + R_3 i_3 = -u_{C2}. \quad (12.4, г)$$

Для контура, включающего резистор R_3 и вход ОУ:

$$-R_3 i_3 = u_{C1}. \quad (12.4, д)$$

При составлении уравнений (12.4) предполагалось, что входные токи ОУ и дифференциальное напряжение на входе равны нулю: $u_d = 0, i_- = 0$.

Решив систему уравнений (12.4), получим:

$$i_{C1} = -\frac{R_1 R_2 + R_1 R_3 + R_2 R_3}{R_1 R_2 R_3} u_{C1} + \frac{1}{R_2} u_{C2} + \frac{1}{R_1} E; \quad (12.5, a)$$

$$i_{C2} = -\frac{1}{R_3} u_{C1}. \quad (12.5, б)$$

Разделив левую и правую части (12.5, а, б) на C_1 и C_2 соответственно, получим уравнения состояния анализируемой цепи:

$$\frac{du_{C1}}{dt} = -\frac{R_1R_2 + R_1R_3 + R_2R_3}{C_1R_1R_2R_3}u_{C1} + \frac{1}{C_1R_2}u_{C2} + \frac{1}{C_1R_1}e(t);$$

$$\frac{du_{C2}}{dt} = -\frac{1}{C_2R_3}u_{C1}.$$

В матричной форме:

$$\begin{bmatrix} \frac{du_{C1}}{dt} \\ \frac{du_{C2}}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_1R_2 + R_1R_3 + R_2R_3}{C_1R_1R_2R_3} & \frac{1}{C_1R_2} \\ -\frac{1}{C_2R_3} & 0 \end{bmatrix} \begin{bmatrix} u_{C1} \\ u_{C2} \end{bmatrix} + \begin{bmatrix} \frac{1}{C_1R_1} \\ 0 \end{bmatrix} E.$$

Рассмотренный алгоритм позволяет формировать уравнения состояния простых цепей вручную. В настоящее время разработаны эффективные алгоритмы формирования уравнений состояния цепей любой сложности на ЭВМ.

Выводы

1. Поведение электрической цепи в общем случае описывается системой дифференциальных уравнений либо одним дифференциальным уравнением n -го порядка. Порядок уравнения равен суммарному числу индуктивных и емкостных элементов: $n = n_L + n_C$.
2. Наиболее рационально записывать уравнения цепи в виде системы дифференциальных уравнений первого порядка, разрешенных относительно первой производной одной из переменных. Такую систему называют *уравнениями состояния*.
3. Если корни характеристического уравнения цепи второго порядка комплексно-сопряженные, переходный процесс в цепи имеет колебательный характер.

12.3 Методы дискретного отображения.

Прямое Z-преобразование цифровых сигналов

На рисунке 12.7, *а* изображен некоторый аналоговый сигнал, а на рисунке 12.7, *б* – соответствующий ему цифровой сигнал, заданный последовательностью ординат $x(nT)$, или, проще, $x(n)$:

$$x(n) = \dots x(-1), \dots x(0), \dots x(1), \dots x(2), \dots \quad (12.6)$$

Эту последовательность в общем виде можно представить суммой:

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k). \quad (12.7)$$

Здесь $\delta(n-k)$ – единичный импульс (площадь = 1) при $n=k$; $\delta(n-k)=1$ при $n=k$; $\delta(n-k)=0$ при $n \neq k$. Это функция Кронекера, она реализуется с помощью АЦП-микросхем.

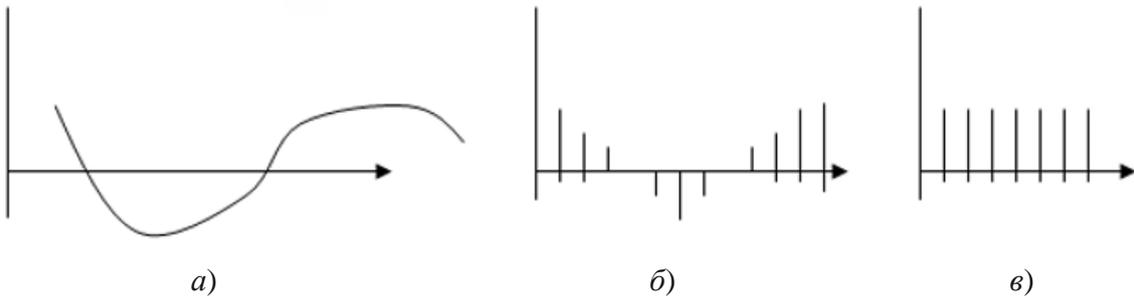


Рис. 12.7 – Аналоговое и цифровое представление сигнала:
 а) кривая сигнала; б) последовательность $x(n)$; в) последовательность $u(n)$

Взяв прямое преобразование Лапласа

$$X(p) = \int_0^{\infty} x(t)e^{-pt} dt,$$

перейдем к прямому z -преобразованию:

$$X(z) = \sum_n x(n)z^{-n}. \quad (12.8)$$

Суммирование начинают с $n=0$, полагая, что $x=0$ при $n < 0$.

Последовательность единичных импульсов, возникающих при $n=0, 1, 2, \dots$ (рис. 12.7, в) принято обозначать $u(n)$. При $n \geq 0$, $u(n)=1$.

Степенной ряд $x(n) = a^n$ при $n \geq 0$ a^0, a^1, a^2, \dots с помощью функции $u(n)$ может быть записан так: $x(n) = a^n u(n)$. Линейно нарастающую функцию $x(n) = 0, 1, 2, 3$ представим так: $x(n) = nu(n)$.

Решение дифференциальных уравнений путем сведения их к разностным

Линейному дифференциальному уравнению с постоянными коэффициентами может быть поставлено в соответствие *разностное уравнение*.

Первая производная по времени от непрерывной функции $y(t)$ может быть аппроксимирована конечной разностью:

$$\frac{dy(t)}{dt} = \frac{y(n-1) - y(n-2)}{T}.$$

Вторая производная

$$\frac{dy^2(t)}{dt^2} = \frac{y(n) - 2y(n-1) + y(n-2)}{T^2}.$$

В качестве примера рассмотрим уравнение

$$\frac{dy^2}{dt^2} + 2\frac{dy(t)}{dt} + 3y(t) = mx(t)$$

с начальными условиями $y(0) = 0$ и составим соответствующее ему разностное уравнение:

$$\frac{y(n) - 2y(n-1) + y(n-2)}{T^2} + 2\frac{y(n-1) - y(n-2)}{T} + 3y(n) = mx(n)$$

или

$$y(n)[1 + 2T + 3T^2] - y(n-1)[2 + 2T] + y(n-2) = Tmx(n). \quad (12.9)$$

По уравнению (12.9) можно последовательно определить значения $y(n)$, придавая n значения 1, 2, 3... и учитывать при этом, что $y(-1) = y(-2) = 0$, а $x(0)$, $x(1)$, $x(2)$ известны.

Выводы

1. Поведение электрической цепи описывается в общем случае системой дифференциальных уравнений либо одним дифференциальным уравнением n -го порядка. Порядок уравнения равен суммарному числу индуктивных и емкостных элементов: $n = n_L + n_C$.
2. Наиболее рационально записывать уравнения цепи в виде системы дифференциальных уравнений первого порядка, разрешенных относительно первой производной одной из переменных. Такую систему называют *уравнениями состояния*.
3. Если корни характеристического уравнения цепи второго порядка комплексно-сопряженные, переходный процесс в цепи имеет колебательный характер.

Контрольные вопросы

1. Чему равен порядок исходного дифуравнения?
2. Какие переменные называются переменными состояния?
3. Какие матрицы называются матрицей параметров схемы и матрицей параметров входного воздействия?
4. Какие замены элементов исходной схемы надо произвести для составления эквивалентной резистивной схемы?
5. Как задается цифровой сигнал?

6. Как аппроксимируется производная по времени от непрерывной функции?

Литература

1. Бенькович, Е. Практическое моделирование динамических систем / Е. Бенькович, Ю. Колесов, Ю. Сениченков. – СПб. : БХВ-Петербург, 2002. – 464 с.
2. Востриков, А. С. Теория автоматического регулирования : учеб. пособие / А. С. Востриков, Г. А. Французова. – Новосибирск : Изд-во НГТУ, 2006. – 368 с.

13 Анализ нелинейных систем

13.1 Основные понятия

Одна из *важнейших задач анализа нелинейных цепей* заключается в определении рабочих точек или точек смещения электронных приборов. *Рабочей точкой* нелинейного элемента называют значения постоянных напряжения и тока, изображаемых в виде точки на его вольт-амперной характеристике (ВАХ).

Рабочую точку определяют при действии постоянных источников напряжения и тока. При этом индуктивные элементы заменяют короткими замыканиями, а емкостные – разрывом.

Простейшим является графический метод определения рабочей точки. В соответствии с этим методом рабочая точка находится на пересечении ВАХ нелинейной подсхемы и нагрузочной прямой линейной цепи.

Рассмотрим графический метод определения рабочей точки. Выделим в анализируемой цепи две подсхемы: линейную N_a и нелинейную N_b (рис. 13.1).

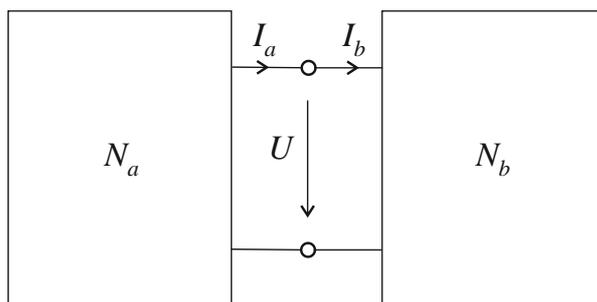


Рис. 13.1 – Линейная и нелинейная подсхемы

Для выбранных направлений напряжений и токов справедливы равенства:

$$I_a = I_b = I; \quad U_a = U_b = U. \quad (13.1)$$

Представим линейную часть цепи по схеме эквивалентного генератора (рис. 13.2).

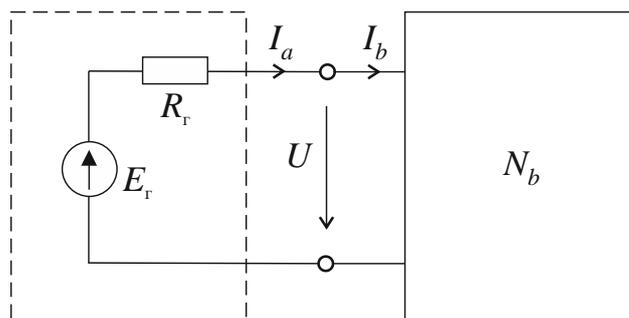


Рис. 13.2 – Схема для определения рабочей точки

ВАХ линейной подсистемы представляет прямую, проходящую через точки на осях напряжений и токов, соответствующие режимам холостого хода и короткого замыкания (рис. 13.3). Ее называют *нагрузочной прямой*.

В соответствии с равенствами (13.1) рабочая точка находится на пересечении ВАХ нелинейной подсистемы N_b и нагрузочной прямой (точка a на рисунке 13.3).

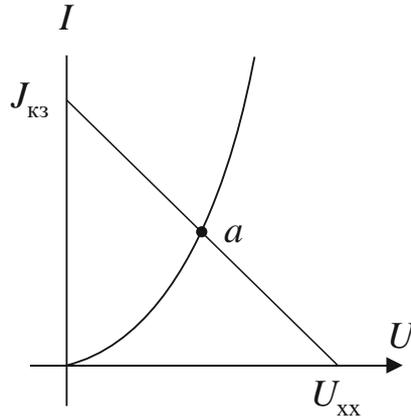


Рис. 13.3 – Иллюстрация рабочей точки

Графический метод определения рабочей точки является приближенным и применяется на практике только для цепей простой конфигурации. Его *достоинствами* являются простота и наглядность. Графический метод можно использовать для определения начального приближения, а затем для получения точного решения применить численный метод.

Пример 13.1. Рассмотрим пример определения рабочей точки нелинейного элемента. Цепь, показанная на рисунке 13.4, имеет параметры: $E = 2\text{В}$, $R = 25\text{ Ом}$. ВАХ нелинейного элемента: $I(U) = 0.04 U^2$.

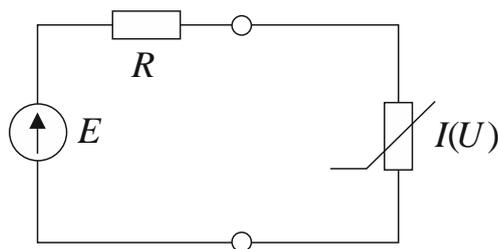


Рис. 13.4 – Схема с нелинейным элементом

Напряжение холостого хода линейного двухполюсника $U_{\text{хх}} = 2\text{В}$. Ток короткого замыкания $I_{\text{кз}} = \frac{E}{R} = 0.08\text{А}$. Нагрузочная характеристика цепи смещения – прямая, проходящая через точки с координатами $(0, 0.08)$ и $(2, 0)$ на рисунке 13.5.

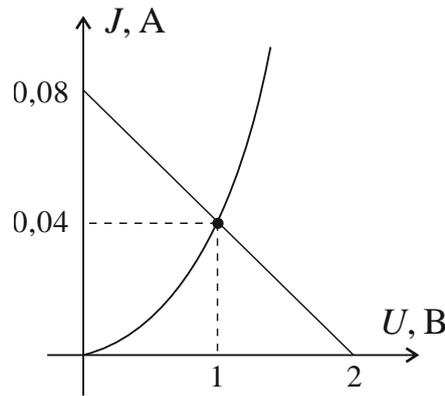


Рис. 13.5 – График с рабочей точкой

Рабочая точка находится на пересечении нагрузочной прямой и ВАХ нелинейного резистора. Ее координаты $(1; 0,04)$.

13.2 Численное решение уравнений нелинейных резистивных цепей. Метод Ньютона – Рафсона

Наиболее распространенным методом решения уравнений, описывающих поведение нелинейных резистивных цепей, является *метод Ньютона – Рафсона*. Это итерационный метод решения нелинейных уравнений, записанных в виде

$$f(x) = 0. \quad (13.2)$$

Предположим, что x_k является приближенным значением корня уравнения (13.2). В окрестности x_{k+1} функцию $f(x)$ можно разложить в ряд Тейлора:

$$f(x_k + h) = f(x_k) + hf'(x_k) + 0,5h^2 f''(x_k) + \dots \quad (13.3)$$

Символом ' в последней формуле обозначена операция дифференцирования. Отбросим в (13.3) нелинейные слагаемые. Полагая, что уточненное значение корня $x_{k+1} = x_k + h$ является *истинным*, найдем значение поправки:

$$h = -f(x_k) / f'(x_k).$$

Решение уравнения (13.2) начинают с того, что определяют начальное приближение корня x_0 . Уточненное значение x_{k+1} , $k = 1, 2, K$ определяется по формуле

$$x_{k+1} = x_k - f(x_k) / f'(x_k). \quad (13.4)$$

Расчет повторяют до тех пор, пока поправка не станет меньше заданной постоянной ε :

$$|x_{k+1} - x_k| < \varepsilon.$$

Равенство (13.4) называют *уравнением Ньютона – Рафсона*.

Метод Ньютона – Рафсона имеет быструю сходимость в том случае, если начальное приближение выбрано достаточно близко от истинного решения. Однако если начальное приближение далеко от точного решения, итерации Ньютона – Рафсона могут не сходиться совсем. Подробно вопросы обеспечения сходимости и численной устойчивости при анализе нелинейных цепей рассмотрены в [1, 2].

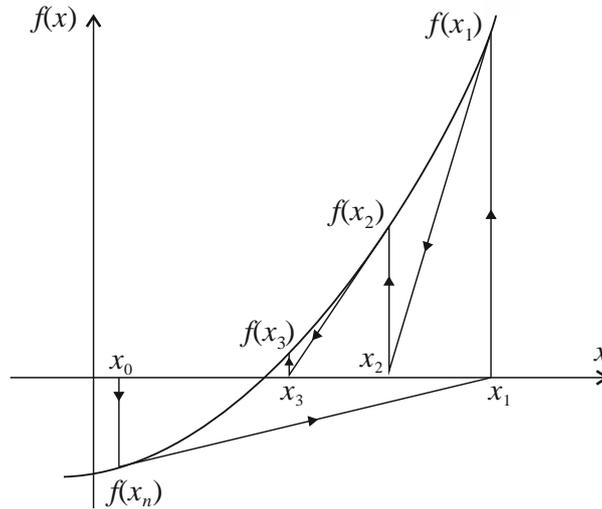


Рис. 13.6 – Графическая интерпретация метода

Метод Ньютона – Рафсона имеет простую геометрическую интерпретацию. Обратимся к рисунку 13.6, на котором изображен график функции $f(x)$. Отбрасыванию нелинейных слагаемых в формуле (13.3) соответствует замена функции $f(x)$ касательной в точке с координатами $\{x_1, f(x_1)\}$. Проведя касательную до пересечения с осью x , найдем поправку h и уточненное значение корня x_2 (рис. 13.6).

Пример 13.1. Расчет нелинейной резистивной цепи методом Ньютона – Рафсона. Цепь, показанная на рисунке 13.7, имеет следующие параметры: $J = 80\text{ мА}$, $G = 0.04\text{ Сим}$. Вольт-амперная характеристика нелинейного резистора описывается выражением $I(U) = 0.04U^2$. Необходимо определить рабочую точку нелинейного элемента.

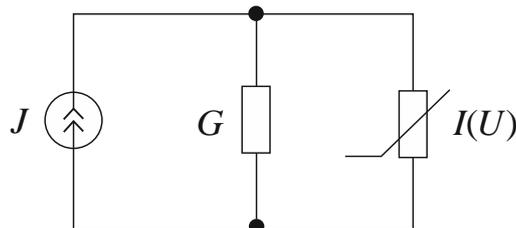


Рис. 13.7 – Схема для определения рабочей точки

Решение. В соответствии с первым законом Кирхгофа

$$-J + GU + 0.04U^2 = 0.$$

Итак, нелинейное уравнение имеет вид:

$$f(U) = -0.08 + 0.04U + 0.04U^2 = 0.$$

Производная определяется уравнением:

$$f'(U) = 0.04 + 0.08U.$$

Выберем начальное приближение $U_0 = 0$. Уточненное решение на первом шаге:

$$U_1 = U_0 - \frac{f(U_0)}{f'(U_0)} = 0 + \frac{0.08}{0.04} = 2.$$

На втором шаге:

$$U_2 = U_1 - \frac{f(U_1)}{f'(U_1)} = 2 - \frac{0.16}{0.2} = 1.2.$$

На третьем и четвертом шагах получим: $U_3 = 1.012$, $U_4 = 1$. Таким образом, напряжение нелинейного элемента $U = 1$ В.

Метод Ньютона – Рафсона можно использовать и для решения систем нелинейных уравнений. Обозначим как $[x^k]$ вектор переменных на k -м шаге. В этом случае уравнение (13.3) примет вид:

$$[x^{(k+1)}] = [x^k] - [J(x^k)]^{-1} [f(x^k)]. \quad (13.5)$$

Здесь $[J(x)]$ – матрица Якоби, элементы которой равны частным производным df_i / dx_j .

Равенство (13.5) называют *n -мерным аналогом уравнения Ньютона – Рафсона*.

Обращение матрицы Якоби в формуле (13.4) требует значительного количества операций. С вычислительной точки зрения более эффективно решить систему линейных уравнений:

$$[J(x^k)][x^{(k+1)}] = [J(x^k)][x^{(x)}] - [f(x^k)]. \quad (13.6)$$

Как и в одномерном случае, в n -мерном случае метод Ньютона – Рафсона обеспечивает быструю сходимость в тех случаях, когда начальное приближение выбрано достаточно близко к истинному решению. Однако объем вычислений при решении многомерной задачи резко возрастает, поскольку на каждом шаге требуется *вычислять n^2 производных* в матрице Якоби.

Рассмотрим основные *этапы процедуры решения системы нелинейных уравнений методом Ньютона – Рафсона*.

1. Выбираем максимальное число итераций m и допустимую величину ошибки ε .
2. Выбираем начальное приближение $[x^{(0)}]$.
3. Решаем уравнение (13.6) и находим уточненное решение $[x^{(1)}]$.
4. Если $\| [x^{(k+1)}] - [x^{(k)}] \| = \sqrt{\sum_{i=1}^N (x_i^{k+1} - x_i^k)^2} < \varepsilon$, решение найдено. Если неравенство не выполняется, повторяем шаг 3, используя уточненное значение вектора переменных.
5. Если число итераций превысило m и решение не найдено, выбирается новое начальное приближение и процедура решения повторяется.

13.3 Дискретные схемы замещения нелинейных резистивных цепей

При использовании метода Ньютона – Рафсона для решения системы нелинейных уравнений на каждой итерации необходимо вычислять частные производные $df_i(x^k)/dx_j$ для получения матрицы Якоби $[J(x^k)]$, что приводит к большим затратам машинного времени. Эту трудность можно преодолеть, если на каждой итерации нелинейные резисторы представлять *эквивалентными линеаризованными схемами*.

Рассмотрим нелинейную цепь, изображенную на рисунке 13.8, *а*. Вольт-амперная характеристика нелинейного резистора показана на рисунке 13.8, *в*.

Поскольку зажимы нелинейного элемента замкнуты накоротко, уравнение нелинейной цепи:

$$U(I) = 0. \quad (13.7)$$

Обозначим начальное приближение $I^{(0)}$. В соответствии с формулой Ньютона – Рафсона

$$I^{(j+1)} = I^{(j)} - (R^{(j)})^{-1} U(I^{(j)}), \quad j = 0, 1, 2, \dots \quad (13.8)$$

Здесь

$$R^{(j)} = \left. \frac{dU(I)}{dI} \right|_{I=I^{(j)}}.$$

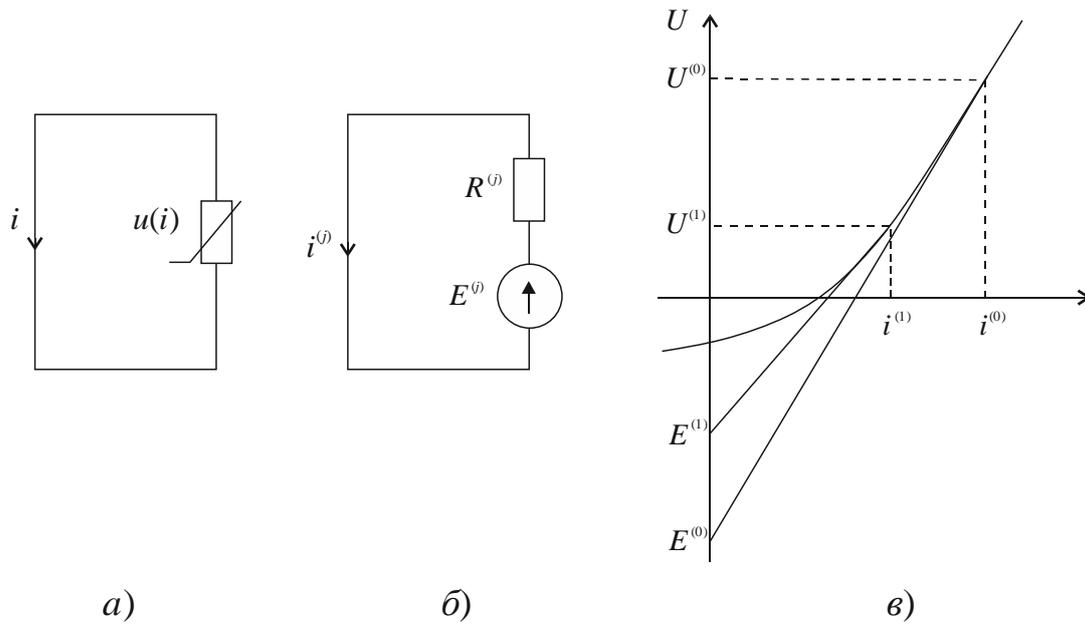


Рис. 13.8 – Эквивалентная схема и графическая иллюстрация

Уравнению (13.5) соответствует замена ВАХ нелинейного элемента касательной в точке с координатами $(I^{(j)}, U^{(j)})$. Уравнение касательной:

$$U(I) = R^{(j)}I^{(j)} - E^{(j)} = 0. \quad (13.9)$$

Величина $E^{(j)}$ равна отрезку, отсекаемому касательной на оси напряжений. Уравнению (13.9) соответствует линейная схема, изображенная на рисунке 13.8, б.

Найдем решение уравнения (13.7), используя метод Ньютона – Рафсона:

$$I^{(j+1)} = I^{(j)} - [R^{(j)}]^{-1} U(I^{(j)}). \quad (13.10)$$

При $I = I^{(j)}$ выполняется равенство $U(I) = U(I^{(j)})$. Следовательно, если ток принимает значение $I = I^{(j)}$, уравнения (13.9) и (13.10) имеют одинаковые решения. Поэтому на j -й итерации нелинейное уравнение (13.10) можно заменить линейным соотношением (13.9) и получить такое же решение $I^{(j+1)}$. Иными словами, на j -й итерации нелинейный элемент заменяется линейной эквивалентной схемой, изображенной на рисунке 13.8, б. Подчеркнем, что линейная схема эквивалентна нелинейному элементу только на j -й итерации. На каждой следующей итерации параметры $R^{(j)}$ и $E^{(j)}$ рассчитываются заново в соответствии с новым значением тока $I^{(j+1)}$.

Если цепь содержит несколько нелинейных элементов, каждый из них можно заменить линейной эквивалентной схемой. Таким образом, решение системы нелинейных уравнений методом Ньютона – Рафсона эквивалентно анализу последовательности линейных схем замещения. На всех итерациях структура линейной цепи остается неизменной, а ее параметры меняются дискретно. Поэтому такие эквивалентные схемы получили название *дискретных линейных схем замещения*.

Достоинство дискретных линейных схем замещения заключается в том, что процедура определения рабочей точки сводится к анализу линейных резистивных схем.

Пример 13.2. Построить линейную схему замещения полупроводникового диода, ВАХ которого описывается выражением:

$$I_D = I_0 \left(e^{U_D/V_t} - 1 \right).$$

Здесь I_0 – обратный ток насыщения диода; V_t – температурная постоянная.

Решение. Вольт-амперная характеристика диода показана на рисунке 13.9.

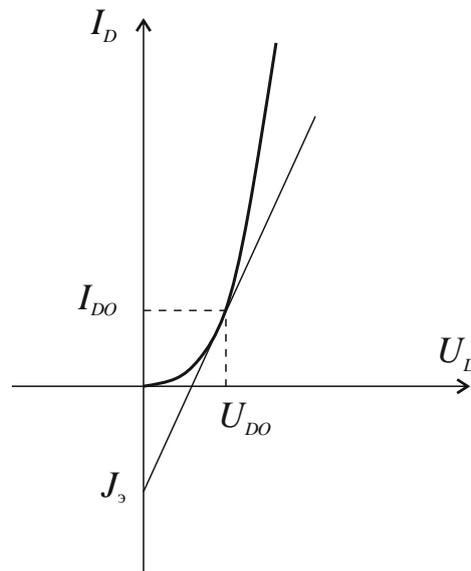


Рис. 13.9 – Вольт-амперная характеристика диода

На рисунке 13.10 показана линейная схема замещения.

Параметры линейной схемы:

$$G_3 = \frac{dI_{D0}}{dU} = \frac{I_0}{V_t} e^{U_{D0}/V_t};$$

$$J_3 = I_{D0} - G_3 U_{D0}.$$

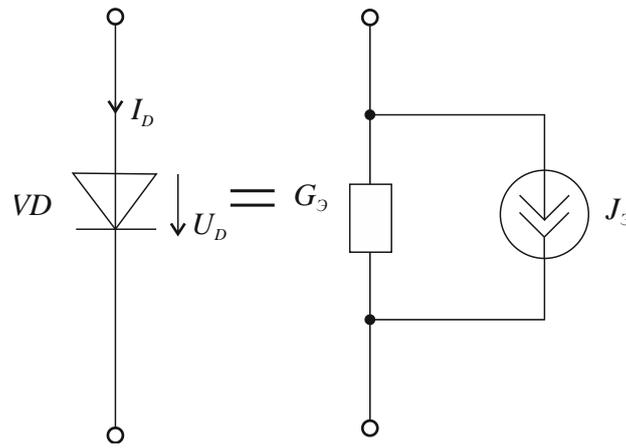


Рис. 13.10 – Линейная схема замещения

При комнатной температуре (25°C) $V_t \approx 25.8$ мВ. Обратный ток насыщения маломощного диода $I_0 = 10^{-15}$ А. Если напряжение диода $U_{D0} = 0.7$ В, ток $I_{D0} = I_0 (e^{U_{D0}/V_t} - 1) = 1$ мА, то параметры дискретной линейной схемы замещения диода таковы:

$$G_3 = \frac{I_0}{V_t} e^{U_{D0}/V_t} = **;$$

$$J_3 = I_{D0} - G_3 U_{D0} = **.$$

Выводы

1. Одна из *важнейших задач анализа нелинейных цепей* заключается в определении рабочих точек или точек смещения электронных приборов.
2. Рабочей точкой нелинейного элемента называют *значения постоянных напряжения и тока*, изображаемых в виде точки на его вольт-амперной характеристике.
3. Для определения рабочих точек нелинейных элементов используют графические и численные методы.
4. Графический метод определения рабочей точки является *приближенным*. Его *достоинствами* являются простота и наглядность. Графический метод можно использовать для определения начального приближения, а затем для получения точного решения применить численный метод.

5. Для анализа сложных нелинейных цепей используют *численные методы*. Наиболее распространенным методом решения уравнений, описывающих поведение нелинейных резистивных цепей, является метод Ньютона – Рафсона.
6. В программах схемотехнического моделирования используют схемную интерпретацию метода Ньютона – Рафсона – метод линеаризованных схем замещения [3].

Контрольные вопросы

1. Что такое рабочая точка и нагрузочная прямая?
2. В каких случаях можно использовать графический метода анализа?
3. Какие эквивалентные схемы получили название дискретных линейных схем замещения?
4. Какие методы используют при анализе сложных нелинейных цепей?
5. Как реализуется схемная интерпретация метода Ньютона – Рафсона?

Литература

1. Влах, И. Машинные методы анализа и проектирования электронных схем / И. Влах, К. Синггал. – М. : Радио и связь, 1988. – 560 с.
2. Норенков, И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры / И. П. Норенков, В. Б. Маничев. – М. : Высш. шк., 1983. – 272 с.
3. Дмитриев, В. М. МАРС – среда моделирования технических устройств и систем / В. М. Дмитриев, А. В. Шутенков, Т. Н. Зайченко, Т. В. Ганджа. – Томск : В-Спектр, 2011. – 278 с.

14 Моделирование дискретных систем

14.1 Математическое описание систем дискретного управления

Дискретно представляемые сигналы описываются *функциями дискретной переменной*. Для описания дискретных систем используются решетчатые функции и разностные уравнения. *Решетчатые функции* являются аналогами непрерывных функций, описывающих непрерывные системы, а *разностные уравнения* являются аналогами дифференциальных уравнений.

Решетчатой функцией называется функция, получающаяся в результате замены непрерывной переменной на дискретную независимую переменную, определенную в дискретные моменты времени kT , $k = 0, 1, 2, \dots$. Непрерывной функции $x(t)$ соответствует решетчатая функция $x(kT)$, где T – период квантования, при этом непрерывная функция является огибающей решетчатой функции. При заданном значении периода квантования T непрерывной функции $x(t)$ соответствует однозначная решетчатая функция $x(kT)$. Однако обратного однозначного соответствия между решетчатой и непрерывной функцией не существует, так как через ординаты решетчатой функции можно провести множество огибающих.

Отсчеты по шкале времени удобно вести в целочисленных единицах периода квантования T . С этой целью вместо переменной t непрерывной функции введем новую переменную $\tau = t/T$, при этом непрерывной функции $x(\tau)$ будет соответствовать решетчатая функция $x(k) \equiv x_k$.

Теорема Котельникова – Шеннона. Процедура преобразования сигнала непрерывного времени $x(t)$ к дискретному виду, квантованному по времени, называется *квантованием*. Такая процедура отражает как реальные процессы, проходящие в цифровых системах управления, так и математические операции, использующиеся в различных сферах теории информации. В результате квантования получается импульсная последовательность $x(kT)$ (решетчатая функция), которая при $t = kT$ совпадает с исходным сигналом: $x(kT) = x(t)|_{t=kT}$ и не определена между отсчетами k . Потери информации при квантовании зависят от величины интервала квантования T (частоты квантования $2\pi/T$). Выбор интервала T обычно осуществляется из соображений теоретической возможности точ-

ного восстановления исходного сигнала по данной дискретной выборке. Согласно теореме Котельникова – Шеннона, если спектр сигнала $x(t)$ ограничен максимальной частотой ω_{\max} , то точное восстановление функции $x(t)$ теоретически возможно при условии, что на одном периоде максимальной частоты в сигнале имеется минимум два дискретных отсчета, т. е. частота квантования ω должна быть более чем в 2 раза больше наибольшей частоты ω_{\max} в сигнале:

$$\omega \geq 2\omega_{\max}, T < \pi / \omega_{\max}.$$

Разностные уравнения. Связь между значениями решетчатой функции при разных значениях аргумента определяется с помощью конечных разностей, которые являются аналогами производных в дифференциальных уравнениях.

Разностью первого порядка (первой разностью) называется разность между последующим дискретным значением решетчатой функции и ее текущим значением:

$$\Delta x(k) = x(k+1) - x(k).$$

Разность первого порядка характеризует скорость изменения решетчатой функции и, следовательно, является аналогом первой производной непрерывной функции.

Разность второго порядка определяется как разность двух соседних разностей первого порядка:

$$\begin{aligned} \Delta^2 x(k) &= \Delta x(k+1) - \Delta x(k) = (x(k+2) - x(k+1)) - (x(k+1) - x(k)) = \\ &= x(k+2) - 2x(k+1) + x(k). \end{aligned}$$

Разности любого m -го порядка вычисляются аналогично:

$$\begin{aligned} \Delta^m x(k) &= \Delta^{m-1} x(k+1) - \Delta^{m-1} x(k); \\ \Delta^m x(k) &= \sum_{n=0}^m (-1)^n x(k+m-n) m! / (k!(m-n)!). \end{aligned}$$

Дискретизация автономных систем. Под дискретизацией системы подразумевается преобразование непрерывной динамической модели к дискретной форме описания в разностных уравнениях. При этом предполагается, что в моменты $t = kT$ импульсные сигналы $x(kT)$ полученной дискретной модели с определенной степенью точности повторяют значения сигналов $x(t)$ исходной непрерывной системы. С использованием разностных уравнений математическое описание линейных импульсных систем приводится к виду:

$$a_m \Delta^m x(k) + a_{m-1} \Delta^{m-1} x(k) + \dots + a_0 x(k) = 0. \quad (14.1)$$

Уравнение (14.1) является *линейным разностным уравнением с постоянными коэффициентами* $a_m (m=0, 1, 2, \dots)$, аналогом однородного линейного дифференциального уравнения при описании непрерывных динамических систем. Решение (14.1) дает значение дискретной переменной $x(k)$ для каждого периода квантования. Уравнение (14.1) можно записать в виде:

$$\sum_{n=0}^m c_n x(k+n) = 0. \quad (14.2)$$

Таким образом, в дискретной системе (14.1) процессы в квантованные моменты времени $t - kT$ точно совпадают с процессами в исходной непрерывной системе. Так как решения дискретной системы в промежуточные моменты времени не определены, то корректный переход к дискретной форме предусматривает выбор интервала квантования T в соответствии с теоремой Котельникова – Шеннона.

Дискретное z -преобразование. В теории импульсных систем для решения разностных уравнений используется *дискретное преобразование Лапласа и его модификация* – дискретное z -преобразование. Преобразование Лапласа для непрерывной функции $x(t)$:

$$X(p) = \int_0^{\infty} x(t) \exp(-pt) dt. \quad (14.3)$$

При переходе к дискретной функции $x(kT)$ заменим интегрирование суммированием:

$$X(p) = T \sum_{k=0}^{\infty} x(kT) \exp(-pkT). \quad (14.4)$$

Введем новую переменную $z = \exp(pt)$:

$$X(z) = T \sum_{k=0}^{\infty} x(kT) z^{-k}. \quad (14.5)$$

Это уравнение представляет собой *дискретное преобразование Лапласа*, в котором выражение

$$X(z) = \sum_{k=0}^{\infty} x(kT) z^{-k} \quad (14.6)$$

называется *z -преобразованием*. Оно лежит в основе метода решения разностных уравнений. Дискретное преобразование Лапласа $X(z)$ отличается от z -преобразования наличием нормирующего множителя T . При анализе дискретных си-

стем z -преобразование позволяет перейти от разностных уравнений к алгебраическим и существенно упростить анализ динамики дискретных систем. В выражении (14.6) функция $x(kT)$ называется *оригиналом решетчатой функции*, а $X(z)$ – ее изображением. Для обратного перехода от изображения к оригиналу (для нахождения исходной решетчатой функции по ее изображению) используется обратное z -преобразование:

$$x(kT) = (1 / 2\pi j) \oint X(z) z^{k-1} dz.$$

Корни p_i характеристического полинома непрерывной системы связаны с корнями z_i характеристического полинома эквивалентной дискретной системы соотношением

$$z_i = \exp(Tp_i). \quad (14.7)$$

В общем случае отображение (14.7) неоднозначно, и нескольким различным значениям p_i может соответствовать одно и то же значение z_i . Взаимнооднозначное соответствие корней непрерывной и эквивалентной дискретной систем выполняется только при интервале дискретизации, удовлетворяющем теореме Котельникова – Шеннона. Преобразование непрерывного сигнала в цифровой код выполняется амплитудно-цифровыми преобразователями (АЦП) и включает *три операции*: квантование сигнала по времени, квантование по уровню и кодирование. Квантование по времени заключается в измерении непрерывной величины $x(t)$ в дискретные моменты времени $t_k = k\Delta t$, $\Delta t = \text{const}$, $k = 0, 1, 2, \dots$, осуществляется импульсным элементом (ИЭ). На выходе импульсного элемента получается решетчатая функция $x(tk)$.

Процесс квантования решетчатой функции $x(tk)$ по уровню можно представить как прохождение сигнала $x(tk)$ через нелинейный элемент с многоступенчатой релейной характеристикой – квантователь по уровню (рис. 14.1). В результате квантования по уровню точно измеренные значения сигнала $x(tk)$ заменяются приближенными ближайшими дискретными значениями $xk \equiv x(k) \cong \cong x(tk)$. Шаг квантования δk характеризует точность преобразователя. Учет квантования по уровню приводит к необходимости рассмотрения нелинейных цифровых систем. Анализ систем упрощается, если элемент с многоступенчатой релейной характеристикой представить в виде параллельного соединения линейного усилительного элемента с коэффициентом $K = 1$, характеристика которого изображена на рисунке 14.1, в, и нелинейного элемента с характеристикой $\delta(k)$,

равной разности между линейной и релейной характеристиками. В этом случае квантованный по уровню сигнал можно представить как сумму точного сигнала $x(tk)$ и добавочного сигнала $\delta(k)$, ограниченного по величине половиной ступени квантования: прежде чем сигнал $x(k)$ поступает на цифровое вычислительное устройство (ЦВУ) системы, осуществляется его кодирование – преобразование в цифровой код $x_{ц}(k)$.

Если в ЦВУ используется двоичная система счисления, то с помощью кодирующего устройства K каждый импульс, поступающий с квантователя по уровню, преобразуется в двоичный цифровой код, соответствующий амплитуде этого импульса. Двоичные числа представляются в виде последовательности импульсов, разделенных интервалом времени t . Каждому разряду двоичного числа отводится интервал времени t' на выставление кодов 0 или 1 (обычно отсутствие или наличие определенного уровня напряжения). На ЦВУ числа могут поступать последовательным или параллельным кодом. В первом случае разряды числа идут последовательно друг за другом по одному каналу, как правило, начиная с младшего. Одно число от другого отделяется специальным маркерным импульсом. Минимальный интервал T передачи числа равен nt , где n – количество разрядов числа. При параллельном коде все разряды числа поступают одновременно по нескольким каналам, число которых равно числу разрядов. Так как при кодировании сигнала не происходит изменения информации, то передаточная функция кодирующего устройства равна единице.

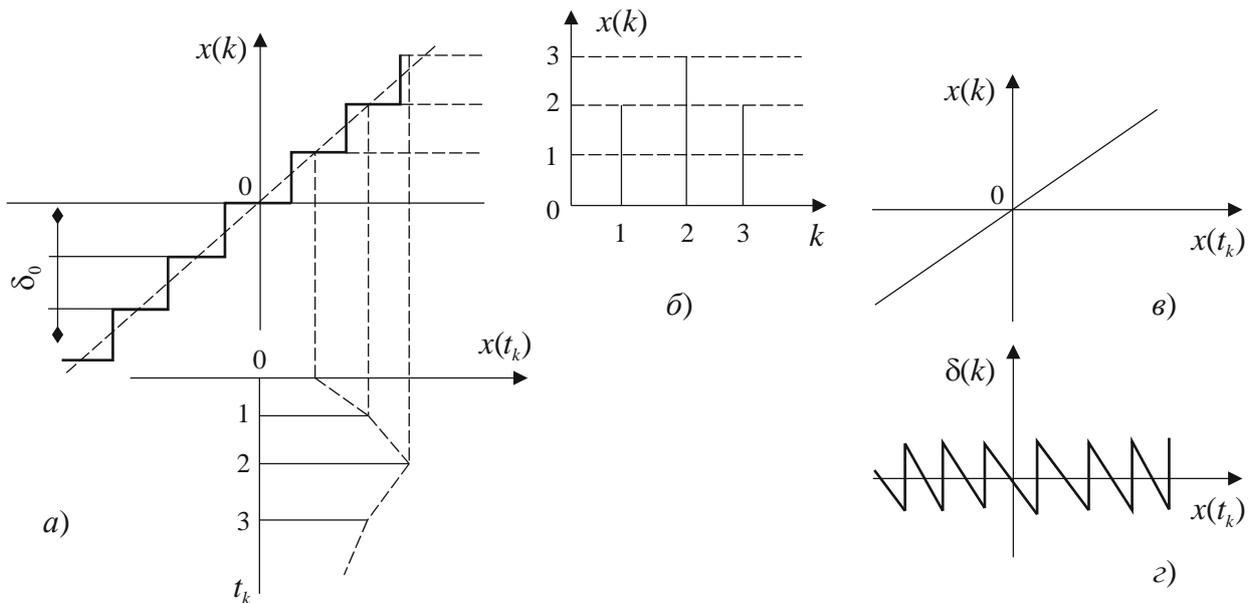


Рис. 14.1 – Квантователь по уровню

Цифровое вычислительное устройство можно рассматривать как дискретный преобразователь, преобразующий входную последовательность чисел $x_{ц}(k)$ в выходную $y_{ц}(k)$ в соответствии с заложенной программой вычислений, представляющей собой алгоритм переработки информации. В дискретной линейной системе связь между входом и выходом (входной и выходной дискретными последовательностями значений сигнала – отсчетами) задается линейным оператором преобразования TL :

$$y(k\Delta t) = TL\{x(k\Delta t)\}.$$

Это выражение отображает краткую запись линейного разностного уравнения:

$$\sum_{m=0}^M a_m y(k\Delta t - m\Delta t) = \sum_{n=0}^N b_n x(k\Delta t - n\Delta t), \quad (14.8)$$

где a_m и b_n – вещественные или, в общем случае, комплексные коэффициенты. Выполним нормировку уравнения (14.8) к $a_0 = 1$ и, принимая в дальнейшем $\Delta t = 1$, приведем его к виду:

$$y(k) = \sum_{n=0}^N b_n x(k-n) - \sum_{m=0}^M a_m y(k-m). \quad (14.9)$$

ЦВУ, которые описываются полным разностным уравнением (14.9), принято называть *рекурсивными цифровыми фильтрами (РЦФ)*, так как в вычислении текущих выходных значений участвуют не только входные данные, но и значения выходных данных фильтрации, вычисленные в предшествующих циклах расчетов. С учетом последнего фактора рекурсивные фильтры называют также *фильтрами с обратной связью*, положительной или отрицательной в зависимости от знака суммы коэффициентов a_m . По существу, полное окно рекурсивного фильтра состоит из *двух составляющих*: нерекурсивной части b_n , ограниченной в работе текущими и «прошлыми» значениями входного сигнала, и рекурсивной части a_m , которая работает только с «прошлыми» значениями выходного сигнала. Техника вычислений для РЦФ приведена на рисунке 14.2.

Передаточные функции ЦВУ. Удобным методом решения разностных уравнений линейных систем является z -преобразование. Применяя z -преобразование к обеим частям равенства (14.8), с учетом сдвига функций ($y(k-m) \Leftrightarrow z^{-m}Y(z)$) получаем:

$$Y(z) \sum_{m=0}^M a_m z^{-m} = X(z) \sum_{n=0}^N b_n z^{-n}, \quad (14.10)$$

где $X(z)$, $Y(z)$ – соответствующие z -образы входного и выходного сигнала. Отсюда, полагая $a_0 = 1$, получаем в общей форме функцию связи выхода фильтра с его входом – уравнение передаточной функции системы в z -области:

$$H(z) = Y(z) / X(z) = \sum_{n=0}^N b_n z^{-n} / \left(1 + \sum_{m=1}^M a_m z^{-m} \right). \quad (14.11)$$

Для нерекурсивных ЦВУ при нулевых коэффициентах a_m

$$H(z) = \sum_{n=0}^N b_n z^{-n}. \quad (14.12)$$

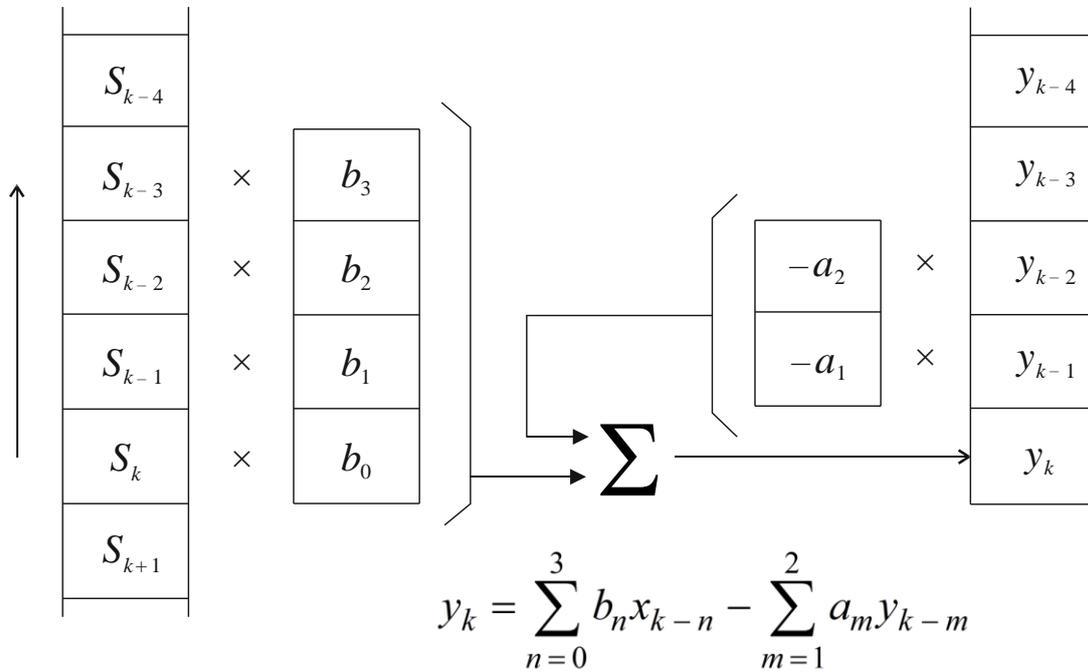


Рис. 14.2 – Рекурсивный цифровой фильтр

При проектировании фильтров исходной, как правило, является частотная передаточная функция фильтра $H(\omega)$, по которой вычисляется ее Z -образ $H(z)$ и обратным переходом в пространство сигналов определяется алгоритм обработки данных. В общей форме для выходных сигналов фильтра:

$$Y(z) = H(z) \cdot X(z).$$

$$Y(z) \cdot \left(1 + \sum_{m=1}^M a_m z^{-m} \right) = X(z) \sum_{n=0}^N b_n z^{-n}.$$

$$Y(z) = X(z) \sum_{n=0}^N b_n z^{-n} - Y(z) \sum_{m=1}^M a_m z^{-m}. \quad (14.13)$$

После обратного Z -преобразования выражения (14.13)

$$y(k) = \sum_{n=0}^N b_n (k-n) - \sum_{n=0}^N b_n (k-n). \quad (14.14)$$

При подаче на вход фильтра единичного импульса Кронекера δ_0 , имеющего z -образ $\delta(z) = z - n = 1$, сигнал на выходе фильтра будет представлять собой импульсную реакцию фильтра $y(k) \equiv h(k)$, при этом

$$H(z) = Y(z) / \delta(z) = Y(z) = TZ[y(k)] = \sum_{k=0}^{\infty} h(k) z^{-k}, \quad (14.15)$$

т. е. передаточная функция фильтра является z -образом ее импульсной реакции. При обратном z -преобразовании передаточной функции соответственно получаем импульсную характеристику фильтра:

$$h(k) \Leftrightarrow H(z). \quad (14.16)$$

Если функция $H(z)$ представлена конечным степенным полиномом, то обратное z -преобразование осуществляется элементарно идентификацией коэффициентов по степеням z . Передаточная функция также может быть представлена степенным полиномом прямым делением числителя на знаменатель правой части выражения (14.11), однако результат при этом может оказаться как конечным, так и бесконечным, т. е. система может иметь либо конечную, либо бесконечную импульсную характеристику. Практически используемые рекурсивные фильтры обычно имеют бесконечную импульсную характеристику (БИХ-фильтры) при конечном числе членов алгоритма фильтрации (14.14). Система устойчива, если при любых начальных условиях ее реакция на любое ограниченное воздействие также ограничена. *Критерием устойчивости* является абсолютная сходимость отсчетов импульсного отклика системы:

$$E_n |h(n)| < \infty. \quad (14.17)$$

Анализ устойчивости может быть проведен по передаточной функции. В устойчивой системе передаточная функция не должна иметь особых точек (полюсов) на и вне единичного круга на z -плоскости. Отсюда необходимое и достаточное условие устойчивости импульсных систем – модули корней передаточ-

ной функции (14.11) должны быть меньше 1 (полюса передаточной функции системы внутри единичного круга на z -плоскости). Чем меньше значения модулей корней, тем больше запас устойчивости системы.

Частотные характеристики ЦВУ. От z -образов сигналов и передаточных функций подстановкой $z = \exp(j\omega\Delta t)$ можно перейти к Фурье-образам функций, т. е. к частотным спектрам сигналов и частотной характеристике цифровых систем, а точнее – к функциям их спектральных плотностей. Передаточная частотная функция (частотная характеристика при $a_0 = 1$):

$$H(\omega) = A(\omega) / B(\omega) = \sum_{n=0}^N b_n \exp(-j\omega n\Delta t) / \left[1 + \sum_{m=1}^M a_m \exp(-j\omega m\Delta t) \right]. \quad (14.18)$$

Частотная характеристика системы представляет собой Фурье-образ его импульсной реакции, и наоборот. При $\Delta t = 1$:

$$H(\omega) = \sum_{n=-\infty}^{\infty} h(n) \exp(-j\omega n), \quad (14.19)$$

$$h(n) = (1/2\pi) \int_{-\pi}^{\pi} H(\omega) \exp(j\omega n) d\omega. \quad (14.20)$$

В общем случае $H(\omega)$ является *комплексной функцией*, модуль которой $R(\omega)$ называется *амплитудно-частотной характеристикой* (АЧХ), а аргумент $\varphi(\omega)$ – *фазово-частотной характеристикой* (ФЧХ).

$$A(\omega) = |H(\omega)| = \sqrt{\operatorname{Re}^2 H(\omega) + \operatorname{Im}^2(\omega)};$$

$$\varphi(\omega) = \operatorname{arctg}(-\operatorname{Im} H(\omega) / \operatorname{Re} H(\omega)).$$

Выбор знака фазового угла ориентирован на каузальные системы с отрицательным временным запаздыванием сигналов. Допустим, что система осуществляет только сдвиг сигнала $x(t)$ вправо по временной оси, т. е. $y(t) = x(t - \tau)$. Для преобразования Фурье функции $y(t)$ имеем:

$$Y(f) = \int_{-\infty}^{\infty} y(t) \exp(-j2\pi ft) dt = \int_{-\infty}^{\infty} x(t - \tau) \exp(-j2\pi ft) dt =$$

$$= \exp(-j2\pi f \tau) \int_{-\infty}^{\infty} x(t) \exp(-j2\pi ft) dt = \exp(-j2\pi f \tau) X(f).$$

Отсюда:

$$H(f) = Y(f) / X(f) = \exp(-j2\pi f \tau), \quad |H(f)| = 1, \quad jh(f) = -2\pi f \tau.$$

Из последнего равенства следует, что фаза представляет собой прямую с отрицательным тангенсом угла наклона $-2pf\tau$. Соответственно, для всех каузальных фильтров, осуществляющих преобразование с определенной задержкой сигнала на выходе, при выполнении операции над частотными составляющими сигнала имеет место:

$$\begin{aligned} Y(f) &= H(f)X(f) = |H(f)|\exp(j\varphi_h(f))|X(f)|\exp(j\varphi_x(f)) = \\ &= |H(f)||X(f)|\exp\{j[\varphi_h(f) + \varphi_x(f)]\} \\ |Y(f)| &= |H(f)||X(f)|, \quad \varphi_y(f) = \varphi_h(f) + \varphi_x(f). \end{aligned}$$

С учетом отрицательного знака $\varphi_h(f)$ фазовой характеристики каузальных фильтров это вызывает сдвиг в «минус» всех частотных составляющих сигнала и соответствующую задержку выходного сигнала относительно входного [1–2].

14.2 Модели состояния линейной дискретной системы

Математические модели дискретных систем управления описывают поведение этих систем только в квантованные моменты времени: tk , $k = 0, 1, 2, \dots$. Дискретным представлением непрерывных сигналов $u(t)$, $y(t)$, $x(t)$ являются последовательности: $\{u(tk)\}$, $\{y(tk)\}$, $\{x(tk)\}$. Математические модели дискретных систем устанавливают взаимосвязь между этими последовательностями. Практически все объекты и процессы управления имеют непрерывный характер своего состояния и динамики развития во времени, поэтому дискретные автоматические системы управления содержат в своей структуре как цифровую (дискретную), так и аналоговую (непрерывную) части. Для согласования этих частей в системе используются аналого-цифровые и цифро-аналоговые преобразователи (АЦП и ЦАП). АЦП ставит в соответствие непрерывной функции $f(t)$, $t \geq t_0$ последовательность $\{f(tk)\} = f(k\Delta t)$, $\Delta t = \text{const}$, $k = 0, 1, 2, \dots$. В свою очередь ЦАП осуществляет преобразование последовательности $\{fk, k = 0, 1, 2, \dots\}$ в некоторую непрерывную функцию, которая является аппроксимацией исходной функции $f(t)$, $t \geq t_0$. Часто используют кусочно-постоянную аппроксимацию, поэтому такой преобразователь называют *экстраполятором*, или *фиксатором нулевого порядка* [3–4].

Контрольные вопросы

1. Какая функция называется решетчатой?

2. Каким термином характеризуется преобразование сигнала непрерывного времени $x(t)$ к дискретному виду?
3. Что характеризует точность преобразователя?
4. Для чего используют дискретное преобразование Лапласа?
5. Какие элементы используются в системе для согласования непрерывных и дискретных сигналов?
6. Какие цифровые операции включает АЦП?
7. Какой преобразователь называют экстраполятором?

Литература

1. Мирошник, И. В. Теория автоматического управления. Линейные системы : учеб. пособие для вузов / И. В. Мирошник. – СПб. : Питер, 2005. – 336 с.
2. Зайцев, Г. Ф. Теория автоматического управления и регулирования / Г. Ф. Зайцев. – Киев : Выща школа, 1989. – 431 с.
3. Повзнер, Л. Д. Теория систем управления : учеб. пособие для вузов / Л. Д. Повзнер. – М. : Изд. МГГУ, 2002. – 472 с.
4. Михайлов, В. С. Теория управления / В. С. Михайлов. – Киев : Выща школа, 1988. – 312 с.

15 Агрегативные модели

15.1 Определение дискретно-событийной системы

Системы называются *дискретно-событийными*, если изменения переменных состояния в них происходят только в явно-определенные моменты времени или под влиянием явно-определенных событий. Находясь в некотором состоянии, дискретная система сохраняет его до наступления очередного события, после которого переменные системы изменяются скачком. Так, на примере банка состояние системы изменяется, когда новый клиент входит в банк или освобождается кассир, а это уже можно считать мгновенными событиями, сопровождаемыми изменением состояния системы.

Динамика моделей схемы сравнения (СС) может быть представлена посредством последовательности происходящих событий. Для этого введем ряд дополнительных понятий. Будем характеризовать каждый элемент $A_j \in A$ набором признаков $\Pi_j = (z_j, v_j)$, где z_j – информация для прогноза будущей динамики элемента; v_j – служебные величины. В каждом элементе A_j могут происходить события, обозначаемые e_j . Время наступления очередного события определяется текущим состоянием z_j , а его содержание признаком Π_j . Тогда в состав состояния элемента A_j можно ввести величину τ_j , равную интервалу времени до наступления очередного события e_j .

Опишем теперь *динамику модели*. Пусть в начальный момент времени t_0 определены состав $A^0 \in A$ и структура системы S^0 . Для каждого элемента $A_j \in A$ известны признаки Π_j и время наступления события e_j . Найдем величину $\tau^1 = \min\{\tau\} : \{A_j \in A^0\}$, равную времени наступления самого раннего события. Будем считать, что ни состав системы A^0 , ни признаки элементов из A^0 , ни структура S^0 не меняются до момента времени $t_1 = t_0 + \tau^1$. В момент t_1 в элементе A_j происходит событие e_j , содержание которого задано признаком Π_j . В этот момент происходит взаимодействие элементов, входящих в A^0 в соответствии со структурой S^0 . В результате взаимодействия формируется новый состав элементов, входящих в A^1 и структурой S^1 . Формально это означает, что

состав элементов из A^1 и признаки элементов из A^1 , а также структура S^1 являются функцией e_j , состава A^0 , признаков элементов из A^0 и структуры S^0 . Далее определяется время τ^2 до наступления самого раннего события в элементах из A^1 , находится следующий момент времени смены состава элементов и их признаков $t_2 = t_0 + \tau^2$ и т. д.

15.2 Агрегативные модели. Кусочно-линейный агрегат

В 60-х гг. XX в. ученые Н. П. Бусленко и И. Н. Коваленко [1] ввели класс моделей СС, названных ими *агрегативными*, а основными элементами таких систем являются кусочно-линейные агрегаты.

Кусочно-линейные агрегаты (КЛА) – это класс объектов, который принято изображать в виде преобразователя (рис. 15.1), функционирующего во времени $t \in T = [0, \infty]$ и способного воспринимать входные сигналы x из X , выдавать выходные сигналы y из Y и находиться в некотором состоянии z из Z . Таким образом, КЛА отражает специфику множеств X, Y, Z , допустимые формы входных и выходных сообщений $x(t)$ и $y(t)$, $t \in T$, траекторий $z(t)$, $t \in T$, а также способ преобразования входного сообщения в выходное.

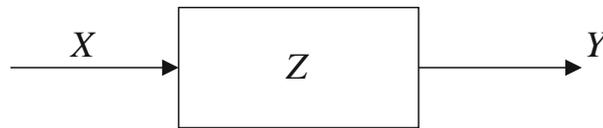


Рис.15.1 – Преобразователь, функционирующий во времени

Отметим, что динамика КЛА носит событийный характер. В КЛА могут происходить *внутренние и внешние* события. *Внутренние* заключаются в достижении траекторий $Z^* \in Z$ состояний; *внешние* – в поступлении входного сигнала. Предполагается, что здесь исключен случай поступления непрерывных сигналов (для систем, описываемых дифуравнениями). Также здесь полагается, что моменты достижения траекторий $z(t)$ множества Z^* изолированы друг от друга. Между событиями КЛА изменяются детерминированным образом. Каждому состоянию z ставится в соответствие величина $\tau = \tau(z)$, трактуемая как потенциальное время до наступления очередного внутреннего события $\tau = \tau(z) = \inf \{t: T_t z \in Z^*\}$. Если t^* – момент наступления внутреннего события $z^* = \lim z(t) \in Z^*$, то состояние КЛА в момент t^* является случайным, зависящим

от t^* и не принадлежащим Z^* . Последнее условие гарантирует изолированность моментов наступления внутренних событий. В момент t^* выдается выходной сигнал y^* , содержание которого будет зависеть лишь от z^* . В частности, выходной сигнал может быть и пустым, т. е. не выдаваться. После случайного скачка вновь определяется время до наступления внутреннего события. Условимся считать, что в КЛА имеется *приоритет входных сигналов над внутренними* – при совпадении моментов наступления входного и внутреннего сигналов выполняется внешний сигнал, который определяет внутренний.

Таким образом, *процесс функционирования КЛА* полностью определяется изменениями, происходящими в особые моменты времени – моменты наступления событий (внешних или внутренних). Между особыми моментами состояния КЛА меняется детерминированно в соответствии с оператором T_Q .

15.3 Схема сопряжения агрегативной системы

Формализуем понятие *структуры СС*. Пусть $A = (A_1, A_2, \dots, A_N)$ – множество КЛА, N – фиксированное число. Обозначим через I_i множество входных клемм КЛА A_i , а через O_i – множество его выходных клемм. Пусть $I = \cup I_i$; $O = \cup O_i$ – множества входных и выходных клемм всех рассматриваемых КЛА.

Рассмотрим множество всех возможных отображений $R, I \rightarrow O$, которое можно интерпретировать как множество потенциальных соединений между собой КЛА, входящих в A . Именно, каждой входной клемме из I ставится в соответствие выходная клемма, с которой на входную клемму поступает сигнал. При этом допускается, что некоторым входным клеммам не ставятся в соответствие никакие выходные клеммы (на них нет сигналов). Точно так же некоторые выходные клеммы могут быть *висячими*. Таким образом, на каждую входную клемму подаются сигналы не более чем с одной выходной клеммы. В то же время с одной выходной клеммы сигналы могут идти на несколько различных входных клемм.

Определим соотношение реализуемости. Пусть отображение R из \mathcal{R} ставит в соответствие l -й входной клемме агрегата A_i (назовем ее (I, i) -входом) r -ю выходную клемму агрегата A_j (назовем ее (j, r) -выходом). Каждому (I, i) -входу соответствует $X_{(I, i)}$ допустимых значений воспринимаемых входных сигналов. Точно так же каждому (j, r) -выходу соответствует множество

$Y_{(j,r)}$ допустимых значений снимаемых выходных сигналов. Тогда необходимым условием того, что (I,l) -вход может воспринять сигнал с (j,r) -выхода, является соотношение

$$Y_{(j,r)} \subset X_{(I,l)}. \quad (15.1)$$

Назовем отображение R допустимым, если для каждого (I,i) -входа из области его определения выполнено соотношение (15.1). Любое допустимое отображение R из \mathcal{R} назовем *схемой сопряжения*. Схема сопряжения указывает адресацию сигналов в системе, состоящей из КЛА. Если (j,r) -выход является значением отображения R (I,l) -входа и при этом отображение R является схемой сопряжения, то будем говорить, что между (j,r) -выходом и (I,l) -входом проложен *канал связи*.

Пример агрегативной системы

Зададим состав агрегатов A и схему сопряжения R для однозначного определения динамики получившейся системы.

Пусть модель состоит из трех КЛА: $A = (A_1, A_2, A_3)$. КЛА A_1 является *автономным*, не имеющим входных клемм и имеющим одну выходную клемму. КЛА A_2 имеет две входные и одну выходную клемму, а КЛА A_3 имеет одну входную и одну выходную клеммы. Схема сопряжения R приведена на рисунке 15.2.

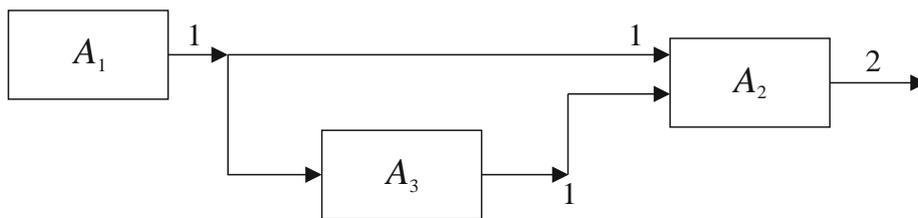


Рис. 15.2 – Схема сопряжения КЛА

15.4 Агрегативная модель укладки бетона

Процесс строительства плотины ГЭС (на примере Курпсайской ГЭС на реке Нарын) состоит из следующих основных этапов: добыча щебня из карьеров, производство бетона на бетонных заводах (БЗ), перевозка бетона с БЗ на плотину, разгрузочные работы и укладка бетона, работы на плотине.

Проблемная ситуация представлена на рисунке 15.3.

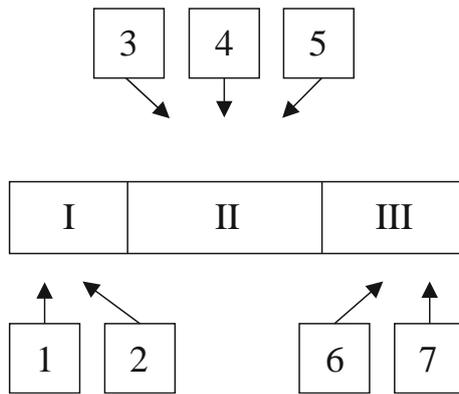


Рис. 15.3 – Проблемные ситуации моделирования строительства ГЭС

Общая схема изучаемой проблемы (рис.15.3).

- Производство бетона (I) обеспечивают карьер и БЗ $\{(1) \text{ и } (2) \rightarrow (I)\}$.
- Для перевозки бетона (II) определяющими являются структура транспортного хозяйства (количество и типы автотранспорта), организация дорожного движения, характеристики дороги $\{(3), (4) \text{ и } (5) \rightarrow (II)\}$.
- Процесс укладки бетона (III) зависит от количества укладочных мест, состава укладочных механизмов, режимов их работы, графиков их ввода в действие по мере укладки плотины $\{(III) (6) \text{ и } (7) \rightarrow (III)\}$.

Однако представленная схема (рис. 15.3) лишь акцентирует внимание на объектах и процессах, требующих изучения или учета. Следующим этапом является описание взаимодействия основных изучаемых процессов, их характеристик и определение границ модели.

Для рассматриваемого примера такое описание может быть представлено в виде, изображенном на рисунке 15.4.

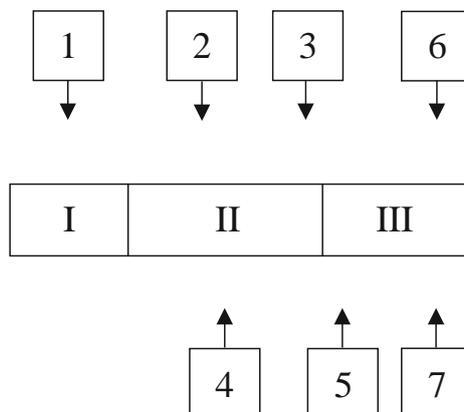


Рис. 15.4 – Факторы, определяющие границы модели при строительстве ГЭС

Факторы, определяющие границы модели строительства ГЭС:

- карьер (1);
- характеристика транспорта (2);
- погодные условия (3);
- организация движения (4);
- характеристики дороги (5);
- обеспечение строительства металлоконструкциями (6);
- организация и технологии производства бетонных работ (7).

После выделения основных элементов и процессов в модели необходимо решить вопрос о степени подробности их представления следует определить, какие факторы и параметры учитываются и регистрируются в каждом из выделенных процессов.

Перечень факторов и параметров системы зависит как от назначения модели системы, так и от исследуемых режимов ее работы. В данном случае модель системы служит для выяснения возможностей по производству, перевозке и укладке бетона. В соответствии с этим для изучаемой системы принята следующая детализация элементов и процессов.

Для построения модели разгрузки и укладки бетона принимаются во внимание следующие факторы: очередь груженого транспорта на разгрузку, процесс разгрузки, рабочий цикл разгрузочного механизма и его надежность.

Предположим, что разгрузочный механизм действует следующим образом: он загружается из первого в очереди бетоновоза, после чего бетоновоз уезжает, а механизм начинает двигаться к месту выгрузки; затем механизм разгружается и движется к месту приемки бетона. Кроме того, возможны случайные отказы разгрузочного механизма, поэтому тратится время на его ремонт.

Для описания модели используется агрегат ПЛОТИНА, имеющий одну входную и одну выходную клеммы. На входную клемму поступают сигналы вида БЕТОНОВОЗ. Аналогичный сигнал снимается с выходной клеммы (рис. 15.5).

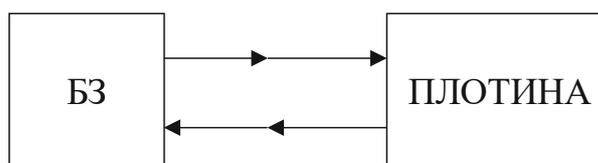


Рис. 15.5 – Агрегативное представление

Состояние КЛА ПЛОТИНА назовем именем РАЗГРУЗКА и представим его в виде следующей структуры:

РАЗГРУЗКА = (БЕТОНОВОЗЫ, МЕХАНИЗМ, ОПЕРАЦИЯ, ОТКАЗ),

где

БЕТОНОВОЗЫ – список данных, соответствующих очереди на разгрузке;

МЕХАНИЗМ – целочисленная величина, характеризующая состояние разгрузочного механизма: 0 – неисправен, 1 – загружается; 2 – движется для разгрузки; 3 – разгрузка; 4 – движется для загрузки;

ОПЕРАЦИЯ – ДЕЙСТВИТЕЛЬНАЯ ВЕЛИЧИНА, представляющая собой время до окончания операции, выполняемой разгрузочным механизмом (загрузка, движение, разгрузка; ремонт);

ОТКАЗ – действительная величина, равная оставшемуся времени безотказной работы механизма.

В КЛА ПЛОТИНА могут происходить следующие события: $E1$ – поступление входного сигнала (приезд нового бетоновоза); $E2$ – обнуление координаты ОПЕРАЦИЯ (окончание разгрузочным механизмом очередной операции); $E3$ – обнуление координаты ОТКАЗ (отказ механизма). Подробно анализ содержания этих состояний приведен в [2].

15.5 Оценка агрегативных систем как моделей сложных систем

В основу понятия агрегативных систем положено структурное представление системы в виде взаимодействующих элементов (КЛА), что соответствует одной из основных концепций сложных систем.

Динамика агрегативных систем полностью определяется последовательностью событий, происходящих в ней, что соответствует концепции алгоритмической модели СС, которая конкретизирует ряд абстрактных понятий: признаки, события, состояния, структура. Следствием этого является удобство реализации агрегативных систем на ЭВМ. Агрегативные системы служат определенным обобщением таких хорошо известных схем и систем, как автоматы и модели массового обслуживания. В агрегативном виде могут быть представлены и сети Петри, и практически любые численные методы решений дифференциальных уравнений. В агрегативном виде можно также представить модели, имеющие вид черного ящика. Структуры данных, описывающие состояния и сигналы агрегативных систем, помогают формализовать концептуальное представление, которое существует у пользователя относительно компонентов системы.

Контрольные вопросы

1. Какие системы называются дискретно-событийными?
2. Дайте определение КЛА.
3. Определите понятие структуры схемы сопряжения.
4. Характеризуйте пример агрегативной системы.

Литература

1. Бусленко, Н. П. Лекции по теории сложных систем / Н. П. Бусленко, Н. Н. Калашников, И. Н. Коваленко. – М. : Советское радио, 1973. – 441 с.
2. Бусленко, Н. П. Моделирование сложных систем / Н. П. Бусленко. – М. : Главная редакция физико-математической литературы, 1978. – 356 с.

16 Язык моделирования алгоритмических конструкций

В лекции рассматривается *язык моделирования алгоритмических конструкций* (язык МАК), предназначенный для формирования принадлежащих логическому уровню многоуровневой компонентной цепи СТУС сценариев решения задач исследования и функционального проектирования сложных технических управляемых систем, а также сценариев функционирования устройств управления, представленных алгоритмическими КЦ. Язык МАК, являясь подязыком языка многоуровневых компонентных цепей, расширяет его на класс дискретно-событийных систем, позволяя формировать и анализировать математико-алгоритмические конструкции. Они могут быть представлены как в графической форме, так и в виде совокупности алгоритмических КЦ или в виде совокупности некоторых подпрограмм, которые должны быть автоматически транслированы в формат алгоритмических КЦ.

16.1 Грамматика языка моделирования алгоритмических конструкций

Для описания языка МАК использованы нотации Бэкуса – Наура, нотации языка представления математических выражений и графические нотации языка МАК, разработанные на основе метода компонентных цепей [1]. *Входными данными* программы, сформированной на языке МАК, являются:

- данные измерения характеристик V_{O_R} реальной СТУС, принадлежащей реальному объектному уровню R_O_L , и результаты анализа модели сложной системы V_{O_M} , составляющей модельный объектный уровень M_O_L . Совместно образуя вектор V_0 , осуществляется их передача с объектного на логический уровень;
- команды интерактивного управления C_I , поступающие в сценарий с визуального уровня;
- данные D внешних хранилищ данных, с помощью которых осуществляется параметризация многоуровневой КЦ СТУС.

В результате работы программы или ее отдельных составных частей должны быть получены следующие *выходные результаты*:

- значения параметров компонентов КМ СТУС P_{O_M} и значения управляющих воздействий на реальную СТУС – систему P_{O_R} . Совместно образуя вектор P_O , они передаются с логического уровня на объектный;
- подлежащие визуализации результаты выполнения шагов сценария функционирования устройства управления V_L , сформированного конструкциями языка МАК, передаваемые с логического уровня многоуровневой КЦ СТУС на ее визуальный уровень;
- информация I , полученная в результате выполнения шагов сценария и предназначенная для добавления ее в документы, формируемые автоматически с помощью генератора отчетных форм.

Формально программу, написанную на языке МАК, можно представить в виде:

$$\mathbf{V}_{\text{OUT}} = \langle F_{\text{MAC}} \rangle \mathbf{V}_{\text{IN}}, \quad (16.1)$$

где $\mathbf{V}_{\text{IN}} = [\mathbf{V}_O \quad \mathbf{C}_1 \quad \mathbf{D}]^T$ – вектор входных данных, являющихся аргументами операторов и функций программы $\langle F_{\text{MAC}} \rangle$; $\mathbf{V}_{\text{OUT}} = [\mathbf{P}_O \quad \mathbf{V}_L \quad \mathbf{I}]^T$ – вектор выходных данных, получаемых в результате работы программы $\langle F_{\text{MAC}} \rangle$.

В векторы \mathbf{V}_{IN} и \mathbf{V}_{OUT} входят числовые данные (целочисленные, вещественные и комплексные значения), называемые далее *данные численных типов*, а также *данные нечисловых типов* (логические, символьные (строковые) и массивы данных перечисленных типов).

Программа $\langle F_{\text{MAC}} \rangle$, сформированная на языке МАК, представляет собой совокупность последовательно выполняемых процедур:

$$\langle F_{\text{MAC}} \rangle \rightarrow \langle F_{\text{MAC}_{-1}} \rangle, \langle F_{\text{MAC}_{-2}} \rangle, \dots, \langle F_{\text{MAC}_{-N}} \rangle, \quad (16.2)$$

позволяющих определять элементы вектора \mathbf{V}_{OUT} . Каждая отдельная процедура $\langle F_{\text{MAC}_{-i}} \rangle$ выражается математико-алгоритмическим выражением

$$\langle F_{\text{MAC}_{-i}} \rangle \rightarrow \langle \text{Expr} \rangle, \quad (16.3)$$

где

$$\langle \text{Expr} \rangle \rightarrow \langle \text{Операнд} \rangle \langle \text{Оператор} \rangle \langle \text{Операнд} \rangle \langle \text{Операнд} \rangle \quad (16.4)$$

Рассмотрим множества операторов $\langle \text{Оператор} \rangle$ и операндов $\langle \text{Операнд} \rangle$, являющиеся *нетерминальными символами грамматики языка МАК*. При построении программ на любом формальном языке программирования любой нетерминальный символ грамматики должен быть представлен одним или совокупностью терминальных символов данного языка.

16.2 Операторы языка МАК

Множество операторов $\langle \text{Оператор} \rangle$ включает в себя арифметические $\langle \text{ОА} \rangle$, логические $\langle \text{ЛО} \rangle$ операторы, операторы сравнения $\langle \text{ОС} \rangle$, а также оператор присваивания ($:=$) и оператор вычисления ($=$), определяющие операции для операндов различных типов данных

$$\langle \text{Оператор} \rangle \rightarrow \langle \text{ОА} \rangle | \langle \text{ОС} \rangle | \langle \text{ЛО} \rangle | := | = \quad (16.5)$$

16.3 Операнды языка МАК

Каждый операнд, входящий во множество нетерминальных символов $\langle \text{Операнд} \rangle$ языка МАК, может быть одним из следующих нетерминальных символов

$$\langle \text{Операнд} \rangle \rightarrow \langle \text{Константа} \rangle | \langle \text{Переменная} \rangle | \langle \text{Функция} \rangle | \langle \text{Expr} \rangle. \quad (16.6)$$

Целое математико-алгоритмическое выражение $\langle \text{Expr} \rangle$, стоящее слева или справа от некоторого оператора множества $\langle \text{Оператор} \rangle$, будет являться *составным операндом по отношению к данному оператору*.

16.4 Отображение операторов языка МАК

Множество операторов, к которым относятся арифметические операторы $\langle \text{ОА} \rangle$, операторы сравнения $\langle \text{ОС} \rangle$ и логические операторы $\langle \text{ЛО} \rangle$ отображаются в формате алгоритмических КЦ во множество соответствующих компонентов.

Отображение арифметических операторов. Каждому из арифметических операторов может быть поставлена в соответствие математическая модель вида

$$y = x_1 \langle A \rangle x_2, \quad (16.7)$$

где x_1, x_2 – операнды, вступающие в математическую операцию, предписанную оператором $\langle A \rangle$; $\langle A \rangle$ – один из арифметических операторов множества $\langle \text{ОА} \rangle$.

Операнды, вступающие в операцию, отображаемую оператором $\langle A \rangle$, могут иметь один из численных *типов данных*: целочисленного, вещественного или комплексного. Если в нее вступают два операнда, имеющие разные типы данных, то производится преобразование данных по следующим правилам:

Правило 1. Если в операцию, соответствующую оператору $\langle A \rangle$, вступают данные целочисленного и вещественного типа, то производится операция над ними, результат которой будет принадлежать вещественному типу данных.

Правило 2. Если один из операндов x_1 или x_2 , вступающих в операцию $\langle A \rangle$, принадлежит комплексному типу данных, то второй операнд преобразуется к комплексному типу, для которого

$$\begin{aligned} \operatorname{Re}(x_i) &= x_i \\ \operatorname{Im}(x_i) &= 0, \end{aligned} \quad (16.8)$$

где $i = 1 \vee 2$. Результат операции $\langle A \rangle$ в этом случае будет принадлежать комплексному типу данных.

Математической модели вида (16.8) ставится в соответствие компонент, отображающий соответствующую арифметическую операцию в формате алгоритмических КЦ. Обобщенный вид компонента приведен на рисунке 16.1.

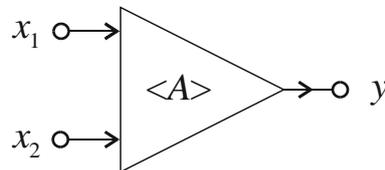


Рис. 16.1 – Обобщенный компонент арифметической операции

Множество арифметических операторов отображается во множество компонентов арифметических операций: сложением, вычитанием, умножением, делением. Графическое обозначение арифметических операций, их имитационные модели и описание принципов работы приведены в инструкции СММ МАРС.

Отображение операторов сравнения. Каждый оператор сравнения $\langle C \rangle$ множества $\langle OC \rangle$ будет описываться математико-алгоритмической моделью вида

$$y = \begin{cases} \text{true}, & \text{если } x_1 \langle C \rangle x_2 \\ \text{false}, & \text{если } (x_1 \langle C \rangle x_2)' \end{cases} \quad (16.9)$$

где x_1, x_2 – операнды целочисленного или вещественного типа данных, вступающие в операцию сравнения, предписанную оператором $\langle C \rangle$; y – переменная

логического типа данных, принимающая значение результата операции сравнения.

Каждой операции сравнения, отображаемой в языке МАК оператором $\langle C \rangle$ из множества операторов сравнения $\langle OC \rangle$, ставится в соответствие компонент, обобщенный вид которого представлен на рисунке 16.2. Множество операторов сравнения $\langle OC \rangle$ отображается во множество компонентов операций сравнения. К ним относятся компоненты, отображающие на языке МАК операции «больше», «меньше», «больше или равно», «меньше или равно», «равно» и «не равно». Графическое обозначение алгоритмических компонентов операторов сравнения, их имитационные модели и описание принципов работы приведены в инструкции СММ МАРС.

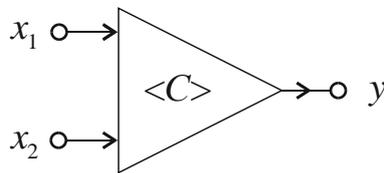


Рис. 16.2 – Обобщенный компонент операции сравнения

Отображение логических операторов. Каждому логическому оператору $\langle L \rangle$ из множества $\langle LO \rangle$, кроме оператора отрицания \neg , ставится в соответствие математическая модель вида

$$y = x_1 \langle L \rangle x_2, \quad (16.10)$$

где x_1 и x_2 – операнды логического типа данных; y – результат выполнения операции, предписанной оператором $\langle L \rangle$, который также приобретает значение логического типа данных.

Каждая логическая операция, представленная в языке МАК оператором $\langle L \rangle$ из множества логических операторов $\langle LO \rangle$, отображается в формате алгоритмических КЦ компонентом, вид которого представлен на рисунке 16.3.

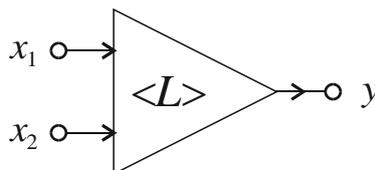


Рис. 16.3 – Обобщенный компонент логической операции

Операции отрицания

$$y = \neg x_1 \quad (16.11)$$

ставится в соответствие оператор «отрицание», представленный на рисунке 16.4.

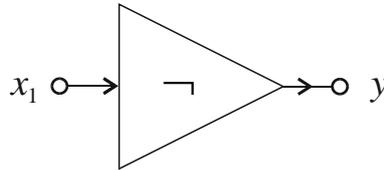


Рис. 16.4 – Компонент операции «отрицание»

Множество логических операторов $\langle ЛО \rangle$, в составе оператора отрицания – логического «НЕ», конъюнкции «И» и дизъюнкции «ИЛИ», отображается во множество компонентов логических операций. Графическое обозначение алгоритмических компонентов логических операций, их имитационные модели и описание принципов работы приведены в инструкции СММ МАРС.

16.5 Отображение функций языка МАК в алгоритмических компонентных цепях

К множеству функций языка МАК относятся:

- *математические функции*, аргументами которых являются скалярные переменные числовых типов данных (целочисленного, вещественного и комплексного);
- *функционалы и блоки обработки*, позволяющие оценивать функции на заданном интервале выбранного аргумента (времени, частоты и др.) некоторым интегральным значением, называемым *параметром-функционалом*;
- *функции пользователя*, имеющие имя, которому ставится в соответствие некоторое математическое выражение относительно аргументов данной функции.

Отображение стандартных математических функций

К стандартным математическим функциям относятся степенные, логарифмические, тригонометрические, обратные тригонометрические, прямые и обратные гиперболические тригонометрические функции. Они подразделяются на логарифмоподобные (ЛПФ) и функции общего вида.

1. *Логарифмоподобными* являются функции, имеющие определенное символическое обозначение и список аргументов:

$$\langle \text{Функция} \rangle \rightarrow \langle \text{Имя_функции} \rangle \langle \text{Список_аргументов} \rangle, \quad (16.12)$$

где

$$\langle \text{Имя_функции} \rangle \rightarrow \langle \text{Имя} \rangle, \quad (16.13)$$

$$\langle \text{Список_аргументов} \rangle \rightarrow \langle \text{Аргумент} \rangle,$$

$$\langle \text{Аргумент} \rangle \langle \text{Список_Аргументов} \rangle, \quad (16.14)$$

$$\langle \text{Аргумент} \rangle \rightarrow \langle \text{Expr} \rangle.$$

2. *Функции общего вида* – функции, каждая из которых имеет специфический графический вид в редакторе математико-алгоритмических конструкций. К ним, в частности, относятся функции, определенные для операторов числовых типов данных и представленные в таблице 16.1. В них черными квадратиками помечены аргументы, т. е. места ввода соответствующих математико-алгоритмических конструкций. Правила их грамматического разбора и интерпретации в алгоритмические КЦ рассмотрены далее.

Таблица 16.1 – Функции общего вида

Знак	Название	Знак	Название
$\sqrt{\blacksquare}$	Корень	$\frac{d}{d\blacksquare}$	Полная производная
$\blacksquare^{\blacksquare}$	Возведение в степень	$\int \blacksquare d\blacksquare$	Определенный интеграл
$\log_{\blacksquare} \blacksquare$	Логарифм по основанию	$\sum_{\blacksquare=\blacksquare} \blacksquare$	Сумма
$\blacksquare!$	Факториал	$\prod_{\blacksquare=\blacksquare} \blacksquare$	Произведение
(\blacksquare)	Взятие в скобки	$\begin{pmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{pmatrix}$	Матрица
$ \blacksquare $	Взятие модуля	$\blacksquare_{\blacksquare}$	Индекс
$\frac{\blacksquare}{\blacksquare}$	Деления (дробь)	$\left\{ \begin{matrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{matrix} \right.$	Ветвление

Для функции задается определенное уникальное имя $\langle \text{Имя_функции} \rangle$ (16.13), которое должно быть зафиксировано в статической таблице идентификаторов. К стандартным функциям общего вида, представленным в таблице 16.1, относятся корень, возведение в степень, логарифм по основанию, факториал, взятие в скобки, взятие модуля, деление (дробь). Остальные функции общего вида будут относиться к функционалам или матрицам.

Каждой стандартной математической функции ставится в соответствие компонент, отображающий ее в формате алгоритмических КЦ, с математической моделью вида

$$y = f(\mathbf{X}), \quad (16.15)$$

где $\mathbf{X} = [x_1 \ x_2 \ \dots \ x_n]^T$ – массив аргументов функции; f – реализованная математическая или математико-алгоритмическая функция; y – результат выполнения функции.

Графическое обозначение алгоритмических компонентов стандартных математических функций, их имитационные модели и описание принципов работы приведены в инструкции к СММ МАРС.

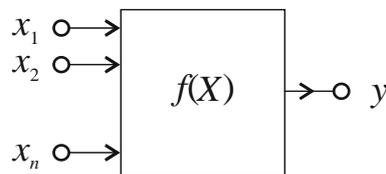


Рис. 16.5 – Обобщенный компонент стандартной математической функции

Отображение блоков обработки данных. Параметрами-функционалами называются численные интегральные значения, характеризующие некоторую функцию на заданном интервале ее аргумента. Их определение осуществляется по определенным алгоритмам, называемым *блоками обработки данных* (БОД). Каждый БОД в формате языка МАК представляется в виде

$$S = F(x, P_1, P_2, f(x, V)), \quad (16.16)$$

где x – варьируемый аргумент функции $f(x, \mathbf{V})$, называемый далее *параметром функционала*, при изменении которого определяется соответствующий параметр-функционал (например, время, частота, сопротивление и т. п.); P_1 – нижний предел изменения параметра функционала x ; P_2 – верхний предел изменения параметра функционала x ; $f(x, \mathbf{V})$ – функция, параметр-функционал которой

подлежит определению; \mathbf{V} – вектор аргументов функции $f(x, \mathbf{V})$, значения которых остаются постоянными при определении параметра-функционала; S – подлежащее определению значение параметра-функционала.

Контрольные вопросы

1. Определите назначение языка МАК.
2. Что является входными данными программы, сформированной на языке МАК?
3. Что получается в результате работы программы МАК или ее отдельных составных частей?
4. Характеризуйте операторы и операнды языка МАК.
5. Виды функций и их отображение в языке МАК.
6. Как отображаются блоки обработки данных в МАК?

Литература

1. Дмитриев, В. М. Среда многоуровневого компьютерного моделирования химико-технологических систем / В. М. Дмитриев, Т. В. Ганджа. – Томск : Изд-во Том. ун-та, 2017. – 332 с.

17 Системная динамика

Идея моделирования сложных систем на самом верхнем уровне абстракции, когда исследователь абстрагируется от индивидуальных объектов системы (сотрудников, машин, документов, товаров) и рассматривает только агрегированные количественные характеристики потоков таких объектов, взаимовлияния и взаимосвязи этих потоков, была предложена Дж. Форрестером 50 лет назад.

Парадигма компьютерного моделирования, при которой для исследуемой системы строятся графические диаграммы причинных связей и глобальных влияний одних параметров на другие во времени, а затем модель, созданная на основе этих диаграмм, имитируется на компьютере, получила название *системная динамика* [1].

Графическая нотация для моделирования всех компонентов системы и их взаимосвязей делают системную динамику (СД) удобным инструментом визуального представления всей системы или организации в целостном виде.

Сложные связи и взаимное влияние процессов часто встречаются в бизнесе, экологии, социальных системах, урбанистике и т. п. Системная динамика оказалась очень *эффективным методом для представления и анализа проблем* динамики организационных систем (анализ рынка, управление проектами, цепочками поставок). Она дает исследователю понимание эффекта, производимого на систему в связи с изменением тех или иных параметров, позволяет сравнить альтернативные решения по управлению системой и выбрать наилучшее из них.

17.1 Нотация и основные идеи системной динамики

Системная динамика выработала свою графическую нотацию для построения структур потоковых диаграмм, представляющих причинно-следственные связи в сложной системе.

Итак, в потоковых диаграммах системной динамики используются *четыре базовых графических объекта*: накопители (уровни, переменные состояния), потоки (связи между накопителями, вентили, регулирующие потоки, функциональные зависимости, определяющие взаимное влияние потоков).

Накопители обозначаются прямоугольниками, *потоки* – направленными переходами, а *вспомогательные переменные* – кружками. Стрелки обозначают причинно-следственные связи в модели.

Накопитель – это параметр, содержащий нечто.

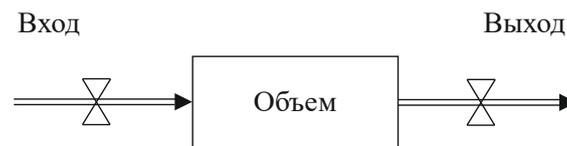
Потоки – это непрерывное перемещение содержимого между накопителями. Накопители и потоки влияют друг на друга через *положительные и отрицательные обратные связи*. Для задания влияния параметров используются *вспомогательные переменные*, обозначаемые кружками. Временная задержка моделируется *блоками задержки*. Параметры-константы в некоторых системах моделирования изображаются ромбиками.

Петли причинных связей отражают влияние некоторого процесса *A* на процесс *B*. Процесс *B*, в свою очередь, может также влиять на *A*, возможно, через длинную цепочку причинно-следственных связей. Изучение процесса *A* и процесса *B* по отдельности невозможно. Только изучение динамики всей системы со всеми ее связями, переменными и временными задержками приводит к корректному пониманию процессов развития системы.

Сложные системы отличаются от простых тем, что в них существует несколько петель ОС и несколько переменных состояния, а причины изменений состояний системы могут лежать далеко от следствий как по времени, так и в пространстве. *Задача моделирования* – выявить реальные причинные зависимости в сложных системах и найти схемы управления, подавляющие нежелательное развитие событий.

17.2 Концепция и терминология системной динамики

В моделях системной динамики представляют накопители некоторого содержимого и анализ изменения их объемов во времени. Содержимое накопителей может иметь любую природу и отражать динамику ресурсов, денег или продуктов в рамках одной модели. При создании модели исследователь абстрагируется от природы содержимого накопителя. Если даже содержимое накопителей является дискретным (материалы, люди, заявки), то его дискретная природа игнорируется. Для переменных модели обычно используются вещественные значения. Вне зависимости от конкретного содержимого накопителя фрагмент структурной схемы имеет обобщенное представление (рис. 17.1).



$$d(\text{Объем})/dt = \text{Вход} - \text{Выход}$$

$$\text{Объем}(t) = \int (\text{Вход}(t) - \text{Выход}(t)) dt + \text{Объем}(t_0)$$

Рис. 17.1 – Обобщенное представление фрагмента структурной схемы

17.3 Формализм метода компонентных цепей для интерпретации моделей системной динамики

Методы компьютерного моделирования, основанные на общей теории цепей [2], нашли широкое применение при исследовании радиоэлектронных, механических, гидравлических систем, систем биомедицинского характера, а также систем автоматического управления [3]. Вполне логичным следует признать их *дальнейшее развитие и адаптацию* в области социально-экономических и эко-номико-экологических систем (ЭЭС) [4–6].

Для моделирования системы нефтегазодобывающего комплекса (НГДК) можно применить метод компонентных цепей и основанную на нем среду моделирования MAPS [2].

Формализованное представление компонентной цепи учитывает вид дифференциальных моделей системной динамики [1]:

$$\frac{dX}{dt} = f(X, U, t), \quad (17.1)$$

$$Y = H(X, U),$$

где $X = (X_1, \dots, X_m)$ – вектор состояния системы; X_1, \dots, X_m – переменные состояния; $U = (U_1, \dots, U_p)$ – вектор входов; $Y = (Y_1, \dots, Y_m)$ – вектор выходов; t – текущее модельное время; H – функция связи входных и выходных переменных.

Общая структура компонентной цепи для моделей системной динамики, к которым можно отнести, например, модели ЭЭС, представлена на рисунке 17.2 [1].

На схеме можно выделить две части:

I – сеть потоков (интеграторов): $X_{t+1} = X_t + hJV_t$;

II – сеть преобразователей: $F(X, V)_t = 0$.

Рассмотрим входящие в общую структуру КЦ для моделей *системной динамики компоненты*:

1. Источник входной переменной (U_1, U_2, \dots, U_p):

исток $\bullet \longrightarrow$;

сток $\longrightarrow \bullet$.

2. Уровни (X_1, X_2, \dots, X_m).

Уровни в КЦ задаются топологическими узлами (рис. 17.3), которым соответствуют переменные X_j .

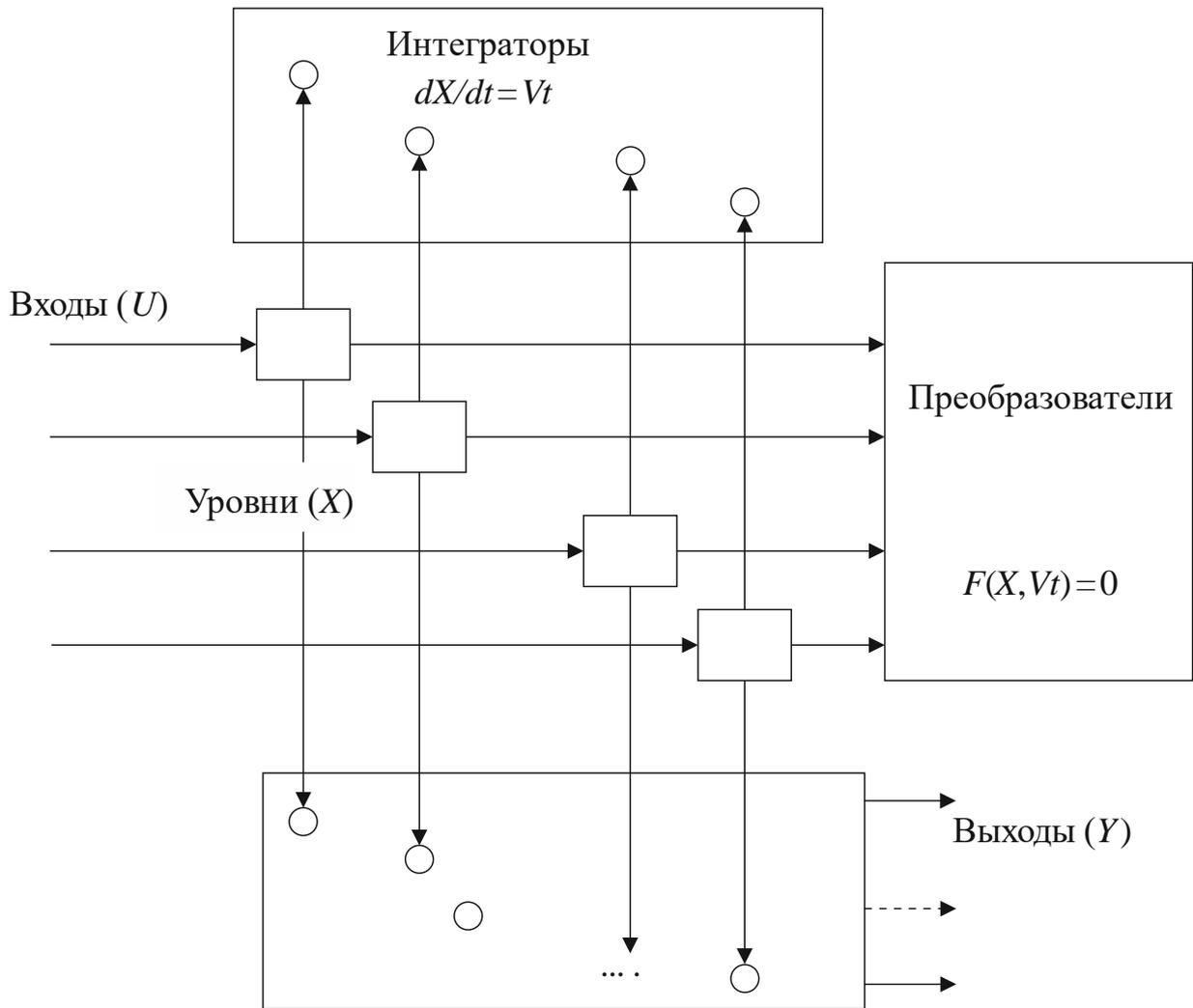


Рис. 17.2 – Общая структура КЦ для моделей системной динамики

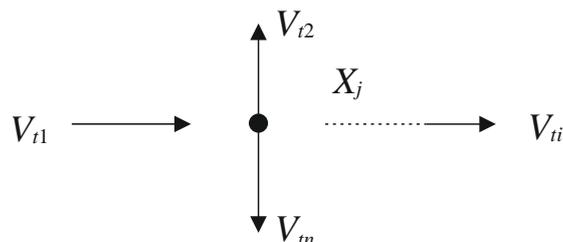


Рис. 17.3 – Топологический узел КЦ

Каждому узлу инцидентно определенное количество ориентированных связей (темпов) с переменными V_{t1}, \dots, V_{tm} .

Для каждого узла (уровня) справедлив узловый топологический закон вида (17.2):

$$\sum_{i=1}^n L_i \cdot V_{ti} = 0, \quad (17.2)$$

где $L_i = \pm 1$ и задает ориентацию связей – темпов. Особого учета требует нулевой узел (уровень) $V_0 = 0$.

3. Преобразователи (F_1, F_2, \dots, F_n) как функции от X_j и V_t .

В отличие от узлов КЦ (рис. 17.3), которые отображают наиболее существенные с точки зрения разработчиков модели состояний выделенных ингредиентов (предмет труда, уровня загрязнения, население и т. д.), ее ветви задают возможные переходы элементов из одного состояния в другое (темпы изменения). Переменные ветвей V_{t1}, \dots, V_{tm} принимаются за характеристики интенсивности (скоростей), с которыми совершаются переходы элементов из состояния в состояние по соответствующим ветвям цепи.

Отображения

$$F_i(X_j, X_{j+1}, V_{ij}) = 0$$

связывают переменные уровней и входные переменные с переменными темпов (рис. 17.4). Эти зависимости могут носить характер уравнений или таблично заданных функций с логическими ключами.

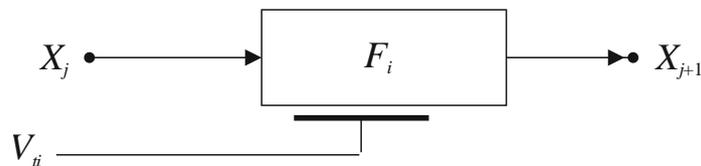


Рис. 17.4 – Связь переменных уровней с переменными темпов

4. Блок интегрирования $(d/dt \leftarrow)$.

Нормативной схемой в моделях системной динамики является одношаговая схема первого порядка:

$$X(t+1) = X(t) + h \cdot A \cdot V_t, \quad (17.3)$$

где A – матрица инцидентий размера $(m \cdot n)$, с элементами $(0, +1, -1)$;

m – размерность вектора X ;

n – размерность вектора V_t ;

h – шаг схемы интегрирования.

5. Общая структурная схема управления.

Модель системной динамики для КЦ имеет вид:

$$\frac{dX}{dt} = V_x, \quad \text{Сеть потоков} \quad (17.4)$$

$$AV_t = 0 \quad \text{Топологический блок}$$

$$F(X_t, V_t) = 0 \quad \text{Компонентный блок} \quad (17.5)$$

$$Y_t = H(X_t, U_t) \quad \text{Сеть информации}$$

Пример. Гидросистема из двух баков

Построим имитационную модель системы из двух баков (рис. 17.5).

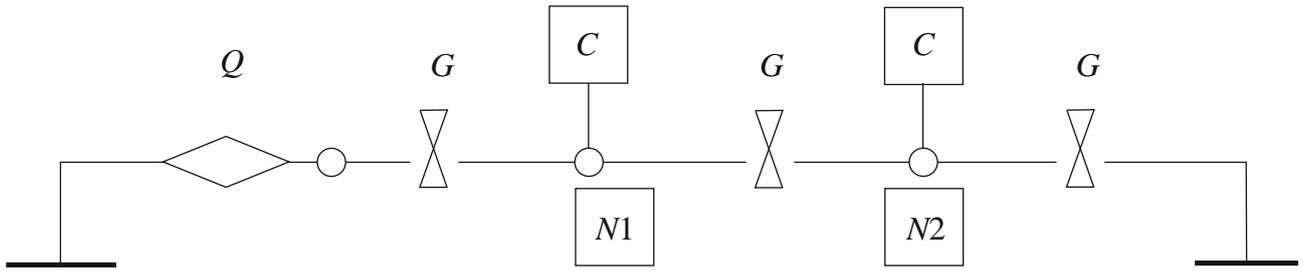


Рис. 17.5 – Модель системы из двух баков и трех вентилей. Q – источник потока; G – вентиль; C – гидроемкость; $N1, N2$ – узлы – уровни жидкости

В ветвях данной цепи циркулируют потоковые переменные (объемные расходы), а *в узлах* – давления, пропорциональные уровням жидкости в гидроемкостях.

18 Базовая модель системной динамики экономико-экологической системы

Представим *базовую модель системной динамики* экономико-экологической системы. Компьютерная модель дает менеджерам возможность *установить соответствие* между продукцией, производственными процессами, ресурсами и затратами на производство. Она *позволяет решить задачу* распределения ресурсов предприятия исходя из требований потребителя, требований к продукции и к ее производству, в том числе исходя из требований и ограничений экосистемы, а также *найти баланс* между потребностью предприятия в увеличении рентабельности производства и сохранением экосистемы, что соответствует парадигме устойчивого развития регионов.

Основная идея, заложенная в предлагаемую базовую модель системной динамики, состоит в том, что свойства производимого продукта определяют его *себестоимость*. Производимые продукты и услуги являются источниками издержек для функций; производственные процессы – источники издержек для ресурсов. Управляем тем, что производится, и себестоимость изменяется как следствие. При компьютерном моделировании задаются требования, указываются значения факторов, влияющих на себестоимости продукции и рентабельность производства, и на основании заданных значений осуществляется рациональное распределение функциональных ресурсов, относительно необходимого и достаточного качества продукта. Для построения модели использованы методический аппарат компонентных цепей, методы системной динамики и функционально-стоимостного анализа [1]. *Модель предназначена* для реализации в среде компьютерного моделирования МАРС (СКМ МАРС). *Актуальность исследований* обусловлена необходимостью развития отечественных средств компьютерного моделирования больших систем управления.

Базовая модель ЭЭС в формате компонентных цепей

В качестве основных уровней управления экономико-экологической системы взяты следующие *6 уровней*:

- уровень готовой продукции V_t ;
- уровень процессов и функций, выполняемых при производстве продукции F_t ;

- уровень загрязнения экосистемы C_t ;
- уровень ресурса RS_t ;
- уровень фонда Φ_t ;
- уровень прибыли P_t .

Каждому из этих уровней соответствует *определенная уровневая переменная*: показатель объема выпускаемой продукции; показатель объема выполненных операций, функций, процессов; показатель загрязнения компонентов природной среды; запасы ресурсов в фонде предприятия, объем фондов производства; объем прибыли от продаж продукции.

1. *Уровень готовой продукции* V_t является переменной уровня. Объем готовой к реализации продукции увеличивается в результате ее производства и снижается за счет продаж.

Уровень готовой продукции V_t за время t можно определить из уравнения:

$$V_t(K) = V_0 + \int (PV_t - TV_t) dt, \quad (18.1)$$

где $V_t(K)$ – объем готовой к реализации продукции, ед. продукции (например, количество, шт.; концентрация нефти в грунте, мг/кг); K – качественные характеристики, параметры продукции; PV_t – темп производства в период времени t , ед. продукции за t ; TV_t – темп продаж в период времени t , ед. продукции за t ; V_0 – объем готовой к реализации продукции в наличии на начало времени t , ед. продукции.

2. *Уровень процессов и функций, выполняемых при производстве продукции* F_t являются переменной уровня. Они уменьшаются за счет расхода функций при изготовлении продукции и увеличиваются за счет приобретения либо высвобождения. Вид и качество продукции определяют то, какие функции будут востребованы.

Запасы функций (услуг) в фонде предприятия в момент времени t и при $PR_t = 0$ определяются уравнением:

$$dF / dt = PF_t - TF_t, \quad (18.2)$$

где F_t – уровень доступных функций, ед. функций; PF_t – приобретение либо высвобождение функций на период t ; TF_t – расход функций за период t , ед. функций на ед. продукции (например, 6 мин на 1 ед. продукции).

Расход функций TF_t задается функцией:

$$TF_t = f(K, P),$$

где K – параметры продукции; P – качественные характеристики, параметры продукции.

3. *Уровень загрязнения природной среды* C_t является переменной уровня. Уровень загрязнения изменяется за счет воздействия на него со стороны производства. Это воздействие может быть, как со знаком «плюс» – при загрязнении, так и со знаком «минус» – при компенсирующем воздействии. Если на предприятиях используют ресурсосберегающие технологии, выполняют мероприятия по предотвращению и ликвидации последствий техногенной нагрузки, то уровень загрязнения относительно фонового может не изменяться или изменяться в пределах нормы.

Изменение уровня загрязнения за время t определяется уравнением:

$$dC / dt = SC_t,$$

где C_t – уровень загрязнения, ед. загрязнения; SC_t – воздействие на экосистему со стороны производства в период времени t , ед. загрязнения/ t .

Воздействие на экосистему SC_t задается функцией:

$$SC_t = F(TF_t, TRS_t, P), \quad (18.3)$$

или определяется по формуле:

$$SC_t = PC_t - TC_t,$$

где TF_t – расход функций за период t , ед. функций на ед. продукции; TRS_t – расход ресурса за период t , ед. ресурса/ t ; PC_t – загрязнение в период времени t , ед. загрязнения/ t ; TC_t – компенсирующее воздействие в период времени t , ед. загрязнения/ t .

Ограничения по уровню загрязнения:

$$C_t < C_{\text{пдк}},$$

где $C_{\text{пдк}}$ – предельно-допустимая концентрация веществ, примесей в компонентах природной среды.

Уровень загрязнения за время t с учетом фонового загрязнения можно определить из уравнения:

$$C_t = C_{\phi} + \int SC_t dt,$$

где C_t – уровень загрязнения, ед. загрязнения (например, концентрация нефти в грунте, мг/кг); C_{ϕ} – фоновый уровень загрязнения; SC_t – воздействие на экосистему со стороны производства в период времени t , ед. загрязнения/ t .

Ресурсы RS_t являются переменной уровня. Они уменьшаются за счет расхода ресурсов при выполнении функций, процессов и увеличиваются за счет приобретения.

Запасы ресурсов в фонде предприятия в момент времени t и при $PRS_t = 0$ определяются уравнением:

$$dRS / dt = PRS_t - TRS_t, \quad (18.4)$$

где RS_t – уровень ресурсов, ед. ресурса (например, шт., кг); PRS_t – приобретение либо высвобождение ресурса за период t ; TRS_t – расход ресурса за период t , ед. ресурса на t .

Расход ресурса TRS_t задается функцией:

$$TRS_t = f(F_t, K, U, P),$$

где F_t – выполняемая функция в период t ; K – параметры продукции, в том числе U – требования к продукции; P – параметры ресурсов.

5. Фонды Φ_t являются переменной уровня. Они уменьшаются за счет затрат на приобретение ресурсов и увеличиваются за счет инвестиций.

Изменение размера фонда Φ_t в момент времени t определяются уравнением:

$$d\Phi / dt = I_t - Z_t, \quad (18.5)$$

где Φ_t – объем фондов, ед. фондов; I_t – темп инвестирования за период t , ед. фондов на t ; Z_t – темп затрат за период t , ед. фондов на t .

Инвестиции I_t могут быть как внутренние I_t^i , так и внешние I_t^o , и определяются по формуле:

$$I_t = I_t^i + I_t^o.$$

Затраты Z_t задаются функцией:

$$Z_t = f(TRS_t, S)$$

или определяются по формуле:

$$Z_t = TRS_t S,$$

где TRS_t – расход ресурса, ед. ресурса за t ; S – стоимость расхода 1 единицы ресурса, ед. фонда.

В случае использования нескольких ресурсов для выполнения функции затраты на их использование суммируются. В некоторых случаях может быть суммирование с коэффициентами.

В функционально-стоимостном анализе расходы распределяются в соответствии с причинно-следственными связями между функциями и объектами затрат. Эти связи фиксируются при помощи источников издержек [7]. *Продукция* является источником издержек для функций, функции источником издержек для ресурсов, ресурсы – для фондов. На практике при моделировании бизнес-процессов нужно учитывать источники издержек на несколько *уровней*:

- а) *уровень единиц* – на этом уровне рассматриваются источники для каждой выпускаемой единицы продукции;
- б) *уровень партий* – источники связаны уже не с единицами, а с партиями выпускаемой продукции;
- в) *уровень продукта* – здесь указываются источники, относящихся к выпуску отдельного вида продукции вне зависимости от количества выпускаемых единиц и партий;
- г) *уровень предприятия* – описываются источники, которые не имеют прямого отношения к продукции, это общие функции, связанные с работой предприятия в целом. Однако вызванные ими затраты также распределяются в дальнейшем на продукцию.

Затраты на производство продукции (себестоимость) считаются как сумма затрат по всем уровням, включая затраты на основные и вспомогательные процессы, процессы развития.

б. *Прибыль P_t* является переменной уровня. Она уменьшается за счет затрат и увеличивается за получения выручки от продаж.

Изменение размера прибыли P_t в момент времени t определяется уравнением:

$$dP / dt = W_t + L_t - Z_t - N_t - I_t^i - Q_t, \quad (18.6)$$

где P_t – объем прибыли, ед. фонда; W_t – выручка от продаж за период t , ед. фонда на t ; L_t – льготы от государства в период t , ед. фонда на t ; I_t^i – инвестиции внутренние за период t , ед. фонда на t ; N_t – уплаченные налоги за период t , ед. фонда на t ; Q_t – непроизводственные затраты за период t , ед. фонда на t ; Z_t – затраты за период t , ед. фондов на t .

Предложенные выше уравнения (18.1)–(18.6) описывают процессы в экономико-экологической системе.

Один из *основных показателей производства* – это рентабельность. Рентабельность R_t рассчитывается как отношение прибыли P_t к выручке W_t , полученной от продаж. Целевой уровень рентабельности устанавливается предприятием, как правило, для коммерческих предприятий не ниже 15%.

Рентабельность R_t задается функцией:

$$R_t = f(V_t(K), F_t, C_t, R_t, \Phi_t)$$

или определяются по формуле:

$$R_t = P_t : W_t,$$

где R_t – рентабельность, %; P_t – объем прибыли, ед. фонда; W_t – выручка от продаж за период t , ед. фонда.

Приведем схему модели системной динамики ЭЭС, иллюстрирующую взаимосвязь основных уровней модели (рис. 18.1).

На рисунке 18.1 определена взаимосвязь основных уровней модели системной динамики экономико-экологической системы. Свойства продукции определяют темп расхода функций и темп расхода ресурсов. Темп расхода ресурсов управляет состоянием ресурсов R_t , фонда Φ_t и прибыли P_t . Также темп расхода функций влияет на состояние уровня загрязнения C_t . Таким образом, на локальном уровне свойства производимого продукта влияют на его себестоимость и, как следствие, на рентабельность производства, на глобальном уровне производственная деятельность, техногенная нагрузка влияет на состояние экосистемы.

Целевая функция системы:

$$R_t \rightarrow \max, \quad (18.7)$$

Ограничения по качеству продукции:

$$K = K_{\text{тр}},$$

$$K < K_{\text{эко}},$$

где K – качественные характеристики, параметры продукции; $K_{\text{тр}}$ – требуемые значения параметров продукции от потребителя; $K_{\text{эко}}$ – предельно-допустимые значения параметров продукции в соответствии с экологическими нормами.

Ограничения по уровню загрязнения:

$$C_i < C_{\text{пдк}},$$

где $C_{\text{пдк}}$ – предельно-допустимая концентрация веществ, примесей в компонентах природной среды.

Компьютерная модель экономико-экологической системы, созданная с использованием методического аппарата методов системной динамики, интерпретированных в формат метода компонентных цепей, *позволяет* сохранить реальную топологию системы, сопрягать модели подсистем, проводить многоаспектные исследования, организовывать процедуры оптимизации деятельности предприятия в контексте парадигмы устойчивого развития региона.

Модель ЭЭС в формате компонентных цепей *предназначена* для реализации в СКМ MAPS, которая является активно развивающимся отечественным средством компьютерного моделирования больших систем управления.

Компьютерная модель экономико-экологической системы будет полезна менеджерам предприятий при решении такой локальной задачи, как рациональное распределение мощности предприятия, исходя из потребностей предприятия и требований к продукции. Также она *может применяться* при решении глобальной задачи экономико-экологического регулирования, которая заключается в поиске баланса между потребностью предприятия в увеличении рентабельности производства и сохранением экосистемы.

Контрольные вопросы

1. Дайте определение понятия «системная динамика».
2. Определите четыре базовых графических объекта системной динамики.
3. Какую природу имеют накопители и как они отражают динамику процессов?
4. Как накопители и потоки влияют друг на друга?
5. Для чего используются вентили при рассмотрении системной динамики?
6. Как в нотациях системной динамики отражено взаимовлияние переменных в динамических процессах?
7. Охарактеризуйте все 6 уровней в ЭЭС.

Литература

1. Дмитриев, В. М. МАРС – среда моделирования технических устройств и систем / В. М. Дмитриев, А. В. Шутенков, Т. Н. Зайченко, Т. В. Ганджа. – Томск : В-Спектр, 2011. – 278 с.
2. Затик, О. С. Многоуровневое компьютерное моделирование эколого-экономических систем / О. С. Затик // Журнал научных публикаций аспирантов и докторантов. – 2009. – № 4. – С. 136–138.
3. Форрестер, Дж. Мировая динамика / Дж. Форрестер. – М. : Наука, 1978. – 168 с.
4. Дмитриев, В. М. Компьютерная модель эколого-экономической системы нефтегазодобывающего региона для формирования экологических программ / В. М. Дмитриев, Т. В. Ганджа, О. С. Затик // Информатика и системы управления. – 2011. – № 4 (30). – С. 27–39.
5. Затик, О. С. Модели компонентов эколого-экономической системы на примере нефтегазодобывающего комплекса в формате метода компонентных цепей для компьютерного моделирования / О. С. Затик, Т. В. Ганджа // Известия Томского политехнического университета. – 2009. – Т. 314, № 5. Управление, вычислительная техника и информатика. – С. 114–116.
6. Дмитриев, В. М. Архитектура среды моделирования и базовый состав компонентов компьютерной модели эколого-экономической системы / В. М. Дмитриев, Т. В. Ганджа, О. С. Затик, И. Я. Клепак // Инфокоммуникационные и вычислительные технологии и системы : материалы III Междунар. конф. (г. Улан-Удэ, 6–11 сентября 2010 г.). – Улан-Удэ : Изд.-полиграф. комплекс ФГОУ ВПО ВСГАКИ, 2010. – С. 120–123.
7. Описание метода функционально-стоимостного анализа [Электронный ресурс]. – Режим доступа: http://www.cfin.ru/management/controlling/iso_abc.shtml (дата обращения: 10.02.2020).

19 Системы массового обслуживания

19.1 Классификация систем массового обслуживания и их показатели эффективности

Системы, в которых в случайные моменты времени возникают заявки на обслуживание и имеются устройства для обслуживания этих заявок, называются *системами массового обслуживания (СМО)*.

СМО могут быть *классифицированы* по признаку организации обслуживания (рис. 19.1).

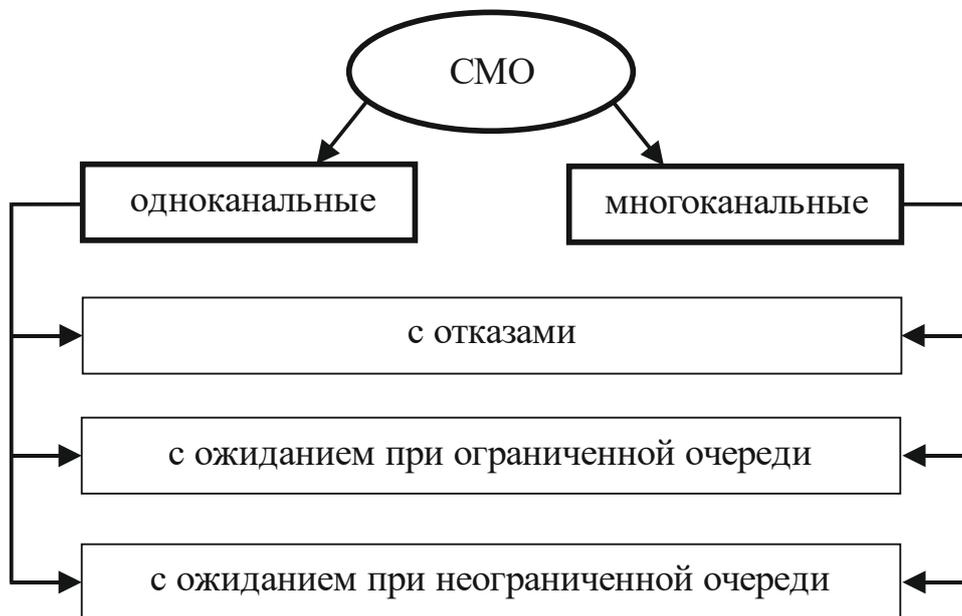


Рис. 19.1 – Классификация СМО

Системы с отказами не имеют очередей.

Системы с ожиданием имеют очереди.

Заявка, поступившая в момент, когда все каналы обслуживания заняты:

- покидает систему с отказами;
- становится в очередь на обслуживание в системах с ожиданием при неограниченной очереди или на свободное место при ограниченной очереди;
- покидает систему с ожиданием при ограниченной очереди, если в этой очереди нет свободного места.

В качестве меры *экономической эффективности* СМО рассматривают сумму потерь времени:

- на ожидание в очереди;
- на простои каналов обслуживания.

Для всех видов СМО используются следующие *показатели эффективности*:

- *относительная пропускная способность* – это средняя доля поступающих заявок, обслуживаемых системой;
- *абсолютная пропускная способность* – это среднее число заявок, обслуживаемых системой в единицу времени;
- *вероятность отказа* – это вероятность того, что заявка покинет систему без обслуживания;
- среднее число занятых каналов – для многоканальных СМО [1].

19.2 Моделирование системы массового обслуживания: основные параметры, граф состояний

При всем многообразии СМО они имеют общие черты, которые позволяют унифицировать их моделирование для нахождения наиболее эффективных вариантов организации таких систем.

Для моделирования СМО необходимо иметь следующие *исходные данные*:

- основные параметры;
- граф состояний.

Результатами моделирования СМО являются вероятности ее состояний, через которые выражаются все показатели ее эффективности.

Основные параметры для моделирования СМО включают:

- характеристики входящего потока заявок на обслуживание;
- характеристики механизма обслуживания.

Рассмотрим *характеристики потока заявок*.

Поток заявок – последовательность заявок, поступающих на обслуживание.

Интенсивность потока заявок λ – среднее число заявок, поступающих в СМО в единицу времени.

Потоки заявок бывают простейшими и отличными от простейших. Для простейших потоков заявок используются модели СМО.

Простейшим, или пуассоновским, называется *поток*, являющийся стационарным, одинарным и без последствий.

Стационарность означает неизменность интенсивности поступления заявок с течением времени.

Одинарным поток заявок является в том случае, когда за малый промежуток времени вероятность поступления более чем одной заявки близка к нулю.

Отсутствие последствия заключается в том, что число заявок, поступивших в СМО за один интервал времени, не влияет на количество заявок, полученных за другой интервал времени.

Для отличных от простейших потоков заявок *используются имитационные модели.*

Рассмотрим *характеристики механизма обслуживания.*

Механизм обслуживания характеризуется:

- числом n каналов обслуживания;
- производительностью канала или интенсивностью обслуживания μ ;
- средним числом заявок, обслуживаемых одним каналом в единицу времени;
- дисциплиной очереди (например, объемом очереди m , порядком отбора из очереди в механизм обслуживания и т. п.).

Граф состояний описывает функционирование системы обслуживания как переходы из одного состояния в другое под действием потока заявок и их обслуживания.

Для построения графа состояний СМО необходимо:

- составить перечень всех возможных состояний СМО;
- представить перечисленные состояния графически и отобразить возможные переходы между ними стрелками;
- взвесить отображенные стрелки, т. е. приписать им числовые значения интенсивностей переходов, определяемые интенсивностью потока заявок и интенсивностью их обслуживания [2].

19.3 Сети Петри для моделирования СМО

Сеть Петри (СП) представляет собой двудольный ориентированный граф, состоящий из вершин двух типов – *позиций* (обозначаемые кружочками) и *переходов* (обозначаемые черточками), соединенных между собой дугами. Вершины

одного типа не могут быть соединены непосредственно. В позициях могут размещаться метки (маркеры), способные перемещаться по сети (рис. 19.2) [3].

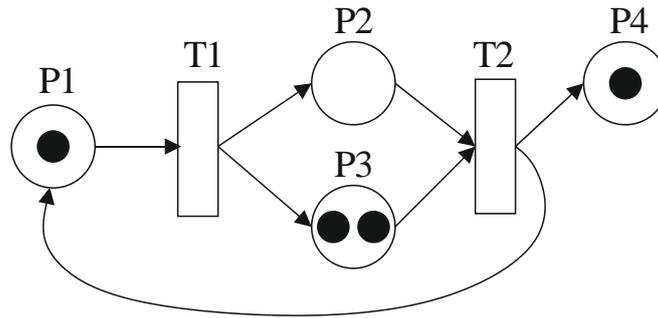


Рис. 19.2 – Пример сети Петри

Событием называют срабатывание перехода, при котором метки из входных позиций этого перехода перемещаются в выходные позиции. События происходят мгновенно либо, при выполнении некоторых условий, одновременно.

Белыми кружками обозначены позиции, полосками – переходы, черными кружками – метки.

Сеть называется *k-ограниченной*, если количество меток в позиции не превышает величины k . Сеть является *безопасной*, если она k -ограничена и $k = 1$. Если метка при попадании в позицию не может ее покинуть при данной разметке, то такая позиция является *ловушкой*. Сеть находится в состоянии тупика, если дальнейшее срабатывание переходов невозможно [4].

Основные определения

Сети Петри и их многочисленные модификации являются одним из классов моделей, неоспоримым *достоинством* которых является возможность адекватного представления не только структуры сложных организационно-технологических систем и комплексов, но также и логико-временных особенностей процессов их функционирования.

Сети Петри представляют собой математическую модель для представления структуры и анализа динамики функционирования систем в терминах «*условие – событие*». Это модель может быть успешно использована для описания так называемых *динамических дискретных систем* различных классов, таких как вычислительные процессы и программы, технологические процессы, информационные, экономические, биологические, социальные и технические системы.

Модели сетей Петри позволяют исследовать работоспособность моделируемых систем, оптимальность их структуры, эффективность процесса их функционирования, а также возможность достижения в процессе функционирования

определенных состояний. Сети Петри и их обобщения являются удобным и мощным средством моделирования асинхронных, параллельных, распределенных и недетерминированных процессов, позволяют наглядно представить динамику функционирования систем и составляющих их элементов. Свойство иерархического вложения сетей Петри позволяет рассматривать модели различной степени детализации, обеспечивая тем самым необходимую композицию сложных систем и процессов.

Несмотря на описанные выше достоинства сетей Петри, *неудобства применения сетей Петри* в качестве языка программирования заключены в процессе их выполнения в вычислительной системе. В сетях Петри нет строгого понятия процесса, который можно было бы выполнять на указанном процессоре. Нет также однозначной последовательности исполнения сети Петри, так как исходная теория представляет нам язык для описания параллельных процессов.

19.4 Структура сети Петри

Сеть Петри S является четверкой: $S = (P, T, I, O)$. $P = \{p_1, p_2, \dots, p_n\}$ – конечное множество позиций, $n \geq 0$. $T = \{t_1, t_2, \dots, t_m\}$ – конечное множество переходов, $m \geq 0$. Множество позиций и множество переходов не пересекаются, $P \cap T = \emptyset$. $I: T \rightarrow P^\infty$ является входной функцией – отображением из переходов в комплекты позиций. $O: P^\infty \rightarrow T$ есть выходная функция – отображение из комплекта позиций в переходы.

Позиция p_i является входной позицией перехода t_j в том случае, если $p_i \in I(t_j)$; p_i является выходной позицией, если $p_i \in O(t_j)$. Входы и выходы переходов представляют собой комплекты позиций. Комплект является обобщением множества, в которое включены многократно повторяющиеся элементы – тиражированные элементы. Использование комплектов, а не множеств для входов и выходов перехода позволяет позиции быть кратным входом либо кратным выходом перехода. Кратность входной позиции p_i для перехода t_j есть число появлений позиции во входном комплекте перехода, $\#(p_i, I(t_j))$. Аналогично кратность выходной позиции p_i для перехода t_j есть число появлений позиции

в выходном комплекте перехода, $\#(p_i, O(t_j))$. Если входная и выходная функции являются множествами (а не комплектами), то кратность каждой позиции есть либо 0, либо 1 [5].

Графическое представление сети Петри

Структура сети Петри представляет собой совокупность позиций и переходов. В соответствии с этим граф сети Петри обладает двумя типами узлов. Круг \circ является позицией, а планка $|$ – переходом.

Ориентированные дуги (стрелки) соединяют позиции и переходы, при этом некоторые дуги направлены от позиций к переходам, а другие – от переходов к позициям. Дуга, направленная от позиции p_i к переходу t_j , определяет позицию, которая является входом перехода. Кратные входы в переход указываются кратными дугами из входных позиций в переход. Выходная позиция указывается дугой от перехода к позиции. Кратные выходы также представлены кратными дугами.

Граф G сети Петри есть двудольный ориентированный мультиграф, $G = (V, A)$, где $V = \{v_1, v_2, \dots, v_s\}$ – множество вершин, а $A = \{a_1, a_2, \dots, a_r\}$ – комплект направленных дуг, $a_i = (v_j, v_k)$, где $v_j, v_k \in V$. Множество V может быть разбито на два непересекающихся подмножества P и T , таких, что $V = P \cup T$, $P \cap T = \emptyset$, и для любой направленной дуги $a_i \in A$, если $a_i = (v_j, v_k)$, тогда либо $v_j \in P$ и $v_k \in T$, либо $v_j \in T$ и $v_k \in P$ (рис. 19.3).

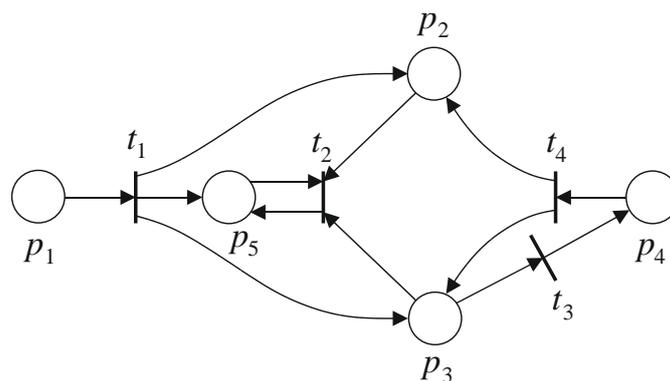


Рис. 19.3 – Граф сети Петри

Матричное представление сети Петри

По отношению к определению сети Петри в виде (P, T, I, O) является определение двух матриц D^- и D^+ , представляющих входную и выходную функции. Каждая матрица имеет m строк (по одной на переход) и n столбцов

(по одному на позицию). Определим $D^-[j, i] = \#(p_i, I(t_j))$, а $D^+[j, i] = \#(p_i, O(t_j))$. D^- определяет входы в переходы, D^+ – выходы. Матричная форма определения сети Петри (P, T, D^-, D^+) эквивалентна стандартной форме, но позволяет дать определения в терминах векторов и матриц.

Маркировка сетей Петри

Маркировка μ есть присвоение фишек позициям сети Петри. Фишка – это примитивное понятие сетей Петри (подобно позициям и переходам). Фишки присваиваются (можно считать, что они принадлежат) позициям. Количество и положение фишек при выполнении сети Петри могут изменяться. Фишки используются для определения выполнения сети Петри.

Маркировка μ сети Петри $C = (P, T, I, O)$ есть функция, отображающая множество позиций P в множество неотрицательных целых чисел N : $\mu: P \rightarrow N$. Маркировка μ может быть также определена как вектор $\mu = (\mu_1, \mu_2, \dots, \mu_n)$, где $n = |P|$ и каждое $\mu_i \in N, i = 1, \dots, n$. Вектор μ определяет для каждой позиции p_i сети Петри количество фишек в этой позиции. Количество фишек в позиции p_i есть $\mu_i, i = 1, \dots, n$.

Связь между определениями маркировки как функции и как вектора очевидным образом устанавливается соотношением $\mu(p_i) = \mu_i$.

Маркированная сеть Петри $M = (C, \mu)$ есть совокупность структуры сети Петри $C = (P, T, I, O)$ и маркировки μ и может быть записана в виде $M = (P, T, I, O, \mu)$. На графе сети Петри фишки изображаются маленькой точкой в кружке, который представляет позицию сети Петри. На рисунке 19.4 приведен пример графического представления маркированной сети Петри.

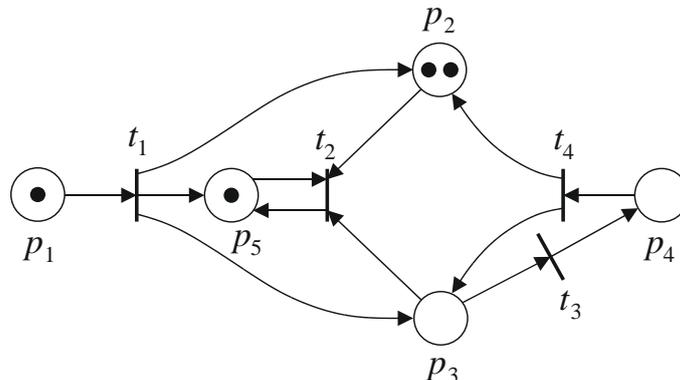


Рис. 19.4 – Маркированная сеть Петри

19.5 Правила выполнения сетей Петри

Сеть Петри выполняется посредством запусков переходов. *Переход запускается* удалением фишек из его входных позиций и образованием новых фишек, помещаемых в его выходные позиции. Переход может запускаться только в том случае, когда он разрешен. *Переход называется разрешенным*, если каждая из его входных позиций имеет число фишек, по крайней мере равное числу дуг из позиции в переход.

Переход $t_j \in T$ в маркированной сети Петри $S = (P, T, I, O)$ с маркировкой μ разрешен, если для всех $p_i \in P$ $\mu(p_i) \geq \#(p_i, I(t_j))$.

Переход запускается удалением всех разрешающих фишек из его входных позиций и последующим помещением в каждую, из его выходных позиций по одной фишке для каждой дуги. Кратные фишки создаются для кратных выходных дуг. Запуск перехода в целом заменяет маркировку μ сети Петри на новую маркировку μ' . Если какая-либо входная позиция перехода не обладает достаточным количеством фишек, то переход не разрешен и не может быть запущен.

Переход t_j в маркированной сети Петри с маркировкой μ может быть запущен всякий раз, когда он разрешен. В результате запуска разрешенного перехода t_j образуется новая маркировка μ' , определяемая следующим соотношением: $\mu'(p_i) = \mu(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j))$.

В качестве примера рассмотрим маркированную сеть Петри, изображенную на рисунке 19.5. При такой маркировке разрешены только три перехода: t_1 , t_3 и t_4 . Переход t_2 не разрешен, так как ни позиция p_2 , ни позиция p_3 , являющиеся входами перехода t_2 , не содержат ни одной фишки. Так как переходы t_1 , t_3 и t_4 разрешены, любой из них может быть запущен. Если запущен переход t_4 , то происходит удаление фишки из каждого входа и помещение фишки в каждый выход. При этом одна фишка удаляется из p_5 , одна фишка помещается в p_3 , а количество фишек в p_4 увеличивается с двух до трех. Новая маркировка, полученная в результате запуска перехода t_4 , показана на рисунке 19.6.

В маркированной сети Петри, изображенной на рисунке 19.5, разрешены только переходы t_1 и t_3 . При запуске перехода t_1 осуществляется удаление фишки из p_1 и помещение фишек в p_2 , p_3 и p_4 (в p_4 – две фишки, так как эта

позиция является кратным выходом перехода t_1). Эта операция образует маркировку, приведенную на рисунке 19.7.

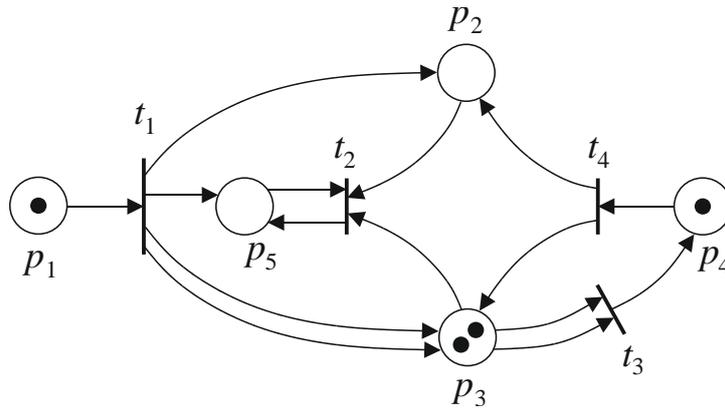


Рис. 19.5 – Маркированная сеть Петри для иллюстрации правил запуска

Переходы t_1, t_3 и t_4 разрешены.

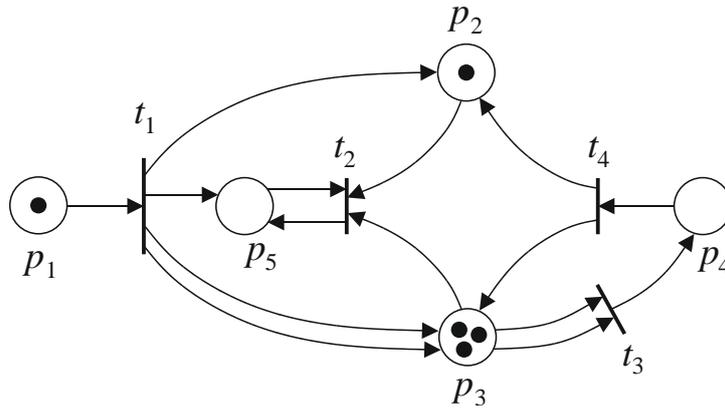


Рис. 19.6 – Маркировка, полученная в результате запуска перехода t_4 в сети на рисунке 19.5

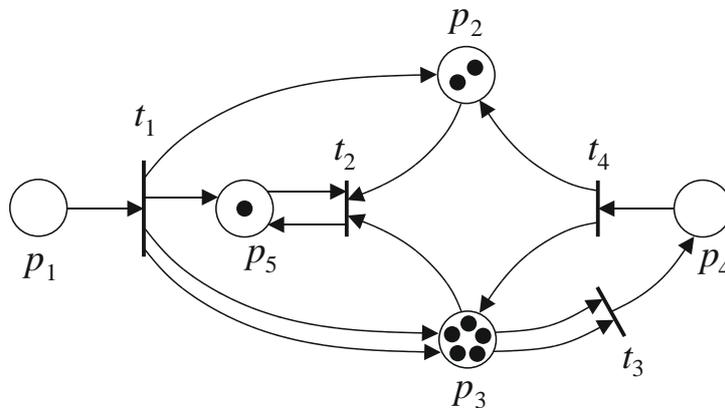


Рис. 19.7 – Маркировка, полученная при запуске перехода t_1 в сети на рисунке 19.6

Запуски могут осуществляться до тех пор, пока существует хотя бы один разрешенный переход. Когда не останется ни одного разрешенного перехода, выполнение прекращается.

Одновременность и конфликт

Приведенные примеры иллюстрируют некоторые особенности сетей Петри и систем, моделируемых с их помощью. Одной из *особенностей* является свойственный сетям и их моделям параллелизм, или одновременность. В модели сети Петри два разрешенных невзаимодействующих события могут происходить независимо друг от друга. Синхронизировать события, пока это не потребуется моделируемой системе, нет нужды. Но когда синхронизация необходима, моделировать ее легко. Таким образом, сети Петри представляются идеальными для моделирования систем с распределенным управлением, в которых несколько процессов выполняются одновременно.

Другая *важная особенность сетей Петри* – это их асинхронная природа. В сети Петри отсутствует измерение времени или течение времени. Это отражает философский подход к понятию времени, утверждающий, что одно из важнейших свойств времени с логической точки зрения, – это определение частичного упорядочения событий. В реальной жизни различные события укладываются в различные интервалы времени, и это отражено в модели сети Петри независимостью от времени управления последовательностью событий. Структура сети Петри такова, что содержит в себе всю необходимую информацию для определения возможных последовательностей событий. Таким образом, на рисунке 19.8 событие «завершение выполнения задания» должно следовать за соответствующим событием «начало выполнения задания». Однако нет и не требуется никакой информации, связанной с количеством времени, необходимым на выполнение задания.

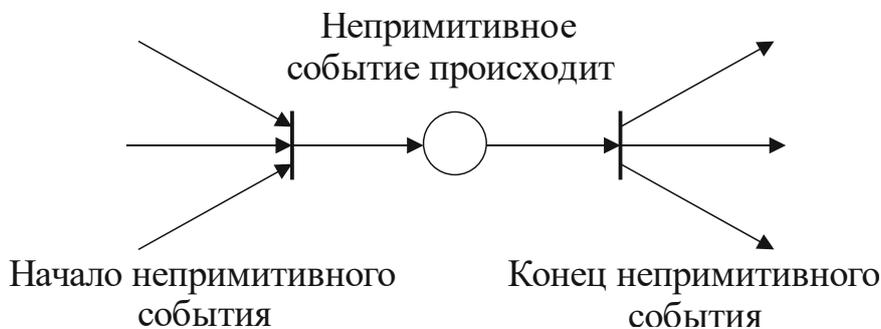


Рис. 19.8 – Моделирование непримитивного события

Выполнение сети Петри (или поведение моделируемой системы) рассматривается здесь как последовательность дискретных событий. Порядок появления событий является одним из возможных, допускаемых основной структурой. Это приводит к явной недетерминированности в выполнении сети Петри. Если в какой-то момент времени разрешено более одного перехода, то любой из нескольких возможных переходов может стать «следующим» запускаемым. Выбор запускаемого перехода осуществляется недетерминированным образом, т. е. случайно. Эта особенность сети Петри отражает тот факт, что в реальной жизненной ситуации, где несколько действий происходит одновременно, возникающий порядок появления событий неоднозначен; скорее может возникнуть любое из множества последовательностей событий. Однако частичный порядок появления события единствен.

Для простоты обычно вводят следующее *ограничение*. Запуск перехода (и соответствующего события) рассматривается как мгновенное событие, занимающее нулевое время, и возникновение двух событий одновременно невозможно. Моделируемое таким образом событие называется *примитивным*; примитивные события мгновенны и не одновременны. *Непримитивными* называются такие события, длительность которых отлична от нуля. Они не являются одновременными и, следовательно, могут пересекаться во времени. Так как осуществление большинства событий в реальном мире занимает некоторое время, то они являются непримитивными событиями и поэтому не могут должным образом моделироваться переходами в сети Петри. Однако это не приводит к возникновению проблем при моделировании систем. Непримитивное событие может быть представлено в виде двух примитивных событий: «начало непримитивного события», «конец непримитивного события» – и условия «непримитивное событие происходит». Эта ситуация может быть промоделирована с помощью сети, показанной на рисунке 19.8.

Рассмотренные ситуации требуют внимательного изучения моделируемых сетями Петри систем для правильного отображения их поведения.

Контрольные вопросы

1. Что называют системами массового обслуживания?
2. Назовите виды СМО.
3. Назовите основные параметры СМО.
4. Что такое сети Петри?
5. Из каких элементов состоит структура сетей Петри?

6. Как графически изображаются сети Петри?
7. Назовите основные правила выполнения сетей Петри.

Литература

1. Ивченко, Г. И. Теория массового обслуживания : учеб. пособие для вузов / Г. И. Ивченко, В. А. Каштанов, И. Н. Коваленко. – М. : Высш. шк., 1982. – 256 с.
2. Ехлаков, Ю. П. Сети Петри в моделировании бизнес-процессов. Теоретические основы и приложения / Ю. П. Ехлаков, О. И. Жуковский, П. В. Сенченко, В. Ф. Тарасенко. – Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2007. – 180 с.
3. Губин, И. Г. Разработка систем автоматизированного проектирования (САПР). Организация системотехнической деятельности при создании САПР : учеб. пособие / И. Г. Губин. – Томск : Томский межвузовский центр дистанционного образования, 2001. – Ч. 1. – 127 с.
4. Питерсон, Дж. Теория сетей Петри и моделирование систем / Дж. Питерсон. – М. : Мир, 1984. – 264 с.
5. Ломазова, И. А. Вложенные сети Петри. Моделирование и анализ распределенных систем с объектной структурой / И. А. Ломазова. – М. : Научный Мир, 2009. – 208 с.

20 Классификация сетей Петри

20.1 Автоматные сети Петри

Автоматные сети Петри (*statemachine*) – сети, у которых переход имеет не более одного входа и не более одного выхода. Такие сети обычно описывают последовательные процессы с ветвлением по условию. Если сеть имеет только одну метку, то сеть является, по сути, графом автомата, который последовательно переходит из одного состояния в другое. Сеть снабжается одной фишкой, расположенной в начальной вершине. Общее число фишек в автоматной сети при переходе от состояния к состоянию не меняется, т. е. SM-сети являются ограниченными, а при наличии одной фишки – безопасными.

20.2 Маркированные сети

Маркированные сети (MG-сети, или *marketgraph*) – сети, у которых каждая позиция имеет не более одного входа и не более одного выхода. С помощью них моделируют последовательно-параллельные процессы. MG-сети называют также *синхрографами*. Переход в синхрографе является *потенциально живым*, если он не входит не в один пустой цикл (не содержит ни одной фишки). Синхрограф является *живым*, если каждый его цикл не пуст при начальной разметке. Живой синхрограф является *безопасным* тогда и только тогда, когда каждое его место входит в определенный цикл, содержащий ровно одну фишку [1].

Маркированной сетью Петри называется набор $\Sigma = (S, T, F, M_0)$, где

1. (S, T, F) – сеть;
2. $M_0 \in \mu S$ – начальная маркировка.

Пример маркированной сети.

На рисунке 20.1 приведен пример маркированной сети. В начальной маркировке место s_1 имеет две метки (токена), место s_3 – одну метку, а места s_2 , s_4 – ни одной метки, т. е. $M_0 = 2s_1 + s_3$.

Сети Петри были разработаны и используются для моделирования параллельных и асинхронных систем. При моделировании в сетях Петри места символизируют какое-либо состояние системы, а переход символизируют какие-то

действия, происходящие в системе. Система, находясь в каком-то состоянии, может породить определенные действия, и наоборот, выполнение какого-то действия переводит систему из одного состояния в другое.

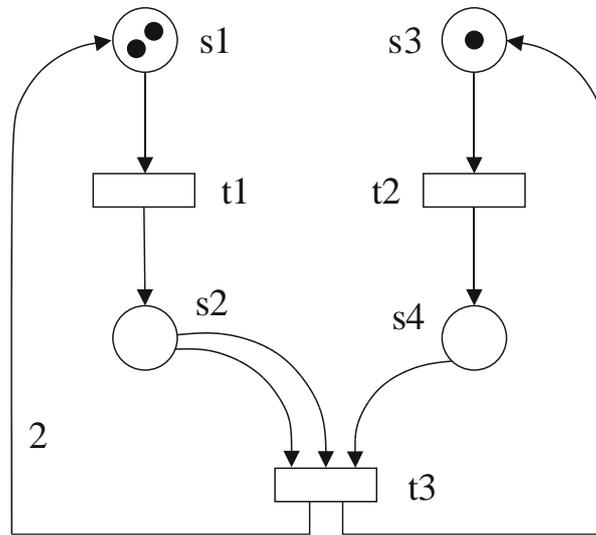


Рис. 20.1 – Пример маркированной сети Петри

Текущее состояние системы определяет маркировка сети Петри, т. е. расположение меток (токенов) в местах сети. Выполнение действия в системе, в сетях Петри определяется как *срабатывание переходов*. Срабатывание переходов порождает новую маркировку, т. е. порождает новое размещение меток (токенов) в сети. Определим функционирование маркированных сетей, основанное на срабатывании отдельных переходов.

Правило срабатывания переходов

Пусть $\Sigma = (S, T, F, M_0)$ – маркированная сеть.

1. Переход $t \in T$ считается *возбужденным* при маркировке $M \in \mu S$, если $M \geq \blacksquare t$;
2. Переход t , возбужденный при маркировке M , может сработать, приводя к новой маркировке M' , которая вычисляется по правилу: $M' = M - \blacksquare t + t \blacksquare$.

Срабатывание перехода обозначается как $M [t > M'$.

Иными словами, переход считается *возбужденным* при некоторой маркировке, если в каждом его входном месте имеется количество меток не менее кратности соответствующих дуг. Возбужденный переход может сработать, причем при срабатывании из каждого его входного места изымается, а в каждое входное добавляется некоторое количество меток, равное кратности соответствующих

дуг. Если одновременно возбуждено несколько переходов, сработать может любой из них или любая их комбинация. Например, пусть в сети на рисунке 20.1 сработают переходы t_1 и t_2 . Получим сеть, представленную на рисунке 20.2.

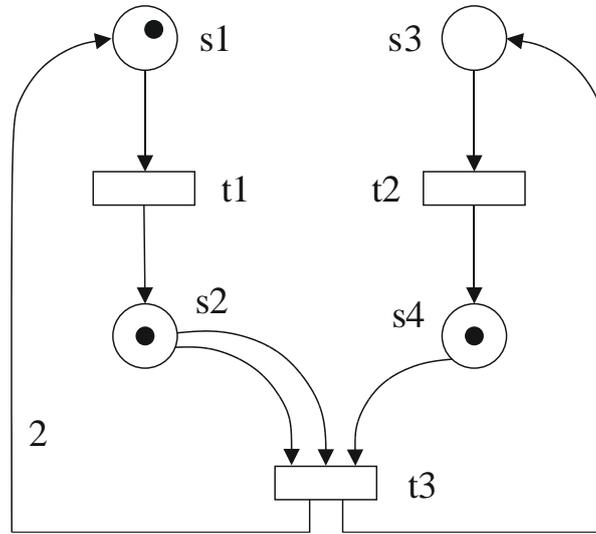


Рис. 20.2 – Новая сеть с маркировкой M'

Композиционный подход к построению сетей Петри предполагает возможность построения более сложных сетей из менее сложных составляющих. Для этого *вводятся точки доступа*, которые позволяют объединять простые сети путем синхронизации событий и состояний (переходов и мест).

Пусть задана сеть $N = (S, T, F)$ и некоторый алфавит $Alph$. T -*точкой доступа* называется набор $\alpha = \langle tid, Alph, \sigma \rangle$, где

1. tid – имя (идентификатор) t -точки доступа;
2. $Alph$ – некоторый алфавит;
3. $\sigma : T \rightarrow \mu Alph \cup \{\tau\}$ – пометочная функция, где $\tau \notin Alph \cup \mu Alph$. Запись $\mu Alph$ обозначает множество всех конечных и непустых мультимножеств, определенных на множестве $Alph$.

Пусть задана сеть $N = (S, T, F)$ тогда s -*точкой доступа* сети N называется набор $\xi = \langle sid, \rho \rangle$, где

1. sid – имя (идентификатор) s -точки доступа;
2. $\rho \subseteq \mu S$ – множество такое, что $\forall M, M' \in \rho. M \leq M' \Rightarrow M = M'$.

Введенные понятия точек доступа предоставляют возможность ввести *две основные операции над сетями Петри* для построения композиционных сетей:

- 1) операция слияния переходов позволяет порождать и описывать синхронизацию параллельных процессов (tmerge) (рис. 20.3);

- 2) операция слияния мест позволяет применять к сетям операции последовательной композиции, выбора, итерации и другие (smerge) (рис. 20.4).

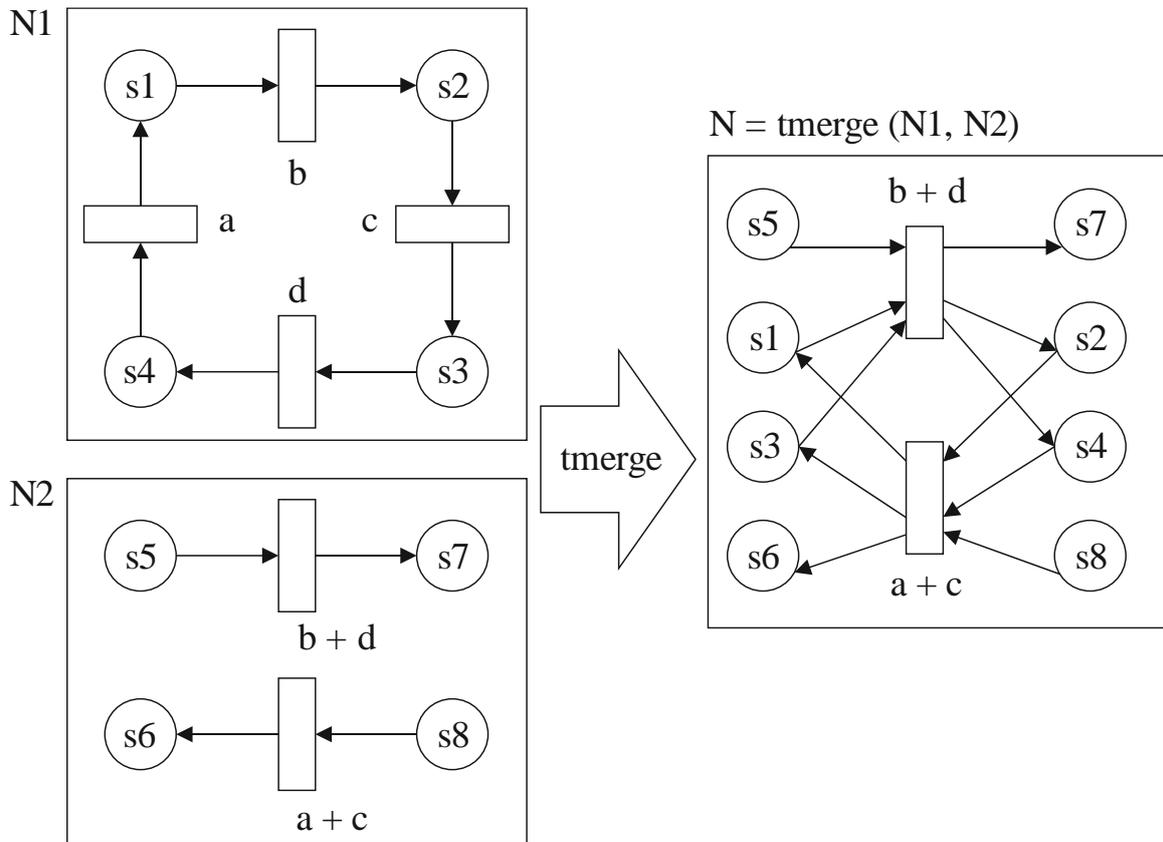


Рис. 20.3 – Пример операции слияния переходов

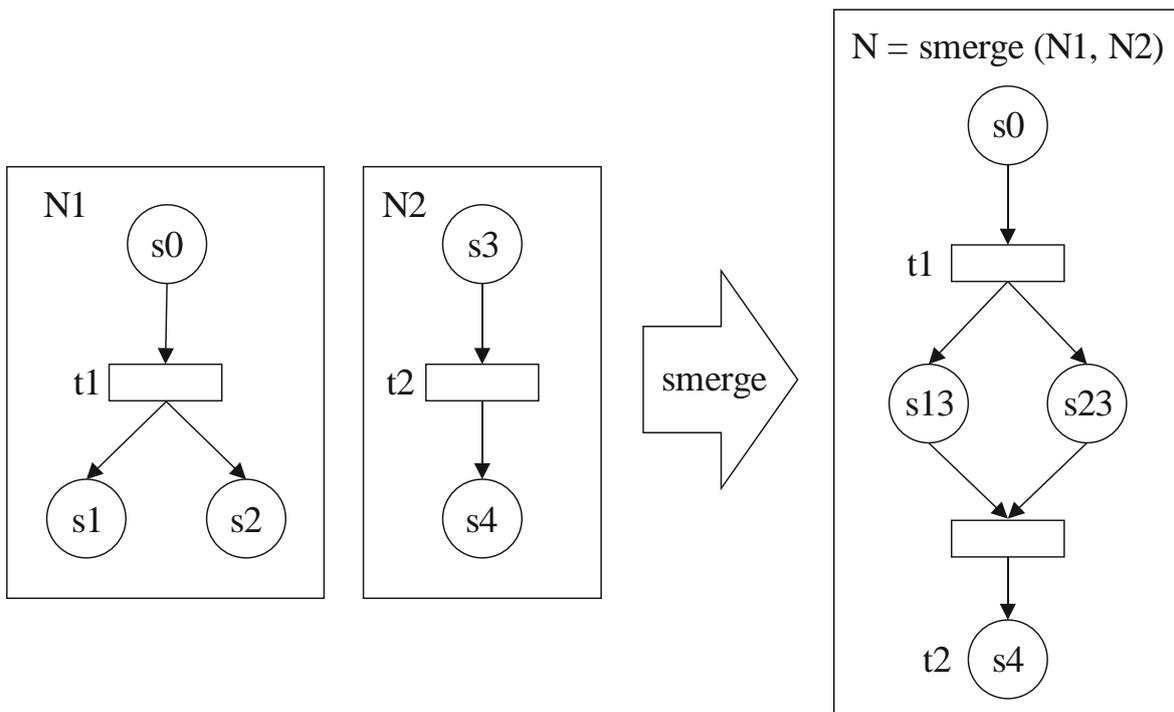


Рис. 20.4 – Пример операции слияния мест

При слиянии мест определяется набор состояний в сети, которые идентифицируются как состояние сети, определенное именем s – точки доступа. Слияние различных сетей производится так, что если в одной сети достигнуто описанное состояние, то в другой сети это состояние также получается достигнутым; при слиянии переходов определяется алфавит событий, видимых из t – точки доступа. Каждый переход в сети помечается либо невидимым событием, либо комбинацией событий из алфавита точки доступа. Слияние по переходам производится так, что если при срабатывании одной сети возникает некоторая комбинация событий, то эта же комбинация событий происходит во второй сети.

Сети свободного выбора (FC-сети, или *freechoice*) – сети, у которых каждая дуга, выходящая из позиции, является либо единственным выходом из нее, либо единственным входом в переход. FC-сети используются для описания процессов управления. Для сетей свободного выбора разработан механизм выявления ловушек и тупиков. Необходимым условием живости сети свободного выбора является то, что тупики должны содержать в себе ловушки. Следовательно, данная сеть живой не является. Путем сети называется последовательность переходов и позиций, связанных направленными дугами. Если начало и конец пути совпадают, то такой путь называется циклом. Критерием близости живой свободной сети является возможность ее покрытия циклами.

Простые сети (SN-сети, или *Simplenets*) – сети, у которых каждый переход может иметь не более одной общей позиции с другими переходами [2].

Простой сетью Петри называется набор $N = (S, T, F)$, где

1. $S = \{s_1, \dots, s_n\}$ – множество мест (позиций);
2. $T = \{t_1, \dots, t_n\}$ – множество переходов таких, что $S \cap T = \emptyset$;
3. $F \subseteq \mu S \times T \times \mu S$ – отношение инцидентности такое, что
 - а) $\forall \langle Q'_1, t_1, Q''_1 \rangle, \langle Q'_2, t_2, Q''_2 \rangle \in F: \langle Q'_1, t_1, Q''_1 \rangle \neq \langle Q'_2, t_2, Q''_2 \rangle \Rightarrow t_1 \neq t_2$;
 - б) $\{t \mid \langle Q', t, Q'' \rangle \in F\} = T$.

Условия в пункте 3 говорят, что для каждого перехода $t \in T$ существует единственный элемент $\langle Q', t, Q'' \rangle \in F$, задающий для него входное мультимножество мест Q' и выходное мультимножество Q'' . Дадим определение входному и выходному мультимножеству.

Входное и выходное мультимножества мест и переходов.

Пусть задана сеть $N = (S, T, F)$.

1. Если для некоторого перехода t имеем $\langle Q', t, Q'' \rangle \in F$, то будем обозначать $\blacksquare t = Q' = \{(s, n) \mid (t, n) \in s^\blacksquare\}$, $t^\blacksquare = Q'' = \{(s, n) \mid (t, n) \in \blacksquare s\}$.
2. $\blacksquare s = \{(t, n) \mid (s, n) \in t^\blacksquare\}$, $s^\blacksquare = \{(t, n) \mid (s, n) \in \blacksquare t\}$.

Будем говорить, что $\|\blacksquare t\|$ – входные, а $\|t^\blacksquare\|$ – выходные места перехода t .

Таким образом, согласно определению, справедливо $\forall t \in T: \langle \blacksquare t, t, t^\blacksquare \rangle \in F$. Далее будем говорить, что место s инцидентно переходу t если $s \in \blacksquare t$ или $s \in t^\blacksquare$.

Расширим функции $\blacksquare(\cdot)$ и $(\cdot)^\blacksquare$ на мультимножества переходов. Пусть $\Theta \in \mu T$ есть мультимножество переходов, такое, что $\Theta = n_1 t_1 + n_2 t_2 + \mathbf{K} + n_k t_k$. Тогда положим

$$\begin{aligned}\blacksquare \Theta &= n_1 \blacksquare t_1 + n_2 \blacksquare t_2 + \mathbf{K} + n_k \blacksquare t_k; \\ \Theta^\blacksquare &= n_1 t_1^\blacksquare + n_2 t_2^\blacksquare + \mathbf{K} + n_k t_k^\blacksquare.\end{aligned}$$

Сети Петри имеют удобную графическую форму представления в виде графа, в котором места изображаются кружками, а переходы – прямоугольниками. Место s соединяется с переходом t , если $(s, n) \in t^\blacksquare$, и t соединяется с s , если $(s, n) \in \blacksquare t$ для некоторого натурального числа $n \in N$. Здесь число n называется *кратностью дуги*, которое графически изображается рядом с дугой. Дуги, имеющие единичную кратность, будут обозначаться без приписывания единицы.

Пример сети Петри.

В качестве простого примера рассмотрим сеть $N = (S, T, F)$, где

1. $S = \{s_1, s_2, s_3, s_4\}$.
2. $T = \{t_1, t_2, t_3\}$.
3. $F = \{\langle s_1, t_1, s_2 \rangle, \langle s_3, t_2, s_4 \rangle, \langle 2s_2 + s_4, t_3, 2s_1 + s_3 \rangle\}$.

В графической форме сеть представлена на рисунке 20.5. Сеть имеет четыре места и три перехода. Отношение F задает дуги сети. Так, например, элемент $\langle 2s_2 + s_4, t_3, 2s_1 + s_3 \rangle$ задает четыре дуги: из s_2 в t_3 и из t_3 в s_1 с кратностями 2, из s_4 в t_3 и из t_3 в s_2 с единичными кратностями. Для перехода t_3 справедливо $\blacksquare t_3 = 2s_2 + s_4$ и $t_3^\blacksquare = 2s_1 + s_3$. Для места s_1 можно вычислить $\blacksquare s_1 = 2t_3$ и $s_1^\blacksquare = t_1$.

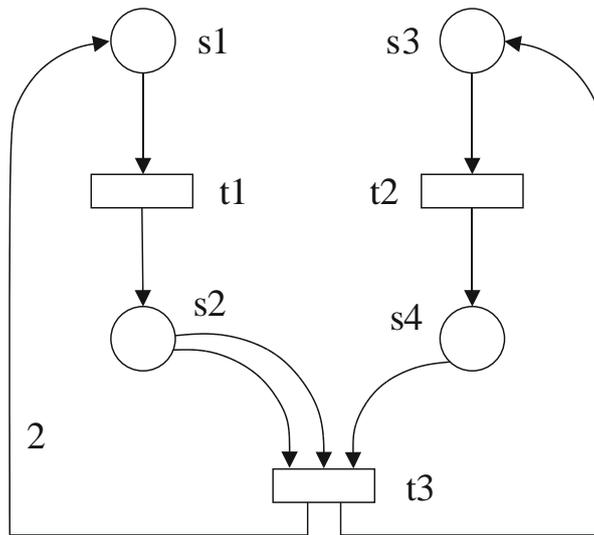


Рис. 20.5 – Пример графа сети Петри

Само по себе понятие сети имеет статическую природу. Для задания динамических характеристик используется понятие *маркировки* сети $M \in \mu S$, т. е. функции $M: S \rightarrow N_0$, сопоставляющей каждому месту целое число. Графически маркировка изображается в виде точек, называемых *метками* (tokens), и располагающихся в кружках, соответствующих местам сети. Отсутствие меток в некотором месте говорит о нулевой маркировке этого места [3].

20.3 Виды расширенных сетей Петри

Ординарные сети (ON-сети, или *Ordinarynets*) – сети, которые не имеют ограничений, кроме одного – кратность дуг должна быть не более единицы. Между узлами прокладывается ровно одна связь. Неординарная сеть может быть преобразована в ординарную. Для этого находят максимальную кратность дуг каждого места и производят размножение позиции в соответствии с установленной кратностью. Эти позиции соединяются друг с другом в кольцо, при этом дуги прорезаются своим переходом. Направление дуг является *однонаправленным* так, чтобы образовывался цикл. Далее восстанавливают связи данной размноженной позиции со всеми переходами. Алгоритм проведения связей жестко не установлен, но связи проводятся так, чтобы они оставались ординарными.

Временные сети Петри (*TimePetriNets*) – сети, у которых с каждым переходом связывают некоторую длительность (время). Для определенности полагают, что изъятие фишек из входных позиций происходит мгновенно, а передача фишек осуществляется за установленное время. В реальности это может соответствовать работе технических устройств и подразделений организации.

Потоковые сети – сети, моделирующие потоковые системы, в которых осуществляется управление данными. Операции выполняются немедленно при готовности данных. В потоковой сети Петри переходы интерпретируются как операторы или вычислительные функции, места интерпретируются как очереди, а данные – как фишки. Если переход имеет n входов, то он реализуется n -местной функцией, которая срабатывает сразу же при наличии фишек во всех входных местах. Данные являются неадресуемыми, другими словами, они содержатся не в центральной, а в распределенной памяти.

Раскрашенные сети (CPN-сети, или *ColouredPetriNets*) – сети, у которых каждая метка имеет свой определенный цвет и переход, связанный с некоторым условием, определяющим наличие связанных с ним входных позиций меток определенного цвета. Цвет метки принято обозначать некоторой буквой. С каждым переходом связывается таблица правил его срабатывания. В таблице переходов столбцы слева от разделяющей линии связываются с входными местами переходов и в совокупности содержат сочетание конкретных фишек, при которых переход может сработать. Столбцы, стоящие справа, указывают на признаки или на вид фишек, которые будут переданы выходным местам [4].

Приведем классификацию расширенных сетей Петри (рис. 20.6).



Рис. 20.6 – Виды расширенных сетей Петри

Рассмотрим подробнее некоторые типы раскрашенных сетей Петри.

Ингибиторная сеть вводит ингибиторные (запрещающие) дуги. С помощью этой сети можно решать различные задачи в сферах электронно-вычислительных машин, их элементов и устройств, сетей ЭВМ, программ и операционных систем, систем сбора и автоматической обработки цифровой информации, системах автоматического управления объектами и процессами, производственных системах дискретного характера, например таких, как сборочные линии и цеха, социально-экономические структуры и т. д.

В *приоритетных сетях* вводят приоритеты срабатывания переходов. Если несколько переходов являются разрешенными, то срабатывает тот из них, который имеет наивысший приоритет. С помощью этой сети можно решать такие задачи, как, например, внеочередной прием срочных телеграмм в отделениях связи; первоочередной ремонт изделий на заводе, которые срочно необходимы для производства; приоритет междугородного звонка перед внутренним; приоритет устранения последствий аварии перед текущим ремонтом изделий и т. д.

В *структурированных сетях* некоторые из переходов являются сложными. При их срабатывании запускается сеть другого уровня иерархии (рис. 20.7).

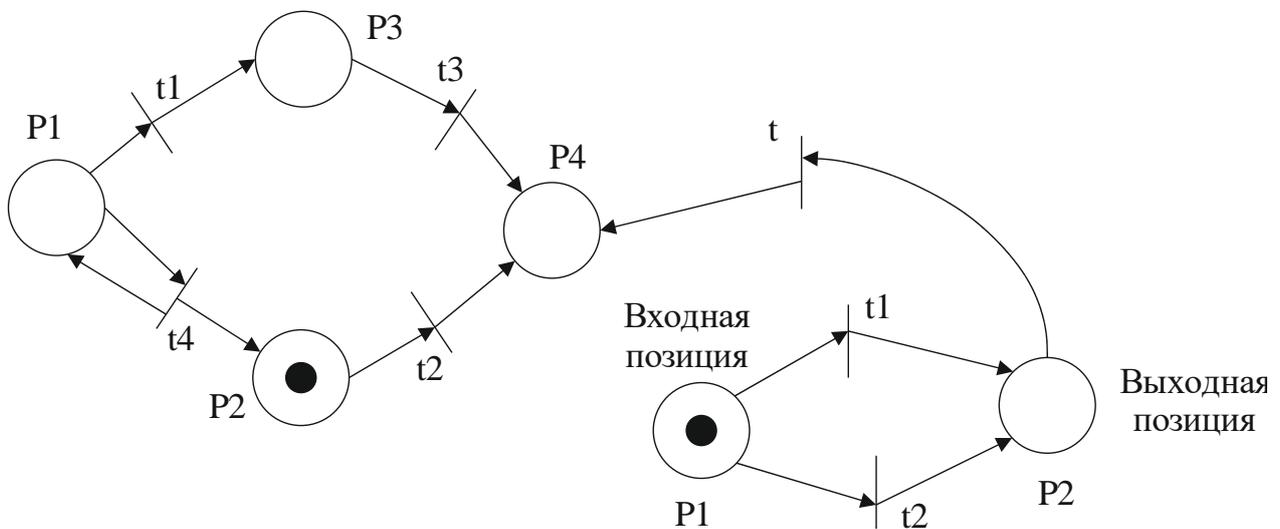


Рис. 20.7 – Структурированная сеть Петри

Срабатывание t_2 приводит к запуску сети другого уровня. Выполнение сложного перехода заключается в помещении во входную позицию по сети фишки. После выполнения сети фишка появляется в ее выходной позиции, затем формируются фишки в выходных позициях сложного перехода.

Преобразование сети к виду, имеющему один вход и один выход, всегда возможно. Такие сети используются для моделирования модульных вычислительных систем.

В цветных сетях вводится понятие цвета для фишек. В общем случае может быть n цветов. В вычислительной технике используются трехцветные сети ($n = 3$). Такие сети используются для моделирования аппаратных средств.

В самомодифицируемых сетях кратность ребра может задаваться либо натуральным числом N , либо определяться количеством фишек, находящихся во входных позициях некоторого перехода.

Качественными характеристиками могут быть: отсутствие зацикливаний в системе, достижение некоторого состояния системы (например, конечного). *Количественными характеристиками* являются: время работы некоторого маршрута в программе, время прохождения сигнала в схеме и т. д.

В стохастических сетях указанные характеристики являются *вероятностными*, т. е. вводится функция плотности вероятности времен срабатывания переходов или времен нахождения фишек в позициях.

Предикатные сети – это сети с логическим описанием состояния системы.

Приведенная классификация является неполной. В последнее время получили развитие новые типы сетей Петри, среди которых отметим *вложенные сети Петри*, которые существенно расширили возможности моделирования сложных систем.

Контрольные вопросы

1. Какие сети называются автоматными?
2. Что подразумевают под маркированными сетями?
3. Назовите виды сетей Петри.
4. В каких сферах применяется моделирование ингибиторными сетями Петри?
5. Какие задачи можно решать с помощью приоритетных сетей Петри?
6. Каковы особенности переходов в структурированных сетях?
7. Назовите количественные и качественные характеристики сетей Петри.

Литература

1. Ехлаков, Ю. П. Сети Петри в моделировании бизнес-процессов. Теоретические основы и приложения / Ю. П. Ехлаков, О. И. Жуковский, П. В. Сенченко, В. Ф. Тарасенко. – Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2007. – 180 с.

2. Котов, В. Е. Сети Петри / В. Е. Котов. – М.: Наука. Главная редакция физико-математической литературы, 1984. – 160 с.
3. Питерсон, Дж. Теория сетей Петри и моделирование систем / Дж. Питерсон. – М. : Мир, 1984. – 264 с.
4. Губин, И. Г. Разработка систем автоматизированного проектирования (САПР). Организация системотехнической деятельности при создании САПР : учеб. пособие / И. Г. Губин. – Томск : Томский межвузовский центр дистанционного образования, 2001. – Ч. 1. – 127 с.

21 Интерпретация сетей Петри в МКЦ

21.1 Основные понятия и определения

С помощью метода компонентных цепей решается задача моделирования и анализа объектов социально-экономической природы.

В тезисном выражении основные характеристики формализма *метода компонентных цепей (МКЦ)* можно представить в виде следующих *составляющих*:

- МКЦ – это объектно-ориентированный язык для моделирования сложных и физически-неоднородных систем с энергетическими и информационными потоками в связях;
- компоненты таких систем могут иметь различную физическую природу (электроника, мехатроника, робототехника, автомобилестроение и др.);
- исследуемый объект представляется в форме компонентной цепи, модель которой строится из моделей независимых компонентов;
- модель компонента формируется автоматически с учетом *четырёх основных аспектов* – топологического, физического, математического (логического) и геометрического – и представляет собой систему алгебро-дифференциальных уравнений в обыкновенных или частных производных. Можно строить модели, содержащие логические соотношения;
- для объектов с функционально обособленными подсистемами введено понятие *структуры* – подцепи, допускающей автономное решение. Здесь четко разделяются непрерывные (уравнения) и дискретные (алгоритмы) процессы;
- форма уравнений компонентной цепи и ее топологическая структура могут меняться в зависимости от поведения переменных или наступления определенных событий [1].

Как описывалось выше, *сеть Петри С* представляется четверкой:

$$C = (P, T, I, O),$$

где $P = \{p_1, p_2, \dots, p_n\}$ – конечное множество позиций, $n \geq 0$;

$T = \{t_1, t_2, \dots, t_m\}$ – конечное множество переходов, $m \geq 0$. Множество позиций и множество переходов не пересекаются, $P \cap T = \emptyset$;

$I: T \rightarrow P^\infty$ является входной функцией – отображением переходов в комплекты позиций;

$O: P \rightarrow T^\infty$ выходная функция – отображение комплектов позиций в переходы (рис. 21.1).

Входы и выходы переходов представляют собой *комплекты позиций*. Комплект (∞) является обобщением множества, в которое включены многократно повторяющиеся элементы – тиражированные элементы. Комплект отличается от множества тем, что может содержать кратные элементы.

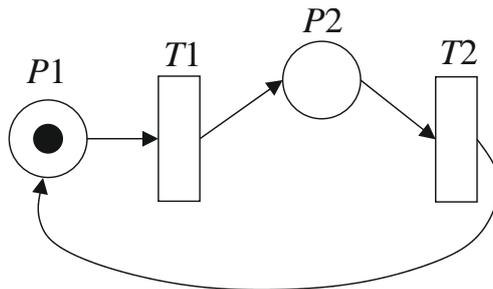


Рис. 21.1 – Пример сети Петри

При преобразовании аппарата сетей Петри в МКЦ компонентная цепь представляется в виде:

$$C_{pn} = (K_{pn}, B_{pn}, N_{pn}, M),$$

где K_{pn} – множество компонентов, (относительно аппарата СП $K_{pn} \sim P$); B_{pn} – множество связей всех компонентов из K_{pn} ; N_{pn} – множество узлов КЦ, образованных при соединении связей компонентов (относительно аппарата СП $N_{pn} \sim T$); M – множество сообщений, передаваемых по связям между компонентами (относительно аппарата СП $MessageFishka = Fishka$).

Графический вид сети Петри в формате МКЦ представлен на рисунке 21.2.

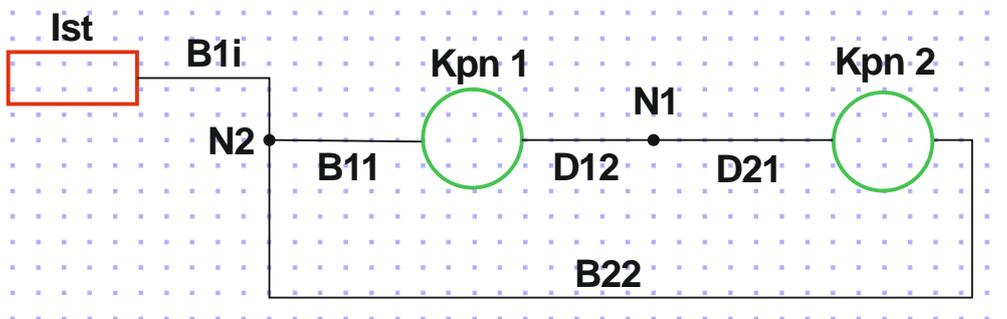


Рис. 21.2 – Графическое представление СП в формате МКЦ

Что касается входной и выходной функций СП, то в формате МКЦ они располагаются во множестве сообщений компонента множества K_p .

Компонент множества K_{pn} :

$$K_{pn} = (P_{pn}, b, n, M),$$

где P_{pn} – множество параметров компонента; b – множество связей компонента; n – множество узлов компонента, $n \sim (SM \text{ и } AM)$; M – множество сообщений, передаваемых и получаемых компонентом, SM и $AM \subset P_{pn}$, где $SM = SendMessage(M, B, N)$ – функция отправки сообщения в узел, $AM = AcceptMessage(M, B, N)$ – функция приема сообщения из узла (относительно аппарата СП $AM \sim I, SM \sim O$).

На рисунке 21.3 изображено графическое представление *обобщенного узла* N_{pn} .

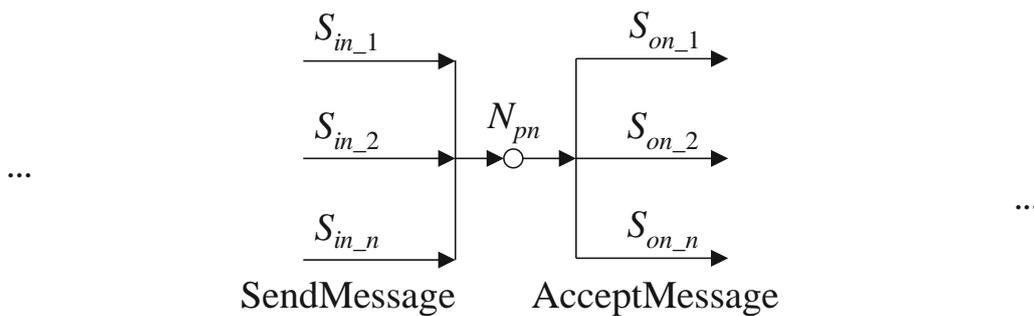


Рис. 21.3 – Графическое представление обобщенного узла

Сообщение M :

$$M = (D, T, B, N),$$

где D – данные любого типа; T – тип данных; B – номер ветви, по которой передается сообщение; N – номер узла, которому передается сообщение.

21.2 Разработка компонентов сетей Петри в СММ МАРС

Генератор моделей компонентов

Создание компонентов производилось в Генераторе моделей компонентов. *Генератор моделей компонентов* (ГМК) предназначен для автоматического формирования программно-алгоритмического кода моделей компонентов среды многоуровневого моделирования МАРС и пополнения ими библиотеки моделей компонентов.

Библиотека моделей компонентов представляет собой совокупность динамически подгружаемых к основному модулю СММ МАРС библиотек (DLL), в которых на объектно-ориентированном языке Visual C++ в виде отдельных классов реализованы модели компонентов.

В СММ МАРС реализованы взаимосвязанные *четыре вида моделирования*:

- *объектный математический*, предполагающий автоматизированное формирование модели исследуемого объекта в виде системы алгебро-дифференциальных уравнений, ее линеаризацию и решение в статическом и динамическом (во временной или частотной области) режимах;
- *объектный имитационный*, предполагающий реализацию статического и динамического имитационного моделирования, а также обработку и обмен информацией между компонентами во времени. Данный вид моделирования может быть применен для объектов с информационными связями, модели компонентов которых представляют собой разнообразные алгоритмы обработки данных. К таким объектам относятся: экологические, экономические, эколого-экономические системы, системы массового обслуживания, моделирование которых основано на аппарате сетей Петри и подобные системы;
- *алгоритмический*, при котором в каждом отдельном компоненте реализуются методы приема входных данных (аргументов) и их преобразование по определенному алгоритму с целью получения и передачи на другие компоненты его результатов либо их визуализации на визуальном слое СММ МАРС;
- *визуальный* – тип моделирования, предназначенный для визуализации результатов вычислительного и алгоритмического видов моделирования, а также для интерактивного управления параметрами исследуемого технического объекта.

В соответствии с указанными типами компьютерного моделирования многослойный редактор СММ МАРС включает в себя три слоя:

- *визуальный*, на котором отображаются компоненты визуального типа моделирования. Их визуальные образы на данном слое изменяют свое изображение в зависимости от полученных данных. Такие компоненты с целью приема информации отображены также на алгоритмическом слое;

- *алгоритмический* – слой, на котором отображаются компоненты алгоритмического уровня моделирования. По соединениям между компонентами данного слоя производится обмен информацией различных типов данных;
- *объектный* – слой, где из моделей компонентов различных физических областей формируется модель исследуемого технического объекта, предназначенная для моделирования в статическом и динамическом (во временной и частотной области) режимах.

ГМК позволяет автоматически сформировать программный код компонента, предназначенный для включения в определенный проект, созданный в среде программирования Microsoft Visual Studio на языке программирования Visual C++.

21.3 Компоненты, реализованные в СММ МАРС

Фишка – примитивное понятие сетей Петри. Фишки используются для определения выполнения сети Петри. *Фишкой* называется единица какого-либо элемента или сообщения, участвующего в процессе.

Компонент «Очередь»

Графическое представление компонента «Очередь» изображено на рисунке 21.4.



Рис. 21.4 – Графическое представление компонента «Очередь»

Алгоритм работы этого компонента: считает, сколько фишек находится в очереди. Время работы задается в свойствах компонента.

Его *математическая модель* имеет вид:

$$count(h+1) = \begin{cases} count(h+1), & \text{если } V_{n1} = 1 \\ count(h), & \text{если } V_{n1} = 0 \end{cases}; \quad count(h+1) = \begin{cases} count(h) - 1, & \text{если } V_{n2} = 0 \\ count(h), & \text{если } V_{n2} = 1 \end{cases};$$

$$V_{n2}(t+h) = \begin{cases} 1, & \text{если } V_{n2}(t) = 1 \text{ или } count \neq 0 \\ 0, & \text{если } V_{n2}(t) = 0 \text{ и } count = 0 \end{cases}.$$

$V_{n1}(t+h) = 0$ – фишку обработали,

где

переменные модели:

внутренние: $count(h)$ – количество фишек;

внешние: V_{n1} – входящие фишки, V_{n2} – фишки на выходе;

параметры модели: h – шаг моделирования, t – текущее время.

Компонент «Оператор»

Графическое представление компонента «Оператор» изображено на рисунке 21.5.

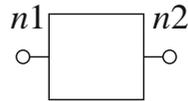


Рис. 21.5 – Графическое представление компонента «Оператор»

Алгоритм работы этого компонента: данный компонент обрабатывает фишки. В свойствах компонента задается время обработки одной фишки.

Математическая модель компонента «Оператор»:

$$T_{\text{вых}} = \begin{cases} t + t_{\text{обсл}}, & \text{если } V_{n1} = 1 \text{ и } F = 0; \\ 0, & \text{если } F = 0 \text{ и } V_{n1} = 0 \end{cases}; \quad F = \begin{cases} 1, & \text{если } F = 1 \text{ или } V_{n1} = 1; \\ 0, & \text{если } F = 0 \text{ и } V_{n1} = 0 \end{cases};$$

$$V_{n1} = 0; \quad V_{n2} = \begin{cases} 1, & \text{если } t = T_{\text{вых}} \text{ и } F = 1; \\ 0, & \text{если } t \neq T_{\text{вых}} \end{cases}; \quad \text{если } V_{n2} = 1, \text{ то } F = 1 \text{ и } t = T_{\text{вых}},$$

где

переменные модели:

внутренние: $t_{\text{обсл}}$ – время обслуживания;

внешние: V_{n1} – входящие фишки, V_{n2} – фишки на выходе, $T_{\text{вых}}$ – время выхода;

параметры модели: t – текущее время, F – обслуживание у оператора.

Компонент «Источник фишек»

Графическое представление компонента «Источник фишек» изображено на рисунке 21.6.



Рис. 21.6 – Графическое представление компонента «Источник фишек»

Алгоритм работы этого компонента: данный компонент подает то количество фишек, которое вы задаете в его свойствах. Также в свойствах компонента вы можете задать время его работы.

Математическая модель компонента «Источник фишек»:

$$V_{n1} = \begin{cases} 1, & \text{если } t = \Delta T \cdot n \\ 0, & \text{если } t \neq \Delta T \cdot n \end{cases}$$

где переменные модели:

внутренние: n – количество фишек, ΔT – промежуток времени, через который поступают фишки;

внешние: V_{n1} – отправляемые фишки;

параметры модели: t – текущее время.

Компонент «Ингибиторная сеть»

Графическое представление компонента «Ингибиторная сеть» изображено на рисунке 21.7.

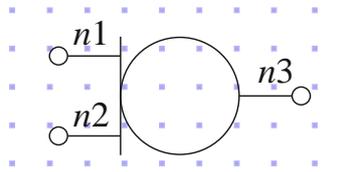


Рис. 21.7 – Графическое представление компонента «Ингибиторная сеть»

Алгоритм работы компонента: если на вход n_1 и n_2 подать 0, то на выход n_3 придет 0; если на вход n_1 подать 0 и на вход n_2 подать 1, то на выход n_3 придет 1; если на вход n_1 подать 1 и на вход n_2 подать 0, то на выход n_3 придет 1; если на вход n_1 и n_2 подать 1, то на выход n_3 придет 0. В свойствах компонента задается время задержки.

Математическая модель компонента «Ингибиторная сеть»:

$$V_{n3}(t) = \begin{cases} 0, & \text{если } V_{n1}(t-h) = V_{n2}(t-h) \\ 1, & \text{если } V_{n1}(t-h) \neq V_{n2}(t-h) \end{cases}; \quad V_{n1}(t) = \begin{cases} 0, & \text{если } V_{n2}(t-h) = 0 \\ 1, & \text{если } V_{n2}(t-h) = 1 \end{cases}; \quad V_{n2}(t) = 0,$$

где

переменные модели:

внешние: V_{n1} – входящие фишки; V_{n2} – входящие фишки; V_{n3} – фишки на выходе;

параметры модели: t – текущее время; h – шаг моделирования.

Компонент «Генератор фишек с приоритетами»

Графическое представление компонента «Генератор фишек с приоритетами» изображено на рисунке 21.8.

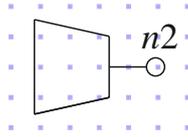


Рис. 21.8 – Графическое представление компонента «Генератор фишек с приоритетами»

Алгоритм работы данного компонента. Данный компонент подает по 1 фишке с приоритетами (приоритет можно задать в свойствах компонента) через определенный промежуток времени. Также в свойствах компонента можно задать промежуток времени между фишками.

Математическая модель компонента «Генератор фишек с приоритетами»:

$$V_{n2} = \begin{cases} \text{приоритет есть, если } t = \Delta T \cdot n \\ \text{приоритет} = 0, \text{ если } t \neq \Delta T \cdot n \end{cases}; \quad n = \begin{cases} n \text{ если } t \neq \Delta T \cdot n \\ (n + 1) \text{ если } t = \Delta T \cdot n \end{cases}$$

где переменные модели:

внутренние: n – количество фишек с приоритетами, ΔT – промежуток времени, через который поступают фишки;

внешние: V_{n2} – фишки с приоритетами;

параметры модели: t – текущее время.

Компонент «Очередь с приоритетами»

Графическое представление компонента «Очередь с приоритетами» изображено на рисунке 21.9.



Рис. 21.9 – Графическое представление компонента «Очередь с приоритетами»

Алгоритм работы: данный компонент выстраивает фишки с приоритетами в очередь, в порядке приоритета (чем меньше значение фишки, тем выше приоритет), а также считает, сколько фишек с приоритетами находится в очереди. В свойствах компонента задается время работы.

Математическая модель компонента «Очередь с приоритетами»:

$$\text{count}(h + 1) = \begin{cases} \text{count}(h + 1), \text{ если } V_{n1} = 1 \\ \text{count}(h), \text{ если } V_{n1} = 0 \end{cases}; \quad \text{count}(h + 1) = \begin{cases} \text{count}(h) - 1, \text{ если } V_{n2} = 0 \\ \text{count}(h), \text{ если } V_{n2} = 1 \end{cases};$$

$$V_{n2}(t+h) = \begin{cases} 1, & \text{если } V_{n2}(t) = 1 \text{ или } count \neq 0 \\ 0, & \text{если } V_{n2}(t) = 0 \text{ и } count = 0 \end{cases};$$

$V_{n1}(t+h) = 0$ – фишку обработали,

где

переменные модели:

внутренние: $count(h)$ – количество фишек с приоритетами;

внешние: V_{n1} – входящие фишки с приоритетами, V_{n2} – фишки с приоритетом на выходе;

параметры модели: h – шаг моделирования; t – текущее время [2].

Компонент «Переход»

Графическое представление компонента «Переход» изображено на рисунке 21.10.

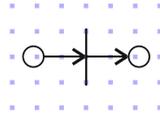


Рис. 21.10 – Графическое представление компонента «Переход»

Алгоритм работы: данный компонент осуществляет переход от одного компонента к другому.

Компонент «Цвет фишек»

Графическое представление компонента «Цвет фишек» изображено на рисунке 21.11.

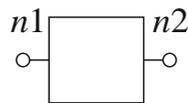


Рис. 21.11 – Графическое представление компонента «Цвет фишек»

Алгоритм работы: принимает фишки и каждой фишке в соответствии присваивает цвет. Например, красный, зеленый или синий. Цвет задается в свойствах компонента.

Математическая модель компонента «Цвет фишек»:

$$V_{n1} = \begin{cases} n, & \text{если } t = \Delta T \cdot n \\ 0, & \text{если } t \neq \Delta T \cdot n \end{cases}; \quad V_{n2} = \begin{cases} c, & \text{если } V_{n1} = n \\ 0, & \text{если } V_{n1} = 0 \end{cases},$$

где переменные модели:

внутренние: n – количество цветных фишек, ΔT – промежуток времени, через который поступают фишки; c – цвет фишек;

внешние: V_{n1} – входящие фишки; V_{n2} – цветные фишки на выходе;

параметры модели: t – текущее время.

Компонент «Предикат»

Графическое представление компонента «Предикат» изображено на рисунке 21.12.



Рис. 21.12 – Графическое представление компонента «Предикат»

Алгоритм работы: делает запрос цветным сетям и принимает от них фишку определенного цвета. Цвет фишки задается в свойствах компонента.

Математическая модель компонента «Предикат»:

$$V_{n1} = \begin{cases} 1, & \text{если } t = \Delta T \cdot n; \\ 0, & \text{если } t \neq \Delta T \cdot n; \end{cases}$$

$$V_{n2} = \begin{cases} 1, & \text{если } V_{n1} = C_3; \\ 0, & \text{если } V_{n1} \neq C_3; \end{cases}$$

где переменные модели:

внутренние: n – количество цветных фишек, ΔT – промежуток времени, через который поступают фишки; c – цвет фишек;

внешние: V_{n1} – входящие цветные фишки; V_{n2} – цветные фишки на выходе;

параметры модели: t – текущее время.

Контрольные вопросы

1. В чем заключается метод компонентных цепей?
2. Из каких составляющих состоит формализм сетей Петри?
3. Как интерпретируется формализм сетей Петри в метод компонентных цепей?
4. Для чего предназначен генератор моделей компонентов?
5. Какие виды моделирования реализованы в СММ МАРС?
6. Какие уровни и слои включает в себя СММ МАРС?
7. Из каких составляющих состоит компьютерная модель компонента?

Литература

1. Дмитриев, В. М. МАРС – среда моделирования технических устройств и систем / В. М. Дмитриев, А. В. Шутенков, Т. Н. Зайченко, Т. В. Ганджа. – Томск : В-Спектр, 2011. – 278 с.
2. Григорьева, Т. Е. Имитационное моделирование сетей Петри с приоритетами / под ред. В. М. Дмитриева // Современные техника и технологии : сб. тр. XIX Междунар. науч.-практ. конф. студентов, аспирантов и молодых ученых (г. Томск, 15–19 апреля 2013 г.). – Томск : Изд-во ТПУ, 2013. – Т. 2. – С. 246–247.

22 Метод нейронных сетей

22.1 Основные определения

Идея нейронных сетей родилась в рамках теории искусственного интеллекта в результате попыток имитировать способность биологических нервных систем обучаться и исправлять ошибки.

Нейронные сети (Neural Networks) – это модели биологических нейронных сетей мозга, в которых нейроны имитируются относительно простыми, часто однотипными элементами (искусственными нейронами).

Нейронная сеть может быть представлена *направленным графом с взвешенными связями*, в котором искусственные нейроны являются *вершинами*, а синаптические связи – *дугами* (синапс – соединение, связь).

Нейронные сети широко *используются* для решения разнообразных задач. Среди областей применения нейронных сетей – автоматизация процессов распознавания образов, прогнозирование, адаптивное управление, создание экспертных систем, организация ассоциативной памяти, обработка аналоговых и цифровых сигналов, синтез и идентификация электронных цепей и систем. *С помощью нейронных сетей* можно, например, предсказывать объемы продаж изделий, показатели биржевого рынка, выполнять распознавание сигналов, конструировать самообучающиеся системы.

Модели нейронных сетей могут быть программного и аппаратного исполнения. Мы будем рассматривать сети первого типа. Если говорить простым языком, слоистая нейронная сеть представляет собой совокупность нейронов, которые составляют *слои*. В каждом слое нейроны между собой никак не связаны, но связаны с нейронами предыдущего и следующего слоев. Информация поступает с первого на второй слой, со второго – на третий и т. д.

22.2 Решаемые задачи

Среди *задач Data Mining*, решаемых с помощью нейронных сетей, будем рассматривать такие:

- *классификация* (обучение с учителем). Примеры задач классификации: распознавание текста, распознавание речи, идентификация личности;

- *прогнозирование*. Для нейронной сети задача прогнозирования может быть поставлена таким образом: найти наилучшее приближение функции, заданной конечным набором входных значений (обучающих примеров). Например, нейронные сети позволяют решать задачу восстановления пропущенных значений;
- *кластеризация* (обучение без учителя). Примером задачи кластеризации может быть задача сжатия информации путем уменьшения размерности данных. Задачи кластеризации решаются, например, самоорганизующимися картами Кохонена. Этим сетям будет посвящена отдельная лекция.

Рассмотрим *три примера задач*, для решения которых возможно применение нейронных сетей.

Медицинская диагностика. В ходе наблюдения за различными показателями состояния пациентов накоплена база данных. Риск наступления осложнений может соответствовать сложной нелинейной комбинации наблюдаемых переменных, которая обнаруживается с помощью нейросетевого моделирования.

Прогнозирование показателей деятельности фирмы (объемы продаж). На основе ретроспективной информации о деятельности организации возможно определение объемов продаж на будущие периоды.

Предоставление кредита. Используя базу данных о клиентах банка, применяя нейронные сети, можно установить группу клиентов, которые относятся к группе потенциальных неплательщиков.

22.3 Элементы нейронных сетей

Искусственный нейрон (формальный нейрон) – элемент искусственных нейронных сетей, моделирующий некоторые функции биологического нейрона. Главная функция искусственного нейрона – формировать выходной сигнал в зависимости от сигналов, поступающих на его входы. В самой распространенной конфигурации входные сигналы обрабатываются адаптивным сумматором, затем выходной сигнал сумматора поступает в нелинейный преобразователь, где преобразуется функцией активации, и результат подается на выход (в точку ветвления). Общий вид искусственного нейрона приведен на рисунке 22.1.

Нейрон характеризуется текущим состоянием и обладает группой *синапсов* – однонаправленных входных связей, соединенных с выходами других нейронов.

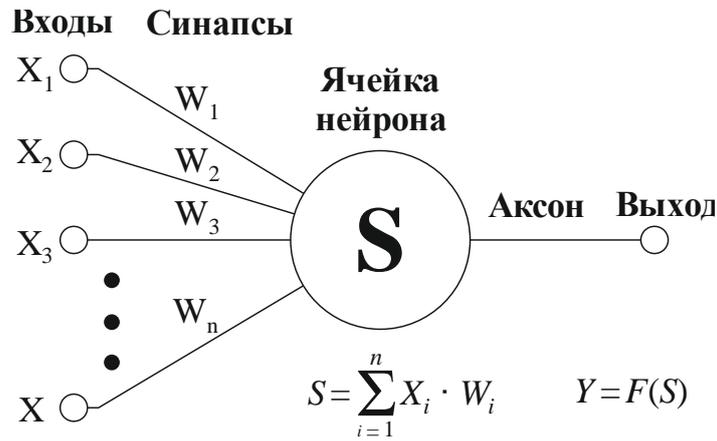


Рис. 22.1 – Искусственный нейрон

Нейрон имеет *аксон* – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Каждый синапс характеризуется величиной синаптической связи (ее весом w_i).

Текущее состояние нейрона определяется как взвешенная сумма его входов:

$$S = \sum_{i=1}^n X_i \cdot W_i.$$

Выход нейрона есть функция его состояния:

$$y = f(s).$$

Активационная функция, которую также называют *характеристической функцией*, – это нелинейная функция, вычисляющая выходной сигнал формального нейрона. *Часто используемые активационные функции*:

- жесткая пороговая функция;
- линейный порог;
- сигмоидальная функция.

Выбор активационной функции определяется спецификой поставленной задачи либо ограничениями, накладываемыми некоторыми алгоритмами обучения.

Нелинейный преобразователь – это элемент искусственного нейрона, преобразующий текущее состояние нейрона (выходной сигнал адаптивного сумматора) в выходной сигнал нейрона по некоторому нелинейному закону (активационной функции).

Точка ветвления (выход) – это элемент формального нейрона, посылающий его выходной сигнал по нескольким адресам и имеющий один вход и несколько выходов. На вход точки ветвления обычно подается выходной сигнал

нелинейного преобразователя, который затем посылается на входы других нейронов.

22.4 Архитектура нейронных сетей

Нейронные сети могут быть *синхронные и асинхронные*.

В *синхронных нейронных сетях* в каждый момент времени свое состояние меняет лишь один нейрон. В *асинхронных* – состояние меняется сразу у целой группы нейронов, как правило, у всего слоя [1]. Можно выделить две базовые архитектуры – *слоистые и полносвязные сети*.

Ключевым в слоистых сетях является понятие слоя. *Слой* – один или несколько нейронов, на входы которых подается один и тот же общий сигнал.

Слоистые нейронные сети – нейронные сети, в которых нейроны разбиты на отдельные группы (слои) так, что обработка информации осуществляется по-слоино. В слоистых сетях нейроны i -го слоя получают входные сигналы, преобразуют их и через точки ветвления передают нейронам $(i + 1)$ -го слоя. И так до k -го слоя, который выдает выходные сигналы для интерпретатора и пользователя. Число нейронов в каждом слое не связано с количеством нейронов в других слоях и может быть произвольным. В рамках одного слоя данные обрабатываются параллельно, а в масштабах всей сети обработка ведется последовательно – от слоя к слою. *К слоистым нейронным сетям* относятся, например, многослойные персептроны, сети радиальных базисных функций, когнитрон, неокогнитрон, сети ассоциативной памяти.

Персептрон – понимание, познание, восприятие. Модель процесса восприятия из 3 слоев (рецепторного (S), преобразующих (A) и регулирующих нейронов (R)).

Однако сигнал не всегда подается на все нейроны слоя. В когнитроне, например, каждый нейрон текущего слоя получает сигналы только от близких ему нейронов предыдущего слоя.

Слоистые сети в свою очередь могут быть *однослойными и многослойными* [2].

Однослойная сеть – сеть, состоящая из одного слоя.

Многослойная сеть – сеть, имеющая несколько слоев.

В многослойной сети первый слой называется *входным*, последующие – *внутренними или скрытыми*, последний слой – *выходным*. Таким образом, промежуточные слои – это все слои в многослойной нейронной сети, кроме вход-

ного и выходного. Входной слой сети реализует связь с входными данными, выходной – с выходными. Таким образом, нейроны могут быть *входными*, *выходными* и *скрытыми*.

Входной слой организован из *входных нейронов (input neuron)*, которые получают данные и распространяют их на входы нейронов скрытого слоя сети.

Скрытый нейрон (hidden neuron) – это нейрон, находящийся в скрытом слое нейронной сети.

Выходные нейроны (output neuron), из которых организован выходной слой сети, выдает результаты работы нейронной сети.

В *полносвязных сетях* каждый нейрон передает свой выходной сигнал остальным нейронам, включая самого себя. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Все входные сигналы подаются всем нейронам.

Обучение нейронных сетей

Перед использованием нейронной сети ее необходимо обучить. Процесс обучения нейронной сети заключается в подстройке ее внутренних параметров под конкретную задачу. Алгоритм работы *нейронной сети* является *итеративным*, его шаги называют *эпохами* или *циклами*.

Эпоха – одна итерация в процессе обучения, включающая предъявление всех примеров из обучающего множества и, возможно, проверку качества обучения на контрольном множестве. Процесс обучения осуществляется на обучающей выборке.

Обучающая выборка включает входные значения и соответствующие им выходные значения набора данных. В ходе обучения нейронная сеть находит некие зависимости выходных полей от входных.

Таким образом, перед нами ставится вопрос – какие входные поля (признаки) нам необходимо использовать. Первоначально выбор осуществляется эвристически, далее количество входов может быть изменено. Сложность может вызвать вопрос о количестве наблюдений в наборе данных. И хотя существуют некие правила, описывающие связь между необходимым количеством наблюдений и размером сети, их верность не доказана. Количество необходимых наблюдений зависит от сложности решаемой задачи. При увеличении количества признаков количество наблюдений возрастает нелинейно, эта проблема носит название «*проклятие размерности*». При недостаточном количестве данных рекомендуется использовать линейную модель. Аналитик должен определить количест-

во слоев в сети и количество нейронов в каждом слое. Далее необходимо назначить такие значения весов и смещений, которые смогут минимизировать ошибку решения. Веса и смещения автоматически настраиваются таким образом, чтобы минимизировать разность между желаемым и полученным на выходе сигналами, которая называется *ошибкой обучения*.

Ошибка обучения для построенной нейронной сети *вычисляется* путем сравнения выходных и целевых (желаемых) значений. Из полученных разностей формируется *функция ошибок*.

Функция ошибок – это целевая функция, требующая минимизации в процессе управляемого обучения нейронной сети. *С помощью функции ошибок* можно оценить качество работы нейронной сети во время обучения. Например, часто используется сумма квадратов ошибок. От качества обучения нейронной сети зависит ее способность решать поставленные перед ней задачи.

Переобучение нейронной сети. При обучении нейронных сетей часто возникает серьезная трудность, называемая *проблемой переобучения (overfitting)*.

Переобучение, или *чрезмерно близкая подгонка*, – излишне точное соответствие нейронной сети конкретному набору обучающих примеров, при котором сеть теряет способность к обобщению. *Переобучение возникает* в случае слишком долгого обучения, недостаточного числа обучающих примеров или перусложненной структуры нейронной сети.

Переобучение связано с тем, что выбор обучающего (тренировочного) множества является случайным. С первых шагов обучения происходит уменьшение ошибки. На последующих шагах с целью уменьшения ошибки (целевой функции) параметры подстраиваются под особенности обучающего множества. Однако при этом происходит «подстройка» не под общие закономерности ряда, а под особенности его части – обучающего подмножества. При этом точность прогноза уменьшается.

Один из вариантов борьбы с *переобучением сети* – деление обучающей выборки на два множества (обучающее и тестовое). На обучающем множестве происходит обучение нейронной сети. На тестовом множестве осуществляется проверка построенной модели. Эти множества не должны пересекаться. С каждым шагом параметры модели изменяются, однако постоянное уменьшение значения целевой функции происходит именно на обучающем множестве. При разбиении множества на два мы можем наблюдать изменение ошибки прогноза на тестовом множестве параллельно с наблюдениями над обучающим множеством. Какое-то количество шагов ошибки прогноза уменьшается на обоих множествах.

Однако на определенном шаге ошибка на тестовом множестве начинает возрастать, при этом ошибка на обучающем множестве продолжает уменьшаться. Этот момент считается концом реального или настоящего обучения, с него и начинается *переобучение*. Описанный процесс проиллюстрирован на рисунке 22.2.

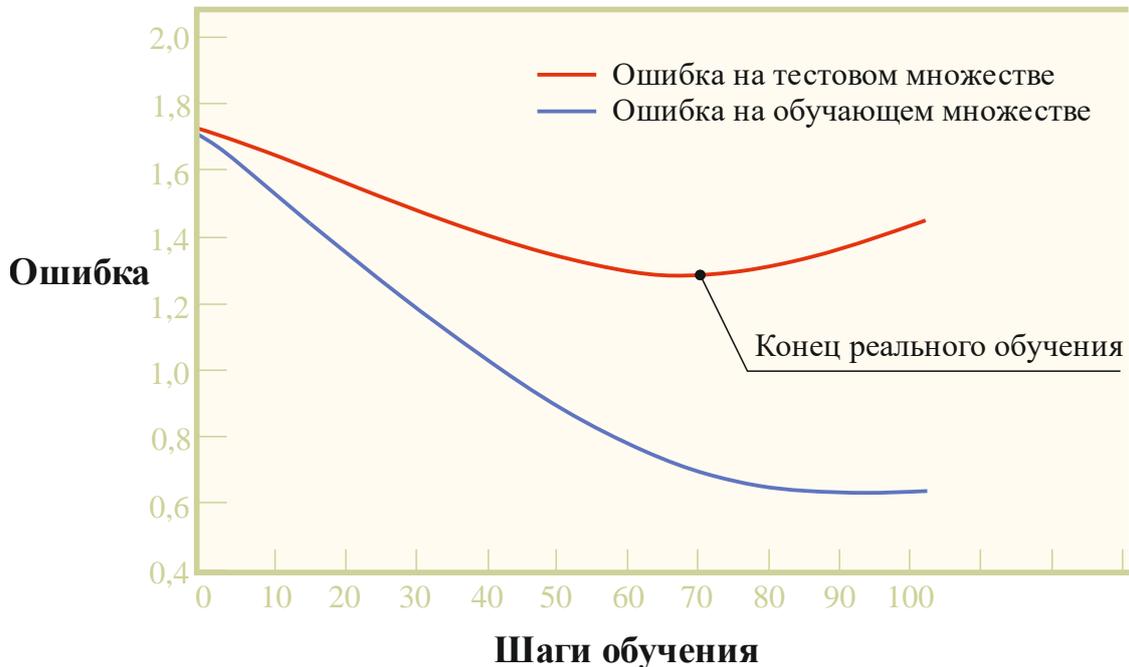


Рис. 22.2 – Процесс обучений сети. Явление переобучения

На первом шаге ошибки прогноза для обучающего и тестового множества одинаковы. На последующих шагах значения обеих ошибок уменьшаются, однако с семидесятого шага ошибка на тестовом множестве начинает возрастать, т. е. начинается процесс переобучения сети. Прогноз на тестовом множестве является проверкой работоспособности построенной модели. Ошибка на тестовом множестве может являться ошибкой прогноза, если тестовое множество максимально приближено к текущему моменту.

Контрольные вопросы

1. Для чего создавались нейронные сети? В чем состоит их основная идея?
2. Дайте определение нейронных сетей.
3. Чем представляется нейронная сеть?
4. Какими могут быть модели нейронных сетей?
5. Назовите ряд задач, решаемых с помощью нейронных сетей.
6. Чем синхронные нейронные сети отличаются от асинхронных?
7. Дайте определение слоя.

8. Дайте определение персептрона.
9. В чем отличие однослойной сети от многослойной?
10. Что такое функция ошибок?
11. С чем связано переобучение в сети?

Литература

1. Дьяконов, В. П. VisSim+Mathcad+MATLAB. Визуальное математическое моделирование / В. П. Дьяконов. – М. : СОЛОН-Пресс, 2004. – 384 с.
2. Технология системного моделирования / под общ. ред. С. В. Емельянова, Е. Ф. Аврамчук, А. А. Вавилова. – М. : Машиностроение, 1988. – 520 с.

23 Статистическое моделирование

23.1 Основные понятия

Основу *статистического моделирования* (СМ) составляет отображение явлений и процессов с помощью случайных (стохастических) событий и их поведений, которые описываются соответствующими вероятностными (статистическими) характеристиками. Напомним, что под вероятностью события понимается $p(A) = m/n$ (где m – число появления события A , n – общее число опытов). *Статистические закономерности* можно представить в виде дискретных случайных величин и их вероятностей или в виде непрерывных закономерностей распределения событий или процессов [1].

Для дискретных событий соотношение между возможными значениями случайных величин X_i и их вероятностями P_i называют *законом распределения* и записывают либо в виде ряда (табл. 23.1), либо представляют в виде зависимостей $F(x)$ (рис. 23.1, а) или $p(x)$ (рис. 23.1, в).

Таблица 23.1 – Набор данных для закона распределения

X	X	X_2	X_i	X_n
$P(X)$	P_1	P_2	P_i	P_n

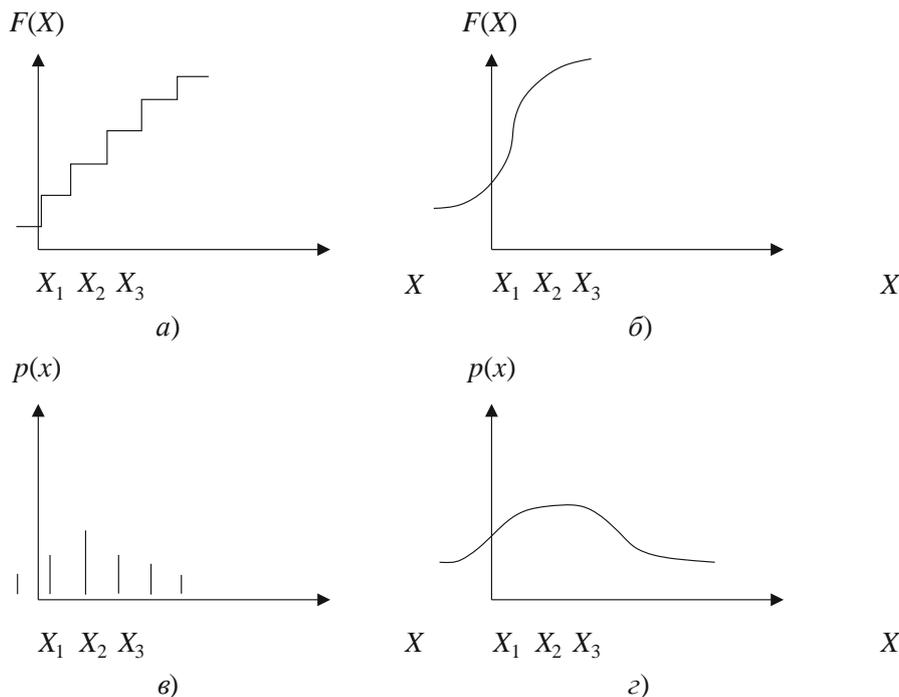


Рис. 23.1 – Представление законов распределения в виде зависимостей

Для непрерывных случайных величин закон распределения представляют либо функцией распределения (интегральный закон распределения (рис. 23.1, б)), либо в виде плотности вероятностей (дифференциальный закон распределения (рис. 23.1, в)). В этом случае

$$P(x) = dF(x) / dx \quad \wedge F(x) = p(x) \wedge x,$$

где $P(x)$ вероятность попадания случайных событий в интервал от x до $x + \Delta x$.

Закон распределения является удобной формой статистического моделирования системы.

Компоненты статистических характеристик

Компоненты статистических характеристик приведены в таблице 23.2.

Арифметическое среднее. Значение характеризует усреднение значения выборки. Для случайных величин имеется также определение математического ожидания.

Оно вычисляется по формуле:

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=0}^{n-1} x_i. \quad (23.1)$$

В компоненте статической обработки x_i заменяется V_{N1} и принимает на вход значения в вещественном виде, n заменяется на S , которое определено как количество элементов в массиве V_{N1} . На выходе компонента формируется число и отправляется на пин V_{N2} .

Медиана. Это возможное значение признака, которое делит ранжированную совокупность на две одинаковые части: 50% «нижних» единиц ряда имеют значение признака больше, чем медиана, а «верхние» 50% – значения признака не меньше, чем медиана.

$$F(X_{0.5}) = F(X < X_{0.5}) = 0.5.$$

На компонент подается вещественное значение. Внутри компонента происходит сортировка данных. В качестве результата в зависимости от количества элементов:

- если количество кратно 2, то возвращается среднеарифметическое двух центральных значений;
- если количество не кратно 2, то возвращается значение, которое находится посередине ряда.

На выходе компонента формируется одно числовое значение.

Дисперсия. Значение разброса случайной величины, т. е. отклонение от математического ожидания:

$$D(X) = M\left(\left(X - M(X)\right)^2\right).$$

На вход подается генеральная совокупность, для которой определяется арифметическое среднее и вычисляется дисперсия. На выходе находится вычисленное значение дисперсии.

Среднеквадратичное отклонение – показатель распространения рассеивания значений случайной величины относительно ее математического ожидания. Измерение производится в единицах измерения самой случайной величины. Среднеквадратичное отклонение равно квадратному корню из дисперсии:

$$s = \sqrt{\frac{n}{n-1} \sigma^2}.$$

В компоненте выполняется вычисление дисперсии, из которой берется квадратный корень. На выходе – вычисленное значение дисперсии.

Максимальное и минимальное значение. Показатель максимального и минимального значения выборки:

$$\begin{aligned} X_{\min} &= \min(X) \\ X_{\max} &= \max(X) \end{aligned}$$

В компонентах выполняется поиск максимального и минимального значения. На выходе получается вещественное число.

Корреляция – Статистическая взаимосвязь двух или нескольких случайных величин. Получаемый коэффициент варьируется от -1 до $+1$ в зависимости от степени зависимости случайных величин между собой:

$$r_{XY} = \frac{\text{cov}_{XY}}{\sigma_X \sigma_Y} = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2 \sum (Y - \bar{Y})^2}}.$$

На данный компонент подаются два источника с данными. На выходе получается коэффициент корреляции.

Таблица 23.2 – Компоненты для обработки генеральных совокупностей

Компонент	Графический вид компонента	Математическая модель
Арифметическое среднее	N1 \circ mean \circ N2	$V_{N2} = \frac{1}{S} \cdot \sum_{j=0}^{S-1} V_{N1, j}$

Медиана	N1 ◯ <input type="text" value="median"/> ◯ N2	
Дисперсия	N1 ◯ <input type="text" value="var"/> ◯ N2	$V_{N2} = M\left(\left(V_{N1} - M(V_{N1})\right)^2\right)$
Среднеквадратичное отклонение	N1 ◯ <input type="text" value="stdev"/> ◯ N2	$V_{N2} = \sqrt{\frac{S}{S-1} D(V_{N1})^2}$
Максимальное значение	N1 ◯ <input type="text" value="max"/> ◯ N2	$V_{N2} = \max(V_{N1})$
Минимальное значение	N1 ◯ <input type="text" value="min"/> ◯ N2	$V_{N2} = \min(V_{N1})$
Мода	N1 ◯ <input type="text" value="mode"/> ◯ N2	
Количество элементов	N1 ◯ <input type="text" value="count"/> ◯ N2	$V_{N2} = S$
Ковариация	N1 ◯ <input type="text" value="cvar"/> ◯ N2	$V_{N2} = \sum(V_{N1} - M(V_{N1}))(V_{N3} - M(V_{N3}))$
Корреляция	N1 ◯ <input type="text" value="corr"/> ◯ N2	$V_{N2} = \frac{c \text{ var}(V_{N1}, V_{N3})}{D(V_{N1})D(V_{N2})}$

23.2 Реализация компонентов класса для функций распределения

Плотность вероятности нормального распределения. Нормальное распределение – это распределение вероятностей, которое задается функцией распределения вида:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

где μ – среднее значение (математическое ожидание) случайной величины, которое указывает координату максимума кривой плотности распределения, а σ^2 – дисперсия, которая должна быть вещественной и строго больше 0. В компонент на пин V_{N1} подаются данные, параметры μ и σ^2 вводятся в панели свойств. По умолчанию эти параметры имеют значения 0 и 1 соответственно. На выходе компонента пин V_{N2} получается массив данных, который можно использовать в дальнейших исследованиях или для построения графика.

Распределение Стьюдента – однопараметрическое семейство абсолютно непрерывных распределений.

$$f(y) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{\pi n} \Gamma(\frac{n}{2})} \left(1 + \frac{y^2}{n}\right)^{-\frac{n+1}{2}},$$

где Γ – гамма-функция Эйлера, n – число степеней свободы распределения. Так как используется гамма-функция и у нас нет цели переходить в комплексную плоскость, то $n > 0$. В компонент подается массив данных, параметр n вводится в панели свойств, по умолчанию $n = 1$.

Распределение Пуассона моделирует случайную величину, представляющую собой число событий, произошедших за фиксированное время, при условии, что данные события происходят с некоторой фиксированной средней интенсивностью и независимо друг от друга.

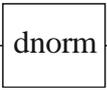
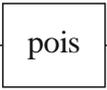
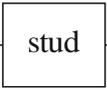
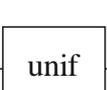
$$f(x) = \frac{\lambda^x}{x!} e^{-\lambda},$$

где $\lambda \in [0; \infty)$ и $x = \{1.2.3...\}$. Параметр λ вводится в панели свойств компонента.

Равномерное распределение характеризуется тем, что вероятность любого интервала зависит только от его длины.

$$f(x) = \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & x \notin [a, b] \end{cases}.$$

Таблица 23.3 – Компоненты для обработки генеральных совокупностей

Компонент	Графический вид компонента	Математическая модель
Плотность вероятностей нормального распределения	N1  N2	$V_{N2} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(V_{N1} - \mu)^2}{2\sigma^2}\right)$
Плотность вероятностей Пуассона	N1  N2	$V_{N2} = \frac{\lambda^{V_{N1}}}{V_{N1}!} e^{-\lambda}$
Плотность вероятностей распределения Стьюдента	N1  N2	$V_{N2} = \frac{\Gamma((n+1)/2)}{\sqrt{\pi n} \Gamma(n/2) (1 + V_{N1}/n)^{(n+1)/2}}$
Плотность вероятностей равномерного распределения	N1  N2	$V_{N2} = \begin{cases} \frac{1}{b-a} & a \leq V_{N1} \leq b \\ 0 & V_{N1} < a, V_{N1} > b \end{cases}$

Плотность вероятностей распределения Вейбула	$N1 \circ \boxed{wb} \circ N2$	$V_{N2} = \begin{cases} (k/\lambda)(V_{N1}/\lambda)^{(k-1)} e^{-(V_{N1}/\lambda)^k} & V_{N1} \geq 0 \\ 0 & V_{N1} < 0 \end{cases}$
--	--------------------------------	--

23.3 Метод Монте-Карло

В общем случае исследуемая система содержит ряд элементов, обладающих некоторой неопределенностью. Подобные системы обычно называют *стохастическими (вероятностными)*, т. к. их поведение не может быть полностью предсказано заранее. При имитации стохастических систем требуется описывать изменчивость элементов в терминах теории вероятностей. Не следует также забывать, что результаты, полученные с помощью имитационной модели, носят вероятностный характер и требуют статистической интерпретации.

В основе стохастического моделирования лежит метод статистических испытаний – *метод Монте-Карло (ММК)*. Следует сразу выделить *два основных класса задач*, решаемых ММК.

Во-первых, ММК позволяет моделировать любой процесс, на протекание которого влияют случайные факторы. Во-вторых, для многих математических задач – детерминированных, т. е. не связанных с какой-либо вероятностью, – можно искусственно придумать вероятностную модель (и даже не одну), позволяющую решать эту задачу.

Примером такой задачи является вычисление определенного интеграла

$I = \int_a^b f(x) dx$, который геометрически равен площади заштрихованной фигуры (рис. 23.2).

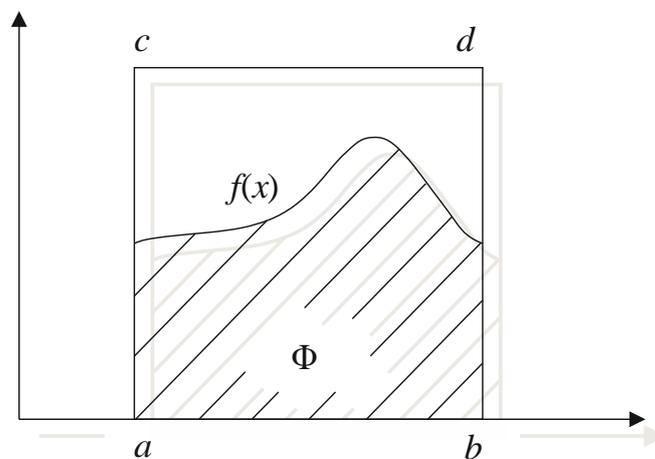


Рис. 23.2 – Интегрирование методом Монте-Карло

Рассмотрим теперь следующую *вероятностную модель*:

1. Допустим, мы умеем моделировать на декартовой плоскости точки, равномерно распределенные в прямоугольнике $abcd$.
2. Пусть мы проводим этот опыт N раз, из них n точек попали в заштрихованную фигуру Φ , тогда из геометрического определения вероятности и чисто интуитивно можно записать:

$$\frac{S(\Phi)}{S_{abcd}} \approx \frac{n}{N}, \quad \Rightarrow \quad I = S(\Phi) \cong S_{abcd} \cdot \lim_{N \rightarrow \infty} \frac{n}{N}.$$

Очевидно, что аналогичным образом можно решить задачу об объеме некоего тела и т. п.

23.4 Построение гистограмм

Гистограммой называется график, аппроксимирующий по случайным данным плотность их распределения. При построении гистограммы область значений случайной величины $(a, b >$ разбивается на некоторое количество *bin* сегментов, а затем подсчитывается процент попадания данных в каждый сегмент. Для построения гистограмм в Mathcad имеется несколько встроенных функций. Рассмотрим их начиная с самой сложной по применению, чтобы лучше разобраться в возможностях каждой из функций.

Гистограмма с произвольными сегментами разбиения:

- $hist(intvis, x)$ – вектор частоты попадания данных в интервалы гистограммы;
- $intvis$ – вектор, элементы которого задают сегменты построения гистограммы в порядке возрастания $a < intvis_i < b$;
- x – вектор случайных данных.

Если вектор $intvis$ имеет bin элементов, то и результат $hist$ имеет столько же элементов. Построение гистограммы иллюстрируется листингом и рисунком 23.3.

Листинг. Построение гистограммы

```
N := 1000
bin := 30
x := rnorm(N, 0, 1)
lower := floor(min(x))
upper := ceil(max(x))
h :=  $\frac{\text{upper} - \text{lower}}{\text{bin}}$ 
```

```
j := 0..bin
intj := lower + h·j
f :=  $\frac{1}{N \cdot h} \cdot \text{hist}(\text{int}, x)$ 
```

Для анализа взято $N = 1000$ данных с нормальным законом распределения, созданных генератором случайных чисел (третья строка листинга). Далее определяются границы интервала (*upper*, *lower*), содержащего внутри себя все случайные значения, и осуществляется его разбиение на количество (*bin*) одинаковых сегментов, начальные точки которых записываются в вектор *int* (предпоследняя строка листинга). В векторе *int* можно задать произвольные границы сегментов разбиения так, чтобы они имели разную ширину.

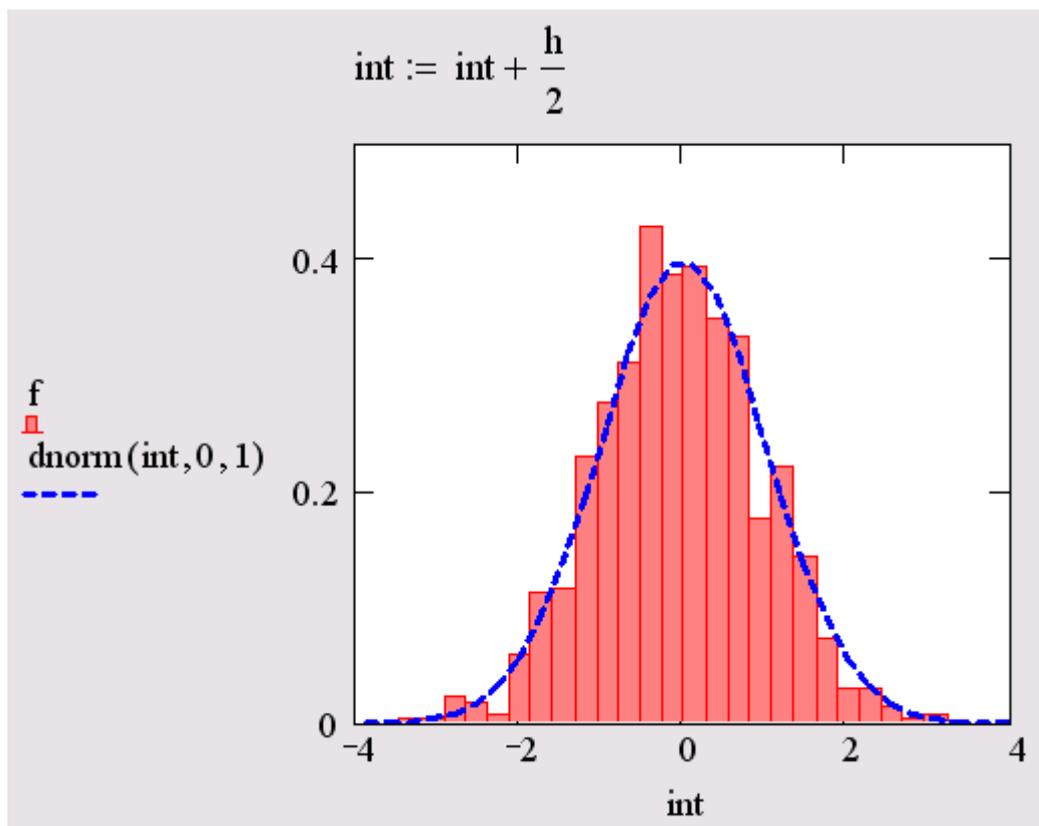


Рис. 23.3 – Построение гистограммы (листинг)

Контрольные вопросы

1. Как представляются статистические закономерности?
2. Назовите основные компоненты статистических характеристик.
3. Что представляет закон распределения?
4. Охарактеризуйте основные законы распределения случайных величин.
5. Для чего применяется ММК?
6. Что такое гистограмма и как она строится?

Литература

1. Райфа, Х. Прикладная теория статистических решений / Х. Райфа, Р. Шлейфер. – М. : Статистика, 1977. – 472 с.

24 Методы оптимизации в системах моделирования

24.1 Структурная схема оптимизации

Одной из важных задач исследования систем и технических объектов является *задача параметрического синтеза*, при решении которой по известным входным воздействиям и откликам цепи, а также структуре объекта производится поиск его параметров, оптимальных в рамках определенного критерия.

Блок-схема программы оптимизации модели представлена на рисунке 24.1.

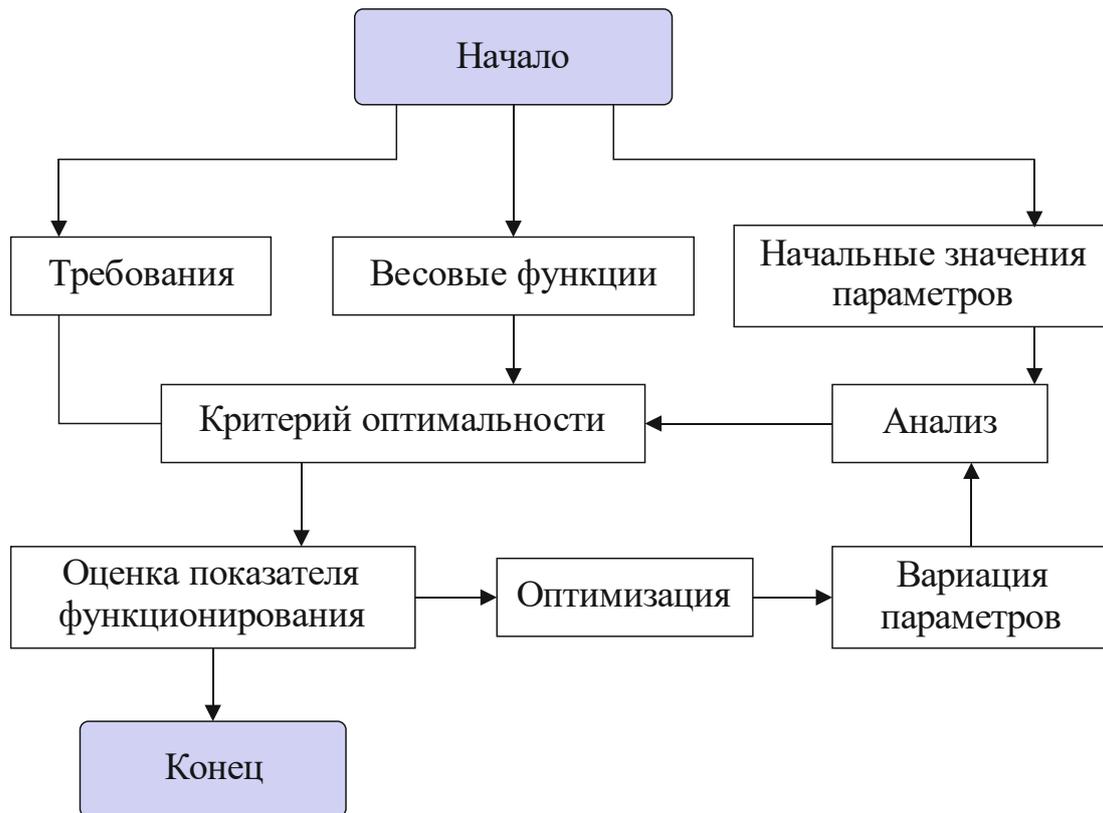


Рис. 24.1 – Блок-схема программы оптимизации модели

Здесь *исходные данные* представляют:

- 1) показатель качества функционирования модели;
- 2) весовые функции, отражающие основные свойства критерия оптимальности;
- 3) начальные значения параметров.

Характеристики, соответствующие начальным значениям параметров элементов, рассчитываются на ЭВМ и сравниваются с заданными посредством выбранного критерия ошибки. Если полученные характеристики удовлетворительны, выполнение программы заканчивается и информация поступает на выход.

При большой ошибке начальные значения параметров корректируются, и процесс повторяется до тех пор, пока характеристики схемы не окажутся в заданных пределах или число итераций не превысит заданное их число [1].

Рассмотрим формализованное представление исходных данных для проведения оптимизации на модели. В таблице 24.1 представлен порядок формирования ограничений на выходные параметры (характеристики) цепи.

Таблица 24.1 – Порядок формирования ограничений на выходные параметры

Выходной параметр	Значение по ТЗ	Нижняя граница	Верхняя граница	Направление изменения	Оптимизация по параметру
Параметр 1	Утз1	Утз1, min	Утз1, max	+	X_i

Параметр k	Утз k	Утз1, min	Утз k , max	–	X_j

В таблице 24.2 представлена процедура выбора управляемых при оптимизации параметров. В первый столбец «Управляемый параметр» автоматически заносятся параметры элементов схемы, управляемые разработчик помечает в таблице знаком.

Таблица 24.2 – Процедура выбора управляемых при оптимизации параметров

Управляемый параметр	Нижняя граница	Верхняя граница	Оптимизация по параметру
Параметр X_i	X_i, \min	X_i, \max	*

Параметр X_n	X_n, \min	X_n, \max	*

Допустимая область поиска обычно формируется из прямых

$$X \in [X_{\min}, X_{\max}]$$

и функциональных ограничений на управляемые параметры:

$$\varphi(x) > 0;$$

$$\varphi(x) < 0;$$

$$\varphi(x) = 0.$$

Здесь X_i, \min и X_i, \max – векторы, задающие нижнюю и верхнюю границы изменения параметров X_i , $\varphi(x)$ – функция управляемых параметров. Целевая функция (ЦФ) $F(Y(X))$ реализует заданный критерий оптимальности.

Она формируется на основе различных критериев оптимизации – аддитивного, мультипликативного и пр. Чаще других используется критерий формы

$$F(X) = \sqrt{\frac{\sum [Y - Y_{ж}]^{**2}}{n}}, \quad (24.1)$$

где $Y_{ж}$ – желаемый вид функции выходного параметра; n – количество узловых точек на характеристиках Y и $Y_{ж}$.

Для задания эталонной, желаемой формы ЦФ либо используется *эталонная модель*, либо она создается на основе накапливаемой истории процесса, которая фиксируется датчиками, измеряющими протекающие в реальном объекте процессы. В этом случае целевая функция формируется из *двух составляющих* (рис. 24.2):

- данных, поступающих с датчиков, которые образуют реальный канал;
- результатов компьютерного моделирования, образующих виртуальный (модельный) канал.

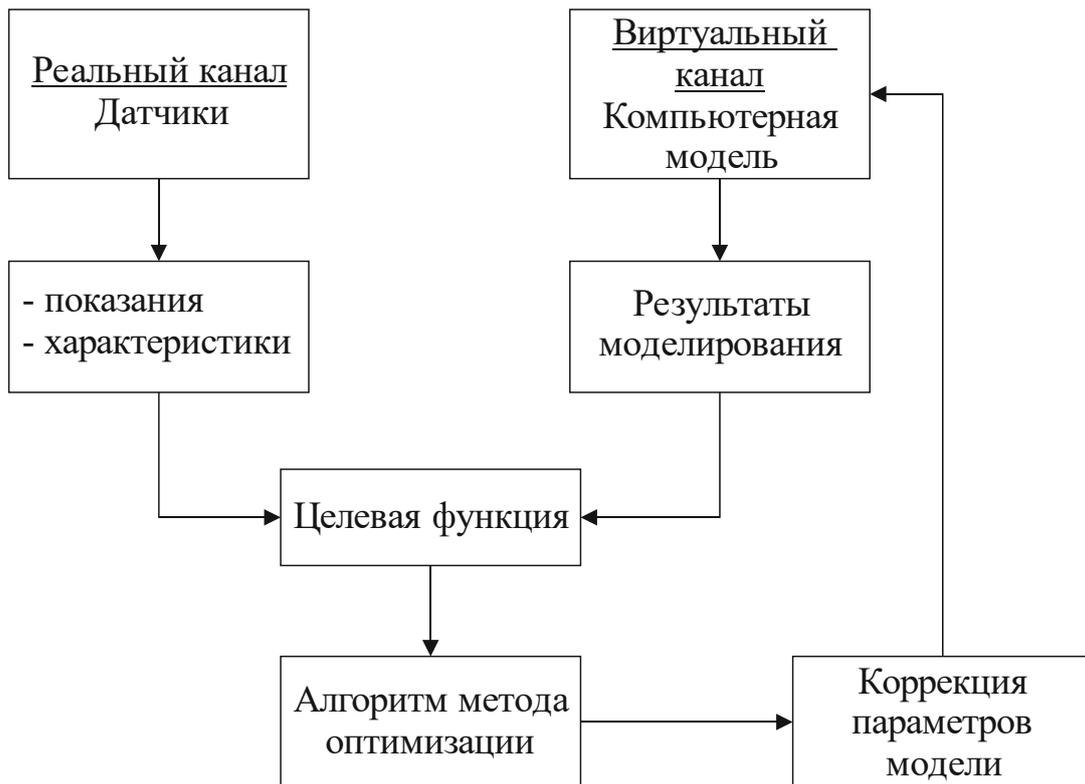


Рис. 24.2 – Схема коррекции параметров

Параметрическая оптимизация предполагает, что по известным откликам объекта на заданные воздействия при неизменной его структуре производится поиск значений его параметров, удовлетворяющих заданным критериям оптимизации.

Задача оптимизации формулируется как задача поиска минимума целевой функции $\min F Y(X, P)$ в допустимой области варьирования параметров модели P . Результатом оптимизации является вектор значений параметров P^* , при которых наблюдается минимальное значение ЦФ F . Эта функция реализует заданный пользователем *критерий оптимальности*. Она формируется на основании различных критериев оптимизации и вычисляется на основе полученных результатов анализа модели объекта, сформированной на объектном уровне многоуровневой компьютерной модели.

Компоненты для представления алгоритмов решения задач параметрической оптимизации

В процессе параметрической оптимизации характеристик системы лежит минимизация целевых функций, сформированных относительно измеряемых переменных модели и наблюдаемых характеристик реальной системы. Их решение основано на многократном анализе модели системы в статическом или динамическом режиме, результаты которого, включая параметры-функционалы полученных зависимостей, используются при формировании целевых функций. Для этого предусматриваются следующие *программно-алгоритмические возможности*:

- 1) использование стандартных компонентов-критериев, к которым относятся средневзвешенный аддитивный, среднеквадратический, мультипликативный критерии, а также критерий формы;
- 2) использование компонента «Целевая функция» (рис. 24.3), с помощью которого пользователем вводится целевая функция любого доступного математического вида относительно переменных его входных связей.

$$S_i = b_i \eta^- n_i \rightarrow V_i, \quad (24.2)$$

где $i = \overline{1, n}$.

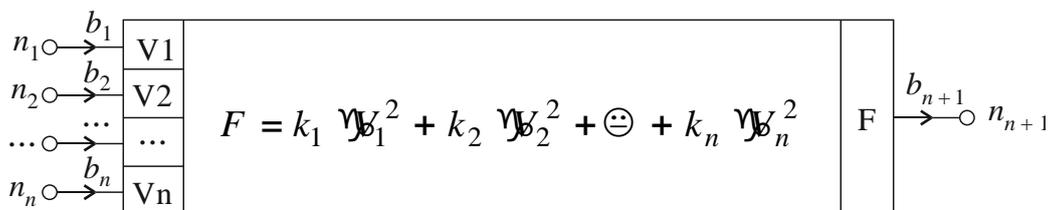


Рис. 24.3 – Компонент «Целевая функция»

Введенная в данный компонент, основанный на интерактивной математической панели, целевая функция автоматически преобразуется в компонентную

подцепь на основе системы отображения математических выражений. Полученная компонентная подцепь подлежит расчету с помощью универсального вычислительного ядра СКМ.

Полученные значения целевой функции передаются другим компонентам, реализующим алгоритмы методов оптимизации, с помощью выходной связи

$$S_{n+1} = b_{n+1} \eta^+ n_{n+1} \rightarrow \{F\}, \quad (24.3)$$

где F – вычисленное значение целевой функции.

Любой метод оптимизации, как безградиентный, так и градиентный, может быть реализован в виде специального компонента $[i]$, на входную связь которого

$$S_{n+1} = b_{n+1} \eta^- n_{n+1} \rightarrow \{F\} \quad (24.4)$$

поступает значение целевой функции, вычисленное с помощью компонента «Целевая функция» (рис. 24.4).

Реализующие градиентные методы оптимизации компоненты, помимо связи S_{n+1} (24.4), по которой в компонент поступает значение целевой функции, имеют входную связь:

$$S_{n+2} = b_{n+2} \eta^+ n_{n+2} \rightarrow \{G\}, \quad (24.5)$$

по которой принимают вычисленный вектор градиента целевой функции относительно переменных P_1, P_2, K, P_n , подлежащих варьированию в процессе работы реализованного в компоненте оптимизационного алгоритма. Значения параметров передаются другим компонентам по связям:

$$S_i = b_i \eta^+ n_i = P_i \quad (24.6)$$

для $i = \overline{1, n}$.

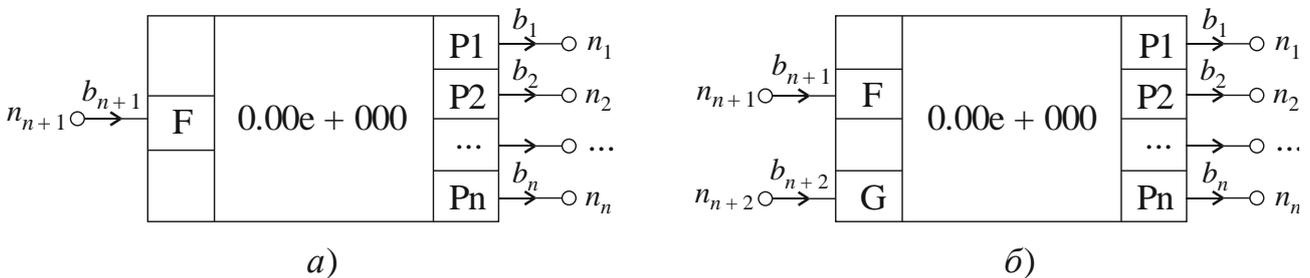


Рис. 24.4 – Компоненты методов оптимизации:
 а) безградиентного; б) градиентного

В случае применения градиентных методов для вычисления градиента целевой функции применяется компонент «Градиент» (рис. 24.5), связи которого предназначены:

$S_1 = b_1 \eta^- n_1 \rightarrow P_1, S_2 = b_2 \eta^- n_2 \rightarrow P_2, \dots, S_n = b_n \eta^- n_n \rightarrow P_n$, – для получения значений варьируемых параметров целевой функции, при которых необходимо найти значение градиента целевой функции;

$S_{n+1} = b_{n+1} \eta^- n_{n+1} \rightarrow \{F\}$ – для получения значения целевой функции при текущих значениях параметров;

$S_{n+2} = b_{n+2} \eta^- n_{n+2} \rightarrow \{G\}$ – для передачи рассчитанного значения градиента целевой функции.

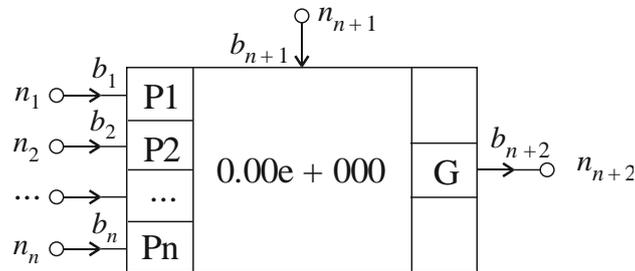


Рис. 24.5 – Компонент «Градиент»

После ввода описанного набора компонентов в формате алгоритмических КЦ появляется возможность формирования и исследования целевых функций и различных алгоритмов оптимизации. *Формирование алгоритмов параметрической оптимизации* осуществляется с помощью стандартных компонентов критериев, компонента «Целевая функция» (рис. 24.3), а также компонентов, реализующих методы условной (рис. 24.4, а) и безусловной оптимизации (рис. 24.4, б). В случае реализации градиентных методов расчет градиента целевой функции осуществляется с помощью компонента «Градиент» (рис. 24.5).

24.2 Программы оптимизации в СММ МАРС

В среде МАРС реализован ряд программ оптимизации, список которых представлен в таблице 24.3.

Таблица 24.3 – Программы оптимизации в СММ МАРС

Название метода	Параметры
<i>При известном начальном приближении</i>	
Покоординатный спуск	$\delta, \varepsilon, Nd, X_{нач}$

Градиентный метод наискорейшего спуска	$\delta, \varepsilon, Nd, X_{\text{нач}}, X_{\text{мин}}, X_{\text{макс}}$
Метод локально-глобального поиска	$\delta, \varepsilon, Nd, Nst, X_{\text{мин}}, X_{\text{макс}}$
Метод стохастического градиента	$\delta, \varepsilon, Nd, X_{\text{мин}}, X_{\text{макс}}$
<i>При неизвестном начальном приближении</i>	
Метод случайного поиска	$Nst, X_{\text{мин}}, X_{\text{макс}}$
Метод случайного поиска с обучением	$Nst, X_{\text{мин}}, X_{\text{макс}}$

Для решения поставленной задачи оптимизации характеристик исследуемого объекта был выбран *метод локально-глобального поиска*, который предназначен для отыскания глобального минимума сложно организованных целевых функций. В основе метода лежат специальные свойства ЦФ, заключающиеся в различном влиянии переменных на значение ЦФ и в образовании некоторыми переменными коррелированных подгрупп. Как показала практика, такие задачи часто возникают при оптимизации технических систем. *Предлагаемая методика заключается* в выделении групп наиболее влияющих переменных и оптимизации по ним, то есть изучении глобальных свойств оптимизируемой функции. Затем производится выделение связанных подгрупп и оптимизация по переменным, входящим в эти подгруппы, то есть изучение локальных свойств оптимизируемой функции. Отсюда и название метода. Метод локально-глобального поиска существенно снижает размерность поставленной задачи, что в свою очередь существенно ускоряет процесс поиска экстремума.

Формальная постановка задачи

Задана функция $F(X) = F(x_1, x_2, \dots, x_n)$; требуется найти $\min F(x_1, x_2, \dots, x_n)$ в области $\{\alpha_i < x_i < \beta_i\}$ $i = 1, 2, \dots, n$. Может быть задано для отыскания экстремума: $-(x_1^0, x_2^0, \dots, x_n^0)$ со значением целевой функции $F(X^0) = F(x_1^0, x_2^0, \dots, x_n^0)$; точность установления экстремума ξ .

Алгоритм локально-глобального поиска реализуется в четыре этапа.

Этап 1.1. Начальное приближение задано. Осуществляется переход к этапу 2.

Этап 1.2. Начальное приближение не задано. Каждая переменная выбирается как равномерно распределенная случайная величина в области $\{\alpha_i < x_i < \beta_i\}$. В зафиксированной случайной точке вычисляется значение целевой функции.

Этап 2. Вычисляются коэффициенты влияния каждой переменной на значение ЦФ. Выделяется R наиболее влияющих переменных. Решается задача оптимизации по R переменным, выбранным из таблицы 24.2 методом оптимизации. Оптимизация в выделенной на этапе 2 группе соответствует глобальному поиску.

Этап 3. Выделяются локальные подгруппы и в них производится оптимизация. Выделение подгрупп производится путем вычисления приращений вида $\delta_i = F(x_1, x_2, \dots, x_i + \Delta(\beta_i - \alpha_i), \dots, x_n)$ и их оценки, на основе которых и выделяются локальные группы из R выбранных переменных. Производится оптимизация по R выделенным переменным. Восстанавливается исходное значение i .

Этап 4. Если при оптимизации на этапах 2 и 3 не произошли изменения оптимальной точки, то работа алгоритма считается законченной. В противном случае осуществляется переход к этапу 2. На этапе 4 оптимум считается не изменившимся, если разность старого и нового значения минимума не превышает заданной точности ε .

24.3 Пример задачи параметрического синтеза

По известным входным воздействиям и откликам цепи, а также структуре объекта производится поиск его параметров, оптимальных в рамках определенного критерия.

Для решения данной задачи в рамках СММ МАРС предложена модель компонента «Оптимизатор» (рис. 24.6), в состав которого входит редактор математических выражений. Узлы N_1, N_2, \dots, N_n данного компонента соответствуют параметрам целевой функции, выраженным параметрами моделей компонентов КЦ исследуемого объекта.

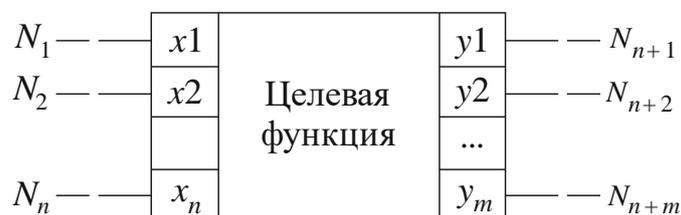


Рис. 24.6 – Компонент «Оптимизатор» с целевой функцией

Узлы $N_{n+1}, N_{n+2}, \dots, N_{n+m}$ соответствуют переменным целевой функции, которые являются одновременно и переменными исследуемой КЦ. В качестве параметров компонента целевой функции пользователь может задавать:

- 1) алгоритм (метод) оптимизации;
- 2) для параметров целевой функции – имя x_i , с которым этот параметр входит в состав целевой функции, его начальное значение X_i (для методов с ограничениями – интервалы изменения переменной) и начальный шаг изменения значения;
- 3) для переменных целевой функции – имена, которыми они представлены в целевой функции.

Целевая функция формируется пользователем в редакторе математических выражений относительно имен переменных целевой функции и (или) имен ее параметров. При закрытии редактора введенное в качестве целевой функции математическое выражение преобразуется в компонентную подцепь на основе системы отображения математических выражений в формат КЦ и встраивается в основную компонентную цепь, набранную в схемном редакторе среды моделирования МАРС. При запуске исследуемой КЦ на моделирование управление передается данному компоненту, в котором реализуется выбранный пользователем алгоритм оптимизации. Для расчета значения целевой функции при текущих значениях параметров целевой функции данный компонент запускает вычислительное ядро для выбранного режима анализа. При этом сам выступает источником значений параметров целевой функции, формируя на этапе вычислительного опроса уравнения вида:

$$V_{Ni} = X_i$$

для всех $i = 1 \dots n$, где X_i – текущее значение аргумента с именем x_i .

Также на данном этапе формируются уравнения всех компонентов математической подцепи, которые добавляются в основную систему уравнений. На этапе измерительного опроса цепи данным компонентом из вектора решения выбирается текущее значение целевой функции, по которому принимается решение о движении алгоритма. По окончании работы данного алгоритма оптимизации узлы, соответствующие аргументам и переменным целевой функции, будут иметь значения, при котором целевая функция достигает своего минимума. Данный компонент позволяет производить как одномерную, так и многомерную оптимизацию технических объектов.

Примеры оптимизации модельных конструкций блоков систем

В качестве тестового примера задачи параметрической оптимизации продемонстрируем поиск параметра сопротивления нагрузки в делителе напряжения, при котором выполняется условие выделения максимальной мощности на

ней. Для ее решения в рамках среды MAPS построена следующая компонентная гидравлическая цепь (рис. 24.7), представленная своим электрическим аналогом.

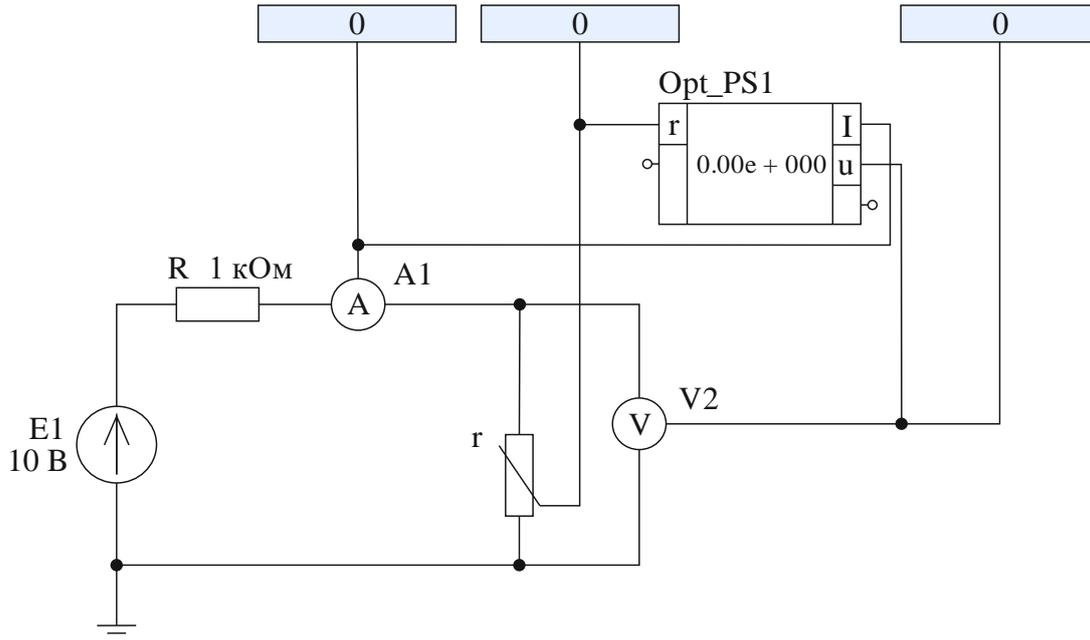


Рис. 24.7 – Компонентная цепь для решения задачи параметрической оптимизации

В качестве параметра целевой функции выступает гидросопротивление нагрузки r , переменными целевой функции являются поток в цепи i и давление на нагрузке u . Для получения максимальной мощности на нагрузке r в редакторе математических выражений формируется целевая функция вида (рис. 24.8).

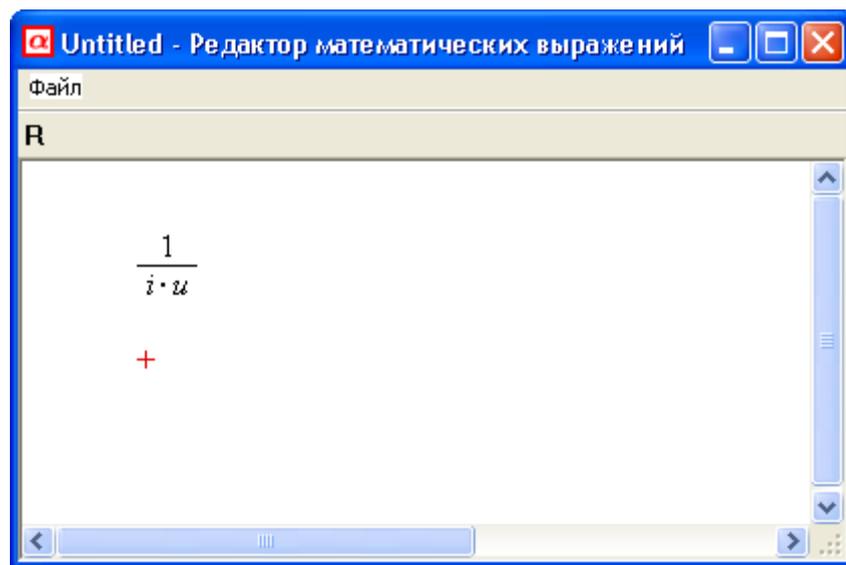


Рис. 24.8 – Целевая функция в редакторе математических выражений

Начальное значение параметра r равно 10 Ом, максимальный шаг изменения параметра – 5 Ом. Результаты проведенной оптимизации методом покоординатного спуска приведены на рисунке 24.9.

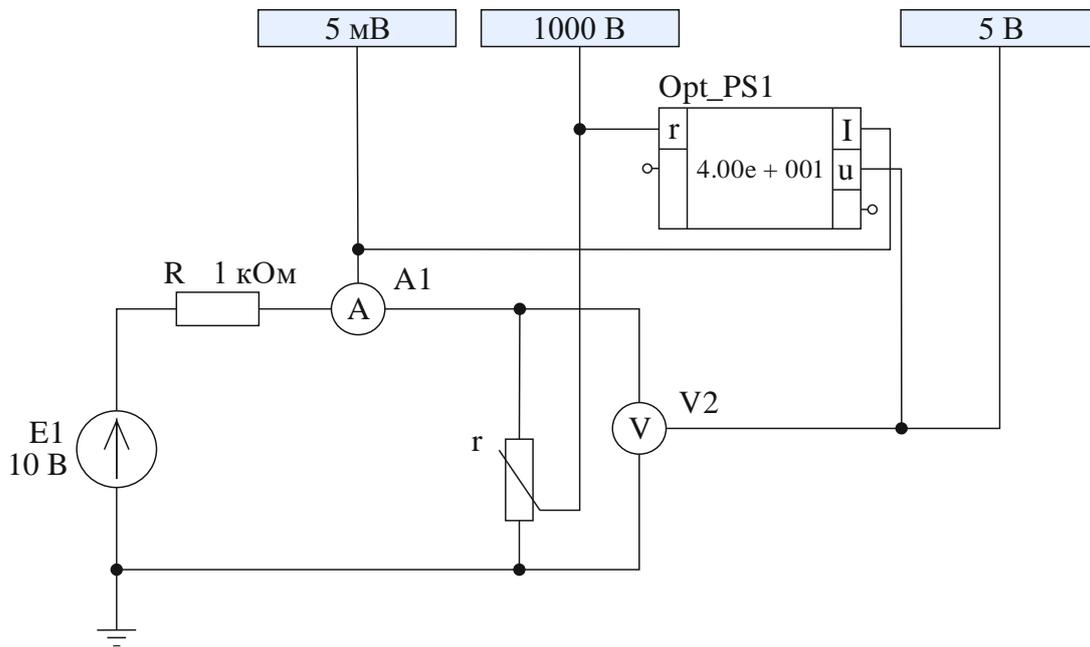


Рис. 24.9 – Результаты оптимизации в среде моделирования MARC

Представленный в данной работе компонент «Оптимизатор» позволяет производить параметрический синтез технических объектов в рамках среды моделирования MARC [2]. На его основе могут быть реализованы различные методы оптимизации, как без ограничений, так и с ограничениями.

Контрольные вопросы

1. Как определяется задача параметрического синтеза?
2. Что собой представляют исходные данные на оптимизацию?
3. Что включает порядок формирования ограничений на выходные параметры объекта?
4. Назовите основные критерии оптимизации.
5. Дайте формальную постановку задачи локально-глобального поиска.

Литература

1. Дмитриев, В. М. Интерактивная математическая панель для моделирования систем с информационными и энергетическими связями / В. М. Дмитриев, Т. В. Ганджа // Электронные средства и системы управления : материалы Междунар. науч.-практ. конф. – Томск : Изд-во Ин-та оптики атмосферы СО РАН, 2004. – Ч. 2. – С. 62–65.

2. Ганджа, Т. В. Применение макрокомпонентов для решения задач функционального моделирования и проектирования // Моделирование. Теория, методы и средства : материалы X Междунар. науч.-практ. конф. (г. Новочеркасск, 5 апреля 2010 г.) / Юж.-Рос. гос. техн. ун-т (НПИ). – Новочеркасск : ЮРГТУ, 2010. – С. 4–8.

25 Измерения и обработка результатов моделирования

25.1 Компоненты-блоки для обработки результатов моделирования

Результаты моделирования в динамическом режиме, который является основным для анализа технических объектов с инерционными компонентами, могут быть выведены на компонент-график в виде временных либо частотных характеристик. В этом случае они описываются временными (стационарными или переходными) или частотными зависимостями. С помощью интерфейсных возможностей (визирных линий, курсоров и лупы) двухмерного графика могут быть найдены различные числовые характеристики зависимостей, называемые *параметрами-функционалами*. Для определения параметров-функционалов описываемых зависимостей в среде моделирования MAPS реализована *библиотека моделей компонентов* для обработки результатов моделирования, *компоненты которой позволяют анализировать*:

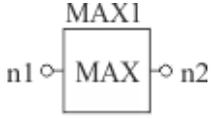
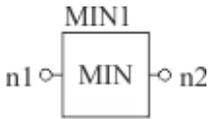
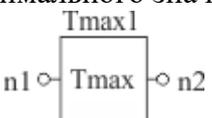
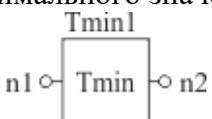
- общие временные и частотные характеристики;
- стационарные колебательные процессы;
- переходные колебательные процессы.

Общие компоненты-блоки обработки результатов моделирования

Общими для любых временных и частотных характеристик являются максимальное и минимальное значение на всем промежутке моделирования, локальные минимумы и максимумы, а также значение времени или частоты, при которых они наблюдаются.

Определение максимального (минимального) значения измеряемой величины (тока, напряжения, сопротивления, мощности или разности потенциалов между током и напряжением), а также времени (частоты) их наступления на всем промежутке моделирования может быть полезным при *моделировании переходных процессов* во временной области, а также при исследовании амплитудно- или фазочастотной характеристик в частотной области (табл. 25.1) [1].

Таблица 25.1 – Компоненты-блоки определения глобальных экстремумов обрабатываемых сигналов

Название, графическое представление	Математическая модель и параметры
<p>Максимальное значение</p> 	$V_{n2} = \text{MAX}(V_{n1})$, где V_{n1} – массив результатов анализа во временной, частотной областях или многовариантного анализа; MAX – функция определения максимального значения
<p>Минимальное значение</p> 	$V_{n2} = \text{MIN}(V_{n1})$, где V_{n1} – массив результатов анализа во временной, частотной областях или многовариантного анализа; MIN – функция определения минимального значения
<p>Абсцисса (время наступления) максимального значения</p> 	$V_{n2} = \text{TMAX}(V_{n1})$, где V_{n1} – массив результатов анализа во временной, частотной областях или многовариантного анализа; TMAX – функция определения абсциссы (времени наступления) максимального значения
<p>Абсцисса (время наступления) минимального значения</p> 	$V_{n2} = \text{TMIN}(V_{n1})$, где V_{n1} – массив результатов анализа во временной, частотной областях или многовариантного анализа; TMIN – функция определения абсциссы (времени наступления) минимального значения

На рисунке 25.1, а приведена схема электрической цепи, на которую подается δ -импульс с целью построения и исследования импульсной переходной характеристики, изображенной на рисунке 25.1, б. δ -импульс подается в цепь с помощью источника трапецеидального воздействия со специально настроенными временными параметрами. На рисунке 25.1, в приведен результат исследования данной характеристики, где определяется максимальное и минимальное значение, а также время их наступления.

Большинство блоков обработки результатов, реализованные в последующих алгоритмах, требуют определения набора локальных экстремумов (максимумов и (или) минимумов) совместно со значением абсциссы (времени), при котором они наблюдаются. Для этих целей реализованы компоненты (табл. 25.2), которые при определении каждого локального экстремума передают вектор M , содержащий значение абсциссы (времени) x и ординаты y точки экстремума, компонентам, подключенным к узлу n_2 .

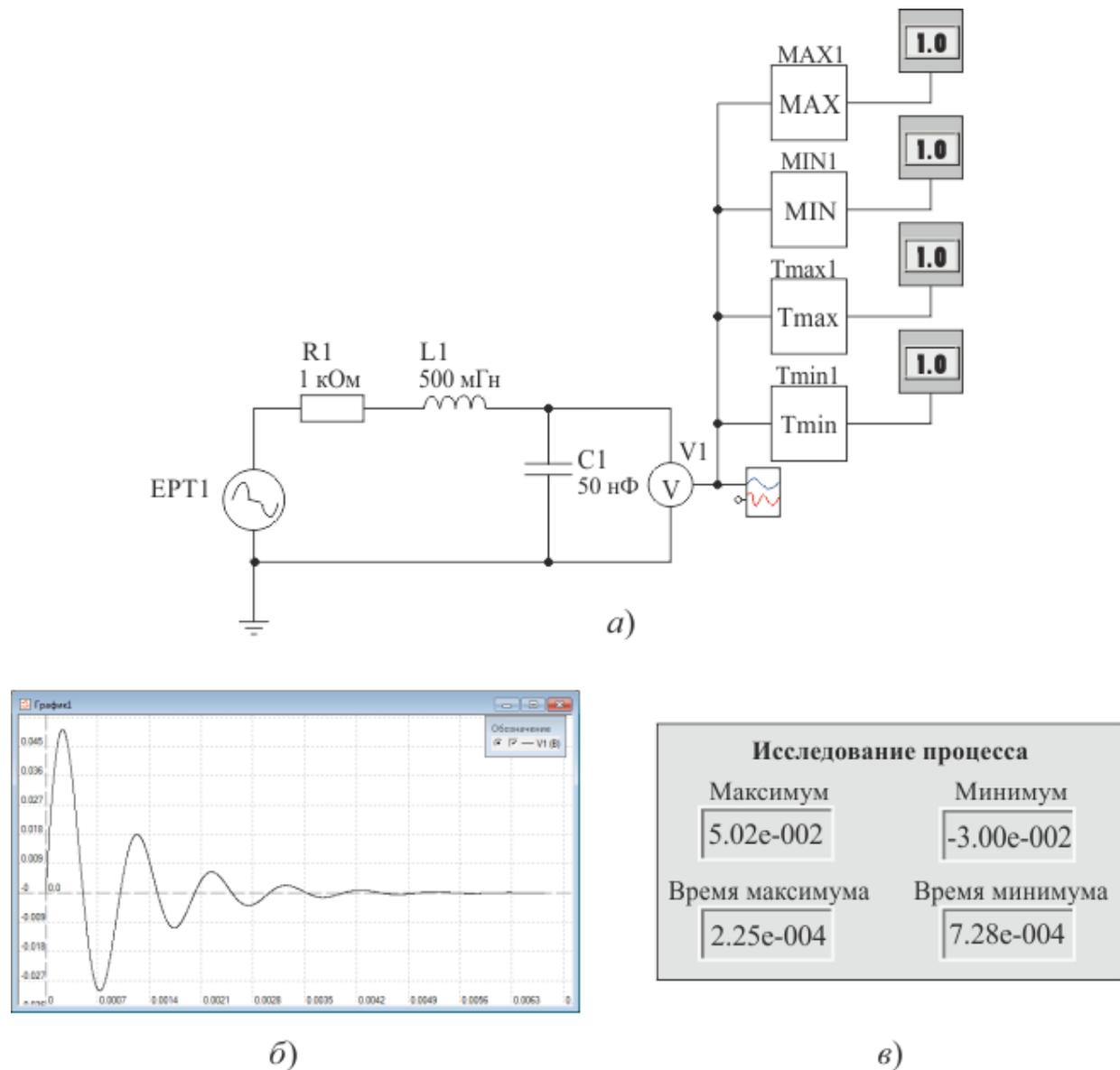


Рис. 25.1 – Исследование максимальных и минимальных значений процессов

Таблица 25.2 – Компоненты-блоки определения локальных экстремумов в массиве результатов моделирования

Название, графическое представление	Математическая модель и параметры
Локальный максимум 	$V_{n2} = LMAX(V_{n1})$, где V_{n1} – массив результатов анализа во временной, частотной областях или многовариантного анализа; LMAX – функция определения локальных максимумов и абсцисс (времени) их наступления
Локальный минимум 	$V_{n2} = LMIN(V_{n1})$, где V_{n1} – массив результатов анализа во временной, частотной областях или многовариантного анализа; LMIN – функция определения локальных минимумов и абсцисс (времени) их наступления

25.2 Блоки обработки результатов стационарных колебательных процессов

Для анализа стационарных колебательных процессов в среде моделирования МАРС применяется быстрое *преобразование Фурье*, на основе которого при разложении сигнала с минимальной фундаментальной частотой можно определять:

- амплитуду,
- частоту,
- фазу

заданной гармонике.

При анализе полученного спектрального разложения сигнала можно определить следующие *параметры-функционалы сигнала*:

- среднее значение сигнала;
- действующее значение сигнала;
- период сигнала;
- частоту сигнала.

Компонент «Преобразование Фурье» (рис. 25.2) позволяет получить спектральное разложение стационарных колебательных процессов, являющихся результатами моделирования в среде МАРС и поступающих на вход n_1 с информационных выводов измерительных компонентов.

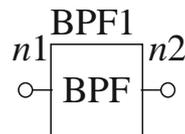


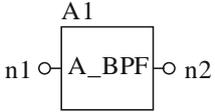
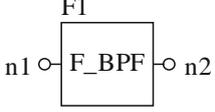
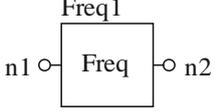
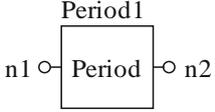
Рис. 25.2 – Компонент «Преобразование Фурье»

В качестве параметров данного компонента пользователем задается значение фундаментальной частоты, т. е. частоты минимальной гармонике, относительно которой будет проводиться разложение сигнала в ряд Фурье, а также количество первых гармоник. Результатом работы данного компонента является набор F , упорядоченных по частотам амплитуд $A(F(i))$ и фаз $\varphi(F(i))$ гармоник сигнала, поступающего на вход N_1 (где: $i = 1, 2, \dots, n_{\max}$ – номер гармонике, n_{\max} – максимальный номер гармонике).

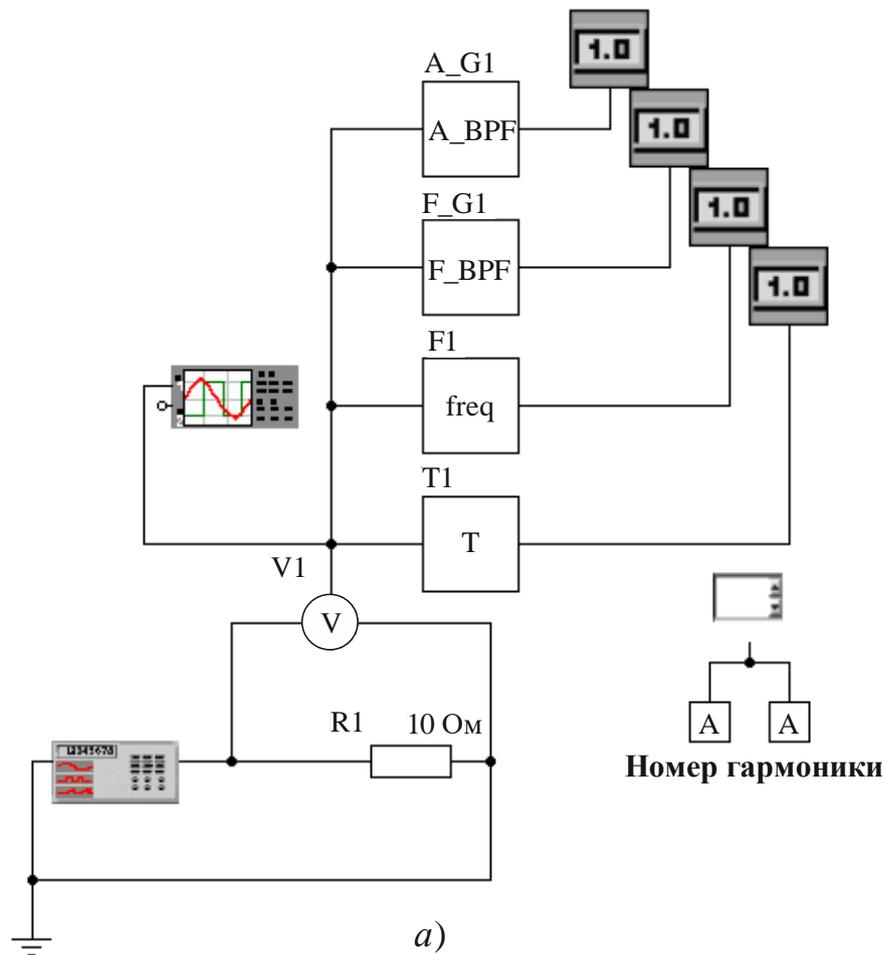
На основе преобразования Фурье реализуются компоненты для анализа стационарных периодических колебаний, поступающих в качестве результатов

моделирования на вход n_1 . *Результатами работы компонентов является числовое значение искомого параметра-функционала, которое передается на узел n_2 для дальнейшей обработки и визуализации с помощью специализированных средств (табл. 25.3).*

Таблица 25.3 – Компоненты-блоки обработки результатов стационарных переходных процессов

Название, графическое представление	Математическая модель и параметры
Амплитуда гармоники A 	$V_{n2} = A(F(n)),$ где n – заданный номер гармоники
Фаза гармоники F 	$V_{n2} = \varphi(F(n)),$ где n – заданный номер гармоники
Частота сигнала 	$V_{n2} = \omega(F(m)),$ где m – номер гармоники с максимальным значением амплитуды
Период сигнала 	$V_{n2} = T = (2 \cdot \pi) / \omega(F(m)),$ где m – номер гармоники с максимальным значением амплитуды

Работа описанных компонентов при подаче в электрическую цепь (рис. 25.3, *а*) сигнала амплитудой 10 В и частотой 100 Гц продемонстрирована на рисунке 25.3, *б*.



б)

Рис. 25.3 – Анализ стационарных периодических колебаний с помощью преобразования Фурье

Спектроанализатор позволяет разложить проходящий через него гармонический сигнал в ряд Фурье (рис. 25.4).

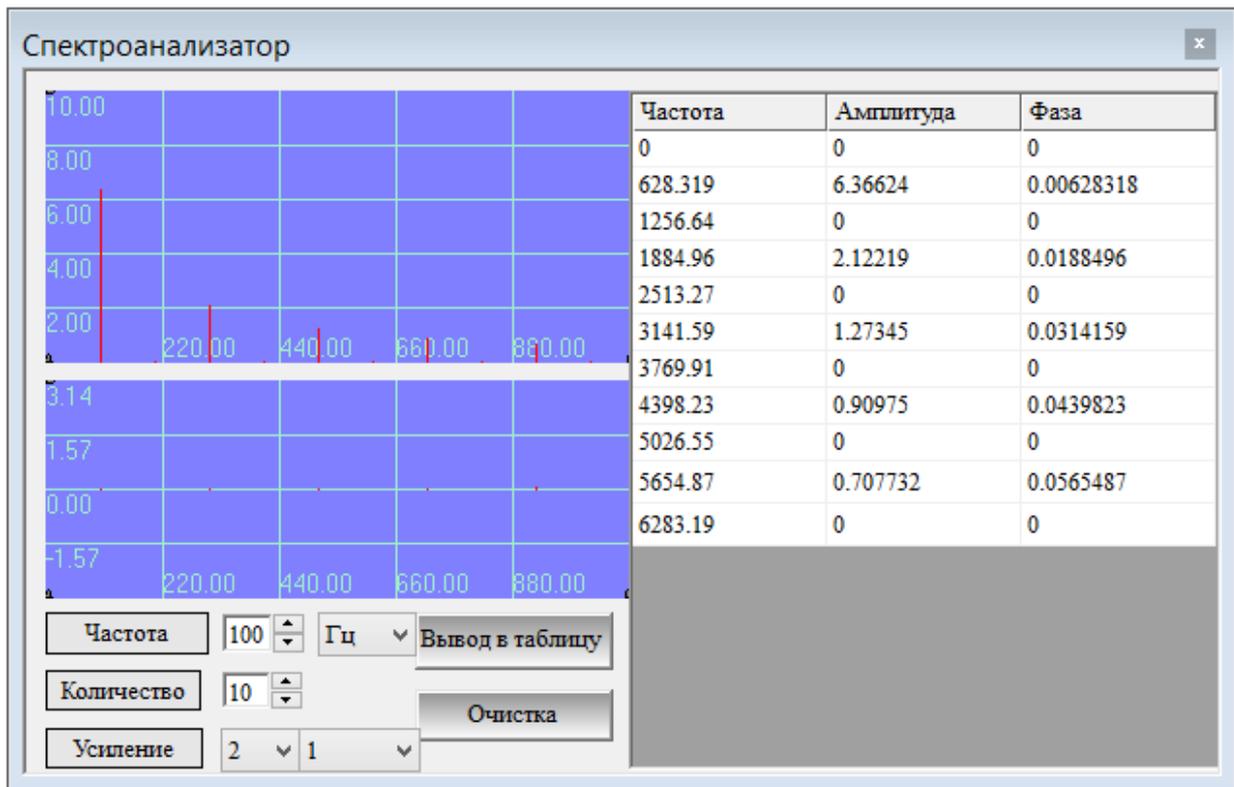


Рис. 25.4 – Лицевая панель спектроанализатора

Фундаментальная частота, относительно которой прибор производит разложение измеряемого сигнала в ряд Фурье, задается органами управления

Частота Гц

Число первых рассчитываемых гармоник задается с помощью органа управления

Количество

Усиление сигнала, соответствующее размеру одной клетки по оси ординат, задается с помощью органа управления

Усиление

Для получения результатов спектрального анализа в табличном виде необходимо нажать на кнопку . Кнопкой производится очистка таблицы. Схема включения спектроанализатора в цепь приведена на рисунке 25.5.

Для поиска разности фаз между двумя синусоидальными сигналами одинаковой частоты, а также для определения начальной фазы синусоидального сигнала используется компонент «Разность фаз» (рис. 25.6). На вход n_1 данного компонента подается сигнал, фазовый сдвиг которого неизвестен и подлежит определению, на вход n_2 подается опорный гармонический сигнал, относи-

тельно которого определяется фазовый сдвиг. С выхода n_3 компонента «Разность фаз» снимается разность фаз в градусах или радианах и передается на компоненты визуализации числовых значений или на дальнейшие блоки математической обработки.

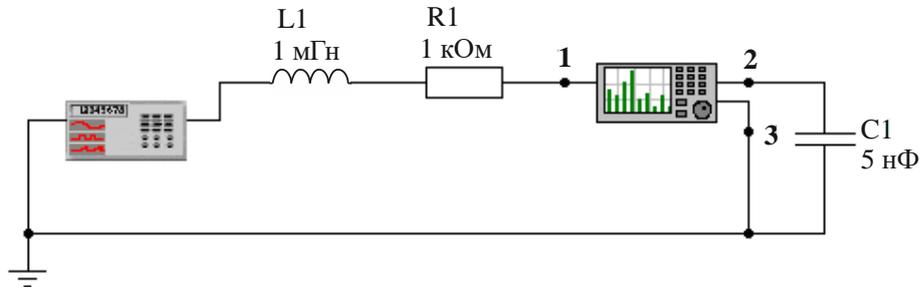
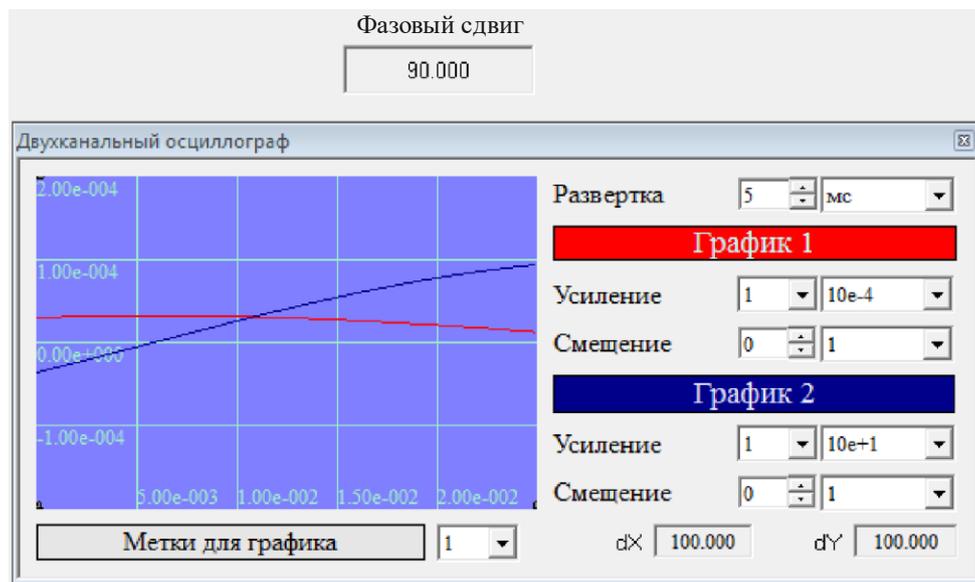
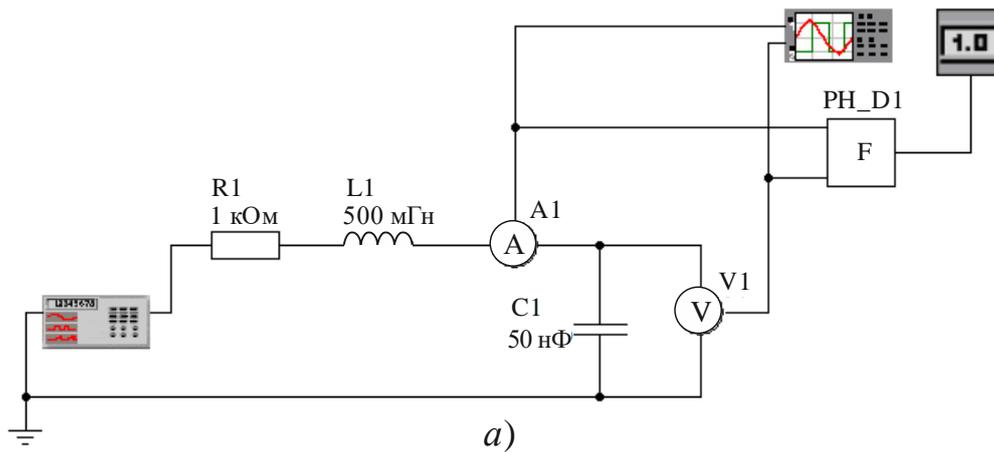


Рис. 25.5 – Схема включения спектроанализатора в цепь

Схема, формируемая для определения разности фаз между двумя сигналами, приведена на рисунке 25.6, а, а результат работы данного компонента приведен на рисунке 25.6, б.

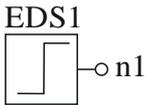
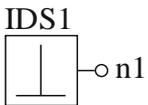
Анализ переходных процессов. Переходные процессы присущи физически неоднородным техническим объектам с инерционными компонентами, математические модели которых описываются обыкновенными дифференциальными уравнениями. Анализ переходных процессов в моделях таких объектах обычно проводится при постоянных, импульсных или гармонических воздействиях. Для генерации таких сигналов в среде моделирования МАРС реализованы источники, приведенные в таблице 25.4.



б)

Рис. 25.6 – Схема для определения фазового сдвига и результаты его определения

Таблица 25.4 – Источники для исследования переходных процессов

Название, графическое представление	Математическая модель	Назначение
Источник единичного импульса EDS1 	$V_{n1} = \begin{cases} 1 & \text{при } t \geq 0; \\ 0 & \text{при } t < 0 \end{cases}$	Построение переходной характеристики объекта
Источник δ -импульса IDS1 	$V_{n1} = \begin{cases} 1 & \text{при } t = 0; \\ 0 & \text{при } t \neq 0, \end{cases}$	Построение импульсной характеристики объекта

Для анализа переходных процессов, вызванных действием описанных источников, а также любых других источников, включенных в библиотеку моделей

компонентов СММ МАРС, применяются *компоненты-блоки обработки результатов анализа переходных процессов*:

- МАХ перерегулирования;
- число колебаний;
- время переходного процесса.

Они реализуются с помощью компонентов обработки статистических параметров (табл. 25.5).

Таблица 25.5 – Компоненты для обработки генеральных совокупностей

Компонент	Графический вид компонента	Математическая модель
Арифметическое среднее	N1 <input type="text" value="mean"/> N2	$V_{N2} = \frac{1}{S} \cdot \sum_{j=0}^{S-1} V_{N1, j}$
Медиана	N1 <input type="text" value="median"/> N2	
Дисперсия	N1 <input type="text" value="var"/> N2	$V_{N2} = M \left((V_{N1} - M(V_{N1}))^2 \right)$
Среднеквадратичное отклонение	N1 <input type="text" value="stdev"/> N2	$V_{N2} = \sqrt{\frac{S}{S-1} D(V_{N1})^2}$
Максимальное значение	N1 <input type="text" value="max"/> N2	$V_{N2} = \max(V_{N1})$
Минимальное значение	N1 <input type="text" value="min"/> N2	$V_{N2} = \min(V_{N1})$
Мода	N1 <input type="text" value="mode"/> N2	
Количество элементов	N1 <input type="text" value="count"/> N2	$V_{N2} = S$
Ковариация	N1 <input type="text" value="cvar"/> N2	$V_{N2} = \sum (V_{N1} - M(V_{N1}))(V_{N3} - M(V_{N3}))$
Корреляция	N1 <input type="text" value="corr"/> N3	$V_{N2} = \frac{c \operatorname{var}(V_{N1}, V_{N3})}{D(V_{N1})D(V_{N2})}$

Контрольные вопросы

1. В каком виде временных либо частотных характеристик выводятся результаты измерений?

2. Чем стационарные колебательные процессы отличаются от переходных?
3. Назовите измеряемые параметры переходных процессов.
4. Какие типы источников служат для исследования переходных процессов?
5. Какие характеристики получаются, если разложить проходящий через объект гармонический сигнал в ряд Фурье?
6. Назовите компоненты обработки статистических параметров.

Литература

1. Функциональный контроль и диагностика электротехнических и электромеханических систем и устройств по цифровым отсчетам мгновенных значений токов и напряжений // В. С. Аврамчук, Н. Л. Бацева, Е. И. Гольдштейн, И. Н. Исаченко, Д. В. Ли, А. О. Сулайманов, И. В. Цапко // под ред. Е. И. Гольдштейна. – Томск : Печатная мануфактура, 2003. – 240 с.

26 Компьютерные модели в системах поддержки принятия решений

26.1 Системы поддержки принятия решений

Система поддержки принятия решений (СППР) представляет собой вид компьютерных информационных систем, помогающих управляющему в принятии решений (ПР), при решении плохо структурированных задач посредством прямого диалога с ЭВМ и с использованием данных, знаний и математических моделей.

Отличительные особенности СППР:

- ориентация на решение плохо структурированных задач, характерных для высоких уровней управления;
- место СППР среди систем – это возможность сочетания традиционных методов доступа и обработки компьютерных данных с возможностями компьютерных моделей и методами решения на их основе;
- направленность на непрофессионального конечного пользователя посредством диалога с ЭВМ;
- высокая адаптивность путем приспособления к особенностям технического и программного обеспечения пользователя.

Информационную системную модель некоторой организации можно представить в виде следующей иерархической модели, включающей в себя 3 уровня: Обработки данных → Обработки информации → Принятие решений (рис. 26.1).

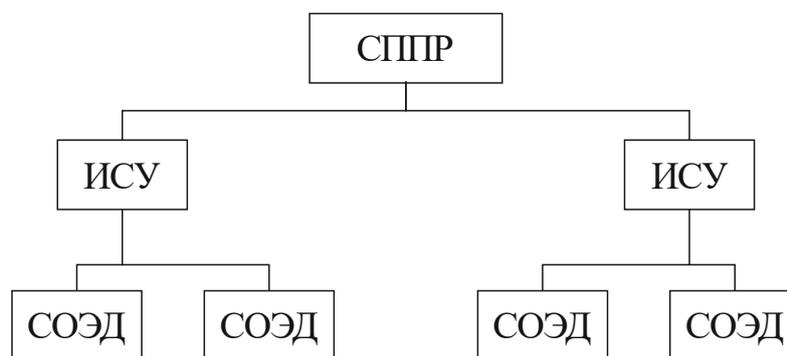


Рис. 26.1 – Иерархия информационных систем в компании

На первом уровне находятся средства обработки экспериментальных данных (СОЭД). Их основными характеристиками являются:

- обработка данных на уровне оперативного контроля;

- обработка коммерческих операций;
- составление расписаний и протоколов;
- составление отчетов для руководства.

На втором уровне (обработки информации), соответствующем уровню управленческого контроля, акцент переносится на процедуры обработки информации, выполняемые информационной системой управления (ИСУ). Ее основными характеристиками следует считать:

- подготовку информации, полезной на уровне среднего руководства;
- структурирование информационных потоков;
- интеграцию данных (ИСУ маркетинга, ИСУ производства и т. п.);
- создание запросно-ответной системы и подготовка отчетов для руководства.

На третьем, самом высоком этапе управления, соответствующим стратегическому планированию, формируются наиболее важные решения организации и СППР имеет следующие характеристики:

- подготовка вариантов решений для высшего руководства;
- оперативные ответы на запросы пользователей;
- обеспечение помощи в принятии решений любым индивидуальным управленцам [1].

26.2 Структура СППР

В состав СППР помимо пользователя входят 3 главных компонента: подсистема обработки и хранения данных; подсистема хранения и использования моделей и программная подсистема (рис. 26.2).

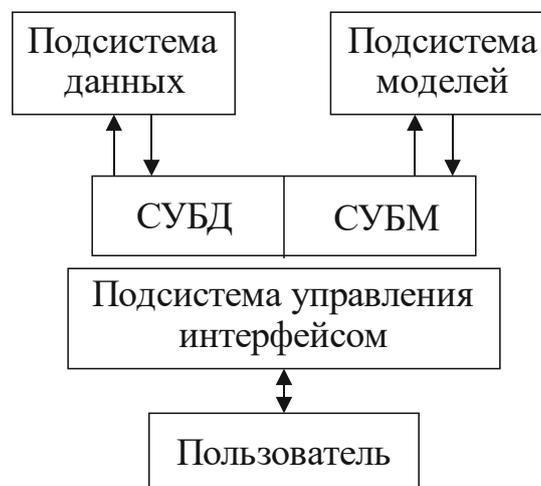


Рис. 26.2 – Структура подсистемы данных СППР

Использование баз данных (БД) в составе СППР характеризуется определенными особенностями. Так, например, БД в составе СППР имеют значительно больший набор источников данных, включая внешние источники, важные для принятия решений на высоких уровнях управления.

Подсистема данных, входящих в состав СППР, должна обладать следующими возможностями:

- составление комбинаций данных путем использования процедур агрегирования и фильтрации;
- оперативное добавление или исключение данных;
- построение структуры данных в терминах пользователя;
- управление данными с помощью функций СУБД;
- обеспечение полной логической независимости БД, входящей в СППР от других операционных БД.

26.3 Подсистема моделей

СППР обеспечивают доступ не только к данным, но и к моделям ПР. Это достигается введением в ИСУ соответствующих моделей и использованием в ней БД как механизма интеграции моделей и коммуникации между ними (рис. 26.3). СУБМ – система управления базами модели.

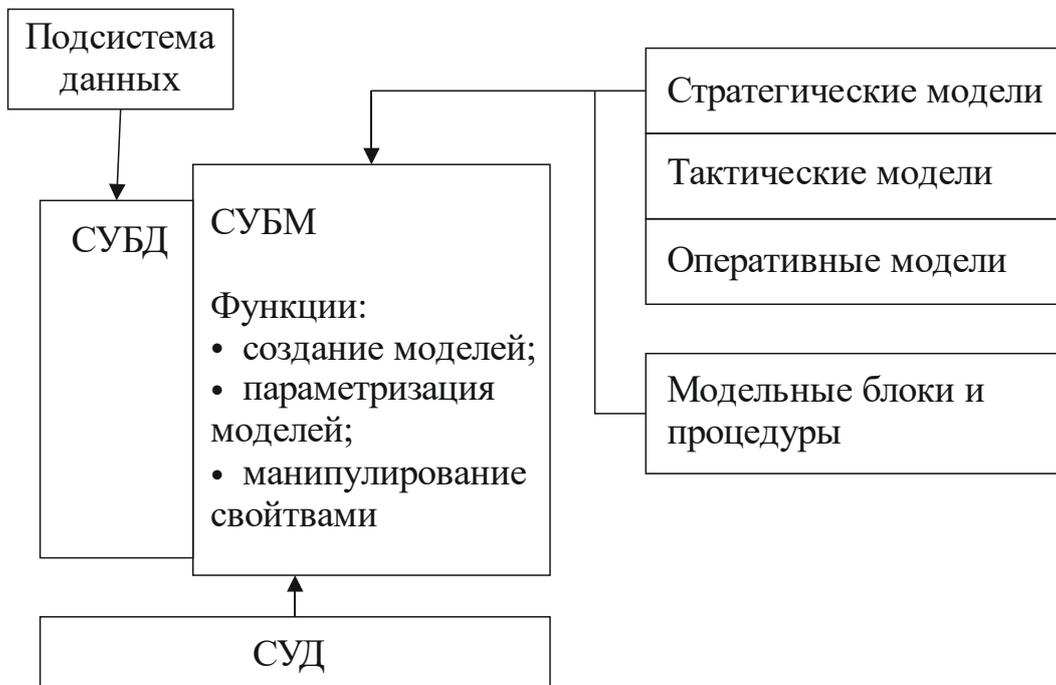


Рис. 26.3 – Структура подсистемы моделей СППР

Использование моделей обеспечивает способность СППР к проведению анализа. Модели, используя математическую интерпретацию проблемы, при помощи определенных алгоритмов способствуют нахождению информации, полезной для принятия решений. *Целью создания моделей* являются либо оптимизация, либо описание некоторого объекта или процесса.

Оптимизационные модели связаны с нахождением точек минимума или максимума некоторых показателей.

Описательные модели описывают поведение некоторой системы и не предназначены для целей управления. Они образуют базу моделей, включающую в себя *стратегические, тактические и оперативные модели*, а также совокупность модельных блоков и модулей для построения моделей.

Стратегические модели используются на высших уровнях управления для установления целей организации, объемов ресурсов, необходимых для их достижения. Эти модели обычно детерминированные, описательные и специализированные для использования на одной определенной фирме.

Тактические модели применяются управляющими среднего уровня для распределения и контроля имеющихся ресурсов. Среди возможных сфер их использования: финансовое планирование, планирование увеличения продаж, построение схем подготовки предприятий. Обычно тактические модели реализуются как *детерминированные, оптимизационные и универсальные*.

Оперативные модели используются на низших уровнях управления для поддержки принятия оперативных решений с горизонтом на дни или недели. Возможные сферы их применения включают в себя календарное производственное планирование, управление запасами и т. д. Обычно оперативные модели *детерминированные, оптимизационные и универсальные*.

В дополнение к этим моделям *база моделей СППР* включает в себя совокупность модельных блоков и процедур. Сюда могут входить процедуры линейного программирования, статистического и регрессионного анализа и т. п.

Рассмотрим *основные шаги при компьютерном моделировании решаемых проблем*.

Первый шаг – выявление факторов или переменных, которые лицо, принимающее решения (ЛПР), считает важными. Они могут быть классифицированы в виде следующих *пяти категорий*:

1. *Управляемые переменные* – это наиболее важные переменные, формирующие альтернативы решения, при условии, что они контролируются со стороны ЛПР.

2. *Внешние, или экзогенные, переменные*, от которых зависит решение рассматриваемой проблемы, но их выбор находится за пределами компетенций ЛПР.
3. *Выбранная политика с ограничениями*. ЛПР часто вынужден действовать в условиях законодательных или физических ограничений.
4. *Выбранный критерий*. Принимая решения, менеджер преследуют определенные цели, которых он хотел бы достичь. Количественным отображением выбранной цели и является критерий.
5. *Промежуточные переменные*. Для того чтобы модель была достаточно полной и удобной для решения и анализа, в нее часто включают дополнительные или промежуточные переменные, например при расчетах себестоимости или нормы прибыли.

На рисунке 26.4 показано, как связаны различные категории переменных в модели.

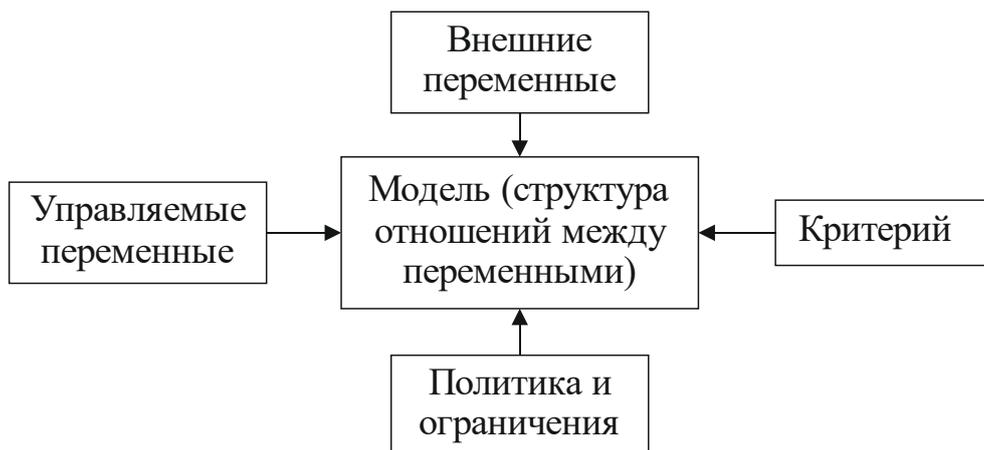


Рис. 26.4 – Связь различных категорий переменных в модели

Второй шаг построения модели связан с определением структуры отношений между переменными.

Модели характеризуются 3 глобальными параметрами:

- *сложностью* (определяемой количеством введенных в рассмотрение переменных);
- *степенью неопределенности рассматриваемых переменных* (от детерминированных до вероятностных);
- *учетом фактора времени* (от статического до динамического).

На рисунке 26.5 представлены 8 вариантов построения модели, соответствующих вершинам параллелепипеда.

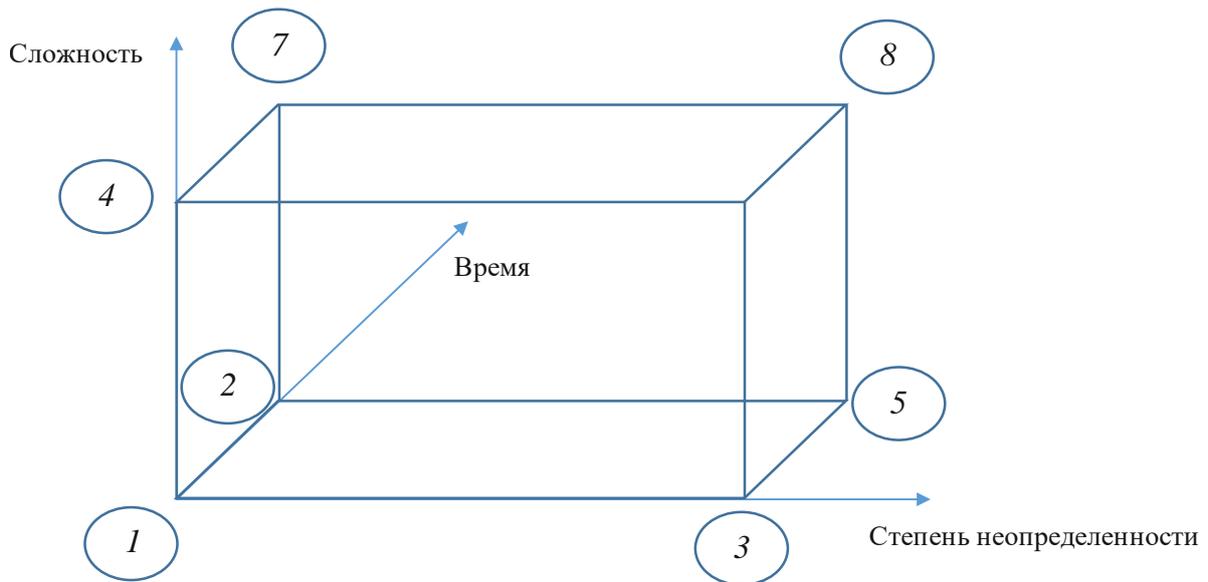


Рис. 26.5 – 8 вариантов построения модели:
 вершина 1 – простая детерминированная статическая модель
 с одной переменной // определение суммарных затрат;
 вершина 2 – детерминированная динамическая модель //
 определение текущей стоимости денежного вклада;
 вершина 3 – вероятностная статическая модель //
 определение объема продаж за текущий месяц;
 вершина 5 – вероятностная динамическая модель //
 определение цены акции

Дальнейшее продвижение к вершинам 4 и 6, 7, 8 параллелепипеда ведет к усложнению модели за счет введения новых переменных, что требует от пользователя более глубоких знаний в предметной области. По мере работы над проблемой используемые модели усложняются (детализируются) по мере того, как идет погружение все глубже в исследуемую область (рис. 26.6).

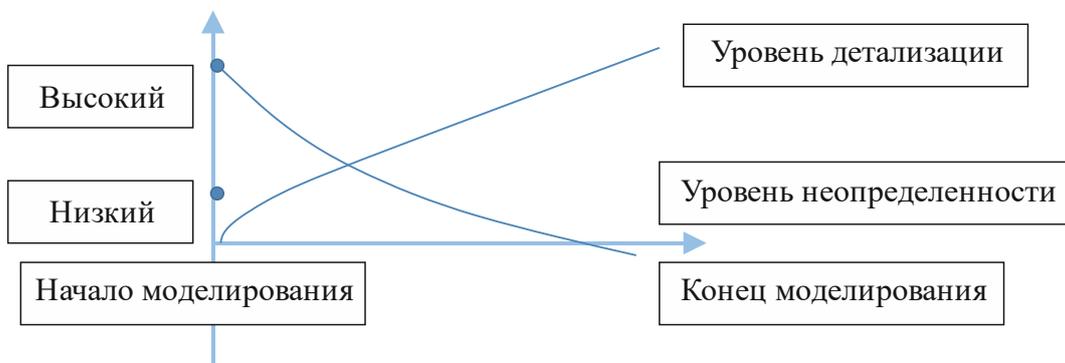


Рис. 26.6 – Усложнение используемых моделей

Последовательная реализация построения модели показана на рисунке 26.7.

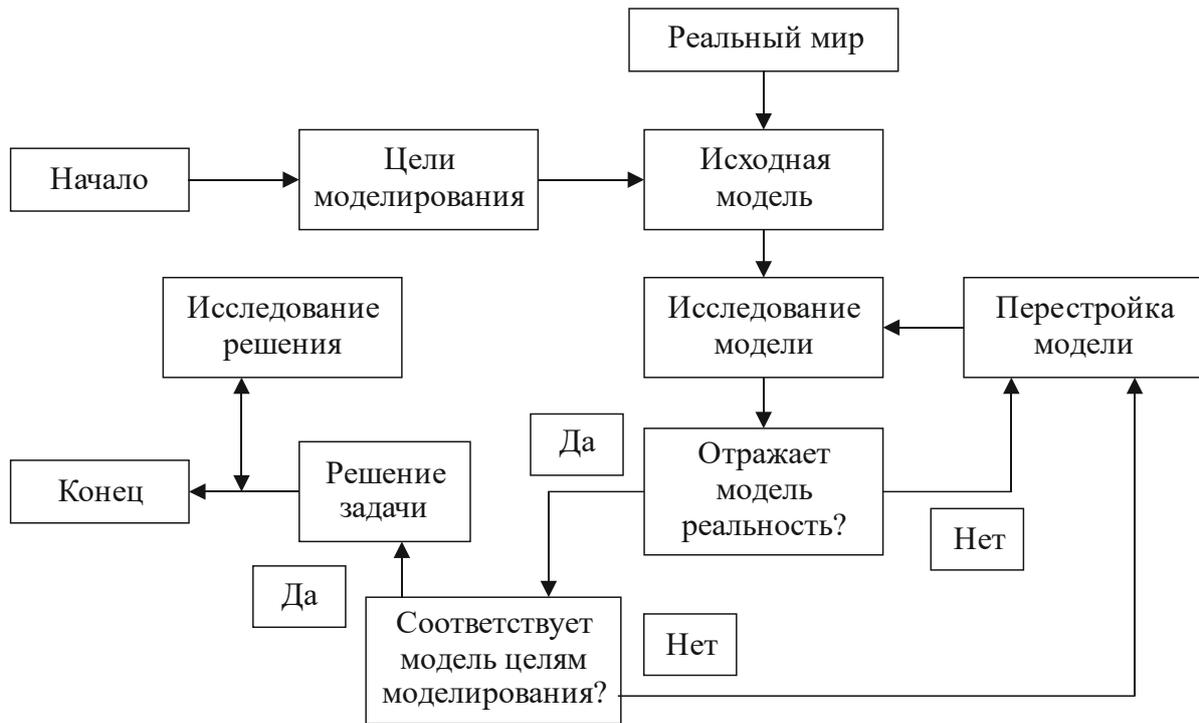


Рис. 26.7 – Последовательная реализация построения модели

Старт происходит после уточнения целей и постановки задач моделирования. Далее строится и оценивается *исходная модель*, которая оказывается или не оказывается удовлетворительной. Далее надо решить, отражает ли модель реальный мир и соответствует ли она установленным целям моделирования, если нет, то она не годится для принятия решений. В этом случае в модель вносятся изменения и процесс итеративно повторяется.

Помощь в построении моделей может оказать *использование 3 положений*:

- 1) уменьшение размерности модели. Производится ее агрегирование и сокращение вводимых в рассмотрение переменных;
- 2) нужно объединять число исходных переменных в значимые и незначимые группы, от рассмотрения незначимых переменных можно отказаться;
- 3) снижение требований к точности входных данных, которая должна быть достаточной для принятия решений.

Типичные ошибки моделирования и их обход

1. Модель никогда полностью не совпадает с реальностью. Поэтому желательна экспериментальная проверка на сравнительно простых случаях. Не стоит экстраполировать действие модели за пределы области, для которой она разработана.

2. Нельзя относиться даже к удачной модели как к безусловной реальности. При возникновении трудностей нужно углубленно исследовать моделируемый объект.
3. При появлении новых данных об объекте модель должна обновляться. Чтобы быть полезной, модель должна служить практическим целям, а не академическим упражнениям.

Контрольные вопросы

1. Дайте определение СППР.
2. Какими особенностями характеризуются СППР?
3. Покажите иерархию информационных систем в компании.
4. Назовите 3 главных компонента структуры СППР.
5. Назовите типы моделей и их функции в СУБМ.
6. Покажите 8 вариантов построения модели.
7. Назовите типичные ошибки моделирования и их обход.

Литература

1. Матвеев, Л. А. Компьютерная поддержка принятия решений / Л. А. Матвеев. – СПб. : Специальная литература, 1998. – 472 с.

27 Алгоритм функционирования программы компьютерного моделирования (PSPICE)

27.1 Основные сведения

Рассмотрим ведущую систему компьютерного моделирования *PSpice*, созданную в США в начале 1980-х гг.

Версии PSpice

PSpice является модификацией известной программы моделирования интегральных схем SPICE (Simulation Program with Integrated Circuit Emphasis), разработанной в Калифорнийском университете. В настоящее время SPICE де-факто стала основной программой моделирования аналоговых электронных цепей.

Первая версия SPICE была разработана в середине 1970-х гг. Она позволяла анализировать линейные и нелинейные цепи во временной области, рассчитывать частотные характеристики линейных цепей. Для анализа линейных резистивных цепей использовался метод узловых напряжений. Библиотека моделей включала модели диода и биполярного транзистора (модель Эберса – Молла).

С течением времени программа совершенствовалась, появлялись новые версии, расширялся список моделей компонентов. В конце 1970-х гг. SPICE стала широко использоваться в промышленности для моделирования электронных схем.

В 1983 г. появилась версия *SPICE2G.6*, разработанная в Калифорнийском университете (Беркли). Это была последняя версия, написанная на языке *FORTRAN*. Для анализа линейных цепей в версии 2G.6 использовался модифицированный метод узловых напряжений. При расчете динамических цепей использовались алгоритмы с автоматическим выбором шага интегрирования. Были переработаны модели биполярных и МОП-транзисторов. Модель биполярного транзистора основана на уравнениях Гуммеля – Пуна.

Заметим, что версия *SPICE2G.6* оказалась очень надежной. Она используется и в некоторых современных программах схемотехнического моделирования.

В 1985 г. была разработана следующая версия программы, *SPICE3*. Она базируется на версии *SPICE2G.6* и является ее развитием. Математическое ядро

программы было *написано на языке С*. Для представления результатов моделирования использовался *графический интерфейс*. В новой версии программы удалось преодолеть некоторые трудности, связанные с *численной неустойчивостью алгоритмов*, используемых при анализе нелинейных цепей. В программу были включены *новые модели электронных компонентов*: длинных линий с потерями, неидеальных ключей и т. д. *Появились модели МОП-транзисторов*, учитывающие физические эффекты, возникающие при уменьшении геометрических размеров приборов.

В середине 1980-х гг. появились программы схемотехнического моделирования, предназначенные *для персональных компьютеров*. В 1984 г. корпорация MicroSim представила версию SPICE для персональных компьютеров, назвав ее *PSpice*.

Первые версии PSpice позволяли моделировать *только аналоговые устройства*. Рассчитывались переходные процессы при действии сигналов различной формы, частотные характеристики, рабочие точки нелинейных приборов.

Современные версии PSpice представляют вычислительную среду, предназначенную для моделирования *аналоговых и цифровых электронных схем*. Существенное *достоинство программы* заключается в возможности моделирования смешанных аналого-цифровых схем без применения вспомогательных устройств согласования аналоговых и цифровых сигналов. Это достигается за счет автоматического использования специальных интерфейсов и значительно облегчает моделирование смешанных аналого-цифровых устройств.

Процедура моделирования электронных схем состоит из трех этапов:

- 1) создание принципиальной схемы;
- 2) моделирование;
- 3) представление результатов моделирования в удобной для пользователя форме.

Для создания графического изображения принципиальных схем *служат редакторы Schematics и Capture*. Они выполняют одновременно функции *управляющей оболочки* для запуска других модулей PSpice. Перед началом моделирования проверяется *правильность соединения* элементов схемы. Разумеется, программа может выявить *только простейшие ошибки*, такие как «*висящий*» узел, к которому подключен только один элемент, или *отсутствие заземления*.

Для представления результатов расчетов в удобной для пользователя форме служит графический постпроцессор Probe. Он выводит на экран графики результатов моделирования и выполняет их математическую обработку.

Базовый набор элементов PSpice включает резисторы, конденсаторы, индуктивные катушки, диоды, биполярные транзисторы, полевые транзисторы, длинные линии, источники напряжения и тока различной формы. Аналоговые интегральные схемы, а также некоторые компоненты (тиристоры, некоторые виды полевых транзисторов) представляются подсхемами, параметры которых задает пользователь.

Базовый набор элементов PSpice *содержит также цифровые функциональные блоки*, выполняющие логические операции. Цифровые элементы могут реализовываться пользователем на основе функциональных блоков [1]. Кроме того, существуют обширные библиотеки серийно выпускаемых цифровых компонентов.

27.2 Функциональные возможности среды PSpice

Возможны *следующие виды анализа*:

- *Bias* – расчет рабочей точки нелинейной резистивной цепи постоянного тока (режим большого сигнала);
- *DC* – анализ резистивных цепей постоянного тока (расчет узловых напряжений, токов и напряжений ветвей);
- *AC* – расчет частотных характеристик линейных цепей (режим малого сигнала, анализируется линейная цепь);
- *Transient* – расчет переходных процессов в нелинейных цепях при действии сигналов произвольной формы;
- *Fourier Analysis* – анализ спектров сигналов, полученных в режиме Transient;
- *Parametric Sweep* – режим вариации параметров цепи;
- *Sensitivity* – расчет характеристик чувствительности линейных цепей к вариациям параметров компонентов в режимах постоянного и переменного тока (метод наихудшего случая);
- *Noise Analysis* – анализ спектральной плотности мощности шума на входе и выходе схемы;
- вероятностный анализ разброса параметров методом Монте-Карло.

27.3 Алгоритм функционирования SPICE

Блок-схема, иллюстрирующая функционирование и связь отдельных модулей программы, показана на рисунке 27.1.

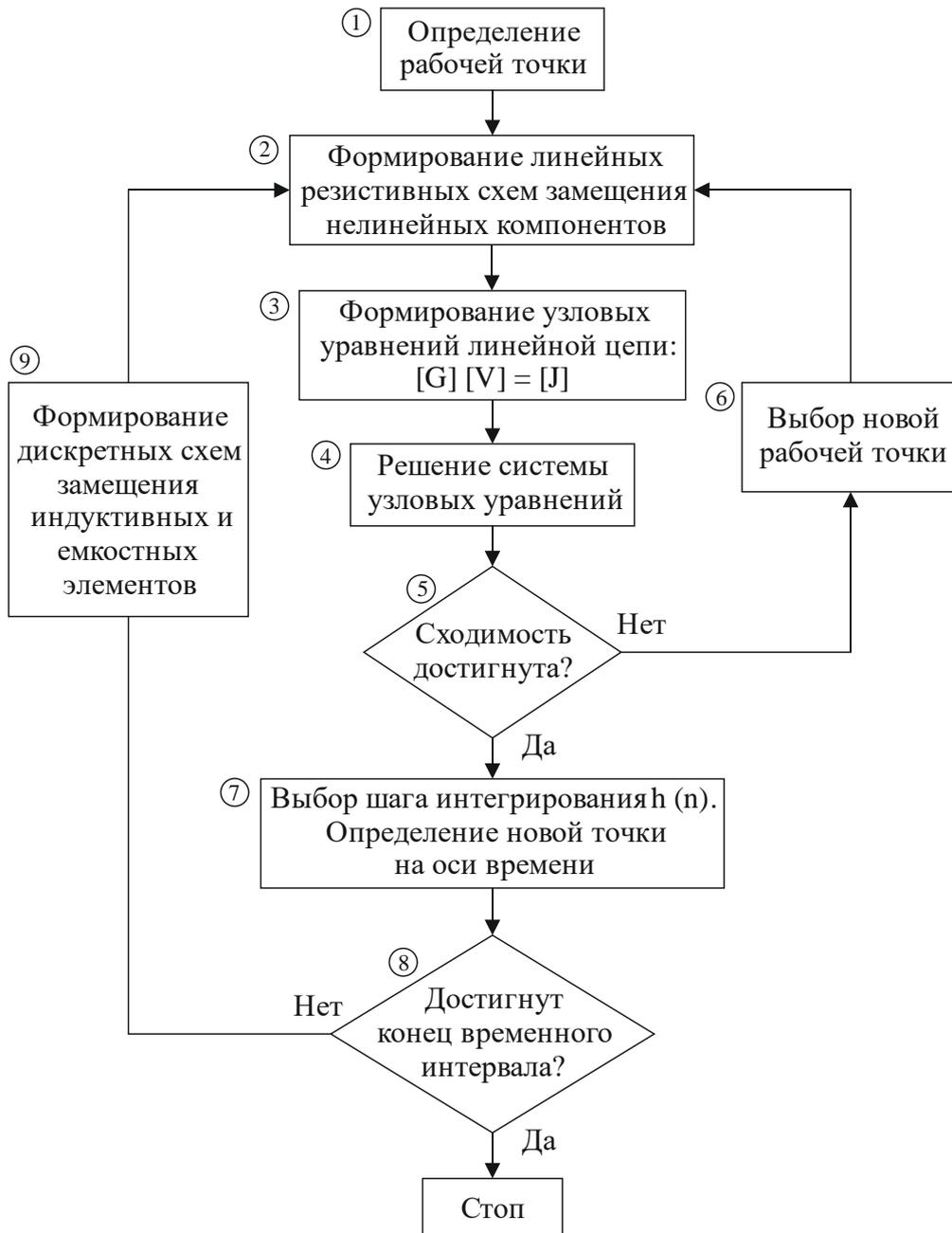


Рис. 27.1 – Функционирование и связь отдельных модулей программы

Перечислим *основные виды анализа*, выполняемые SPICE:

- анализ нелинейных цепей постоянного тока;
- анализ линейных цепей в частотной области;
- анализ нелинейных динамических цепей при действии сигналов произвольной формы.

Остальные виды анализа, такие как статистический анализ, расчет чувствительностей или режим вариации параметров базируются на трех перечисленных режимах. Рассмотрим подробнее перечисленные базовые виды анализа.

Анализ нелинейных резистивных цепей постоянного тока. Этот вид анализа выполняется в следующих случаях:

- при определении рабочей точки нелинейной цепи;
- при расчете частотных характеристик. В этом режиме предварительно выполняется расчет резистивной подсхемы, определяется рабочая точка и строится линейная схема замещения;
- при расчете переходных процессов. В этом режиме рабочая точка определяется на каждом временном шаге;
- в режиме вариации параметров. Расчет рабочей точки выполняется для каждого нового значения варьируемого параметра.

При анализе резистивных цепей индуктивные и емкостные элементы исключаются. Емкостный элемент заменяется разрывом, а зажимы индуктивного элемента закорачиваются.

Простейшим является анализ линейных резистивных цепей. В этом случае используются модули 3 и 4 алгоритма. *Для анализа линейной резистивной цепи* используется модифицированный метод узловых напряжений.

При анализе нелинейных резистивных цепей используется итеративная процедура, включающая блоки 1–6. Эта процедура представляет схемотехническую интерпретацию метода Ньютона – Рафсона. На каждой итерации формируются узловые уравнения эквивалентной линейной резистивной цепи. В результате решения этой системы уравнений определяется уточненное значение рабочей точки.

Расчет частотных характеристик линейных цепей. Расчет частотных характеристик проводится в *два этапа*. Сначала автоматически определяется рабочая точка нелинейной резистивной цепи. Для этого выполняется анализ резистивной цепи, в которой исключены индуктивные и емкостные элементы. Затем нелинейные компоненты заменяются линеаризованными моделями с параметрами, соответствующими рабочей точке. После этого выполняется расчет частотных характеристик. Если в цепи действуют несколько синусоидальных источников, их частоты полагаются одинаковыми. Если действует только один источник, целесообразно установить атрибут $AC = 1$. В этом случае значение реакции будет равно значению передаточной функции.

Необходимо помнить, что при расчете частотных характеристик анализируется линеаризованная схема замещения, искажения сигнала, обусловленные нелинейностью ВАХ элементов, отсутствуют. Поэтому токи и напряжения могут

достигать очень больших величин, совершенно не соответствующих реальной ситуации.

По результатам анализа программа PROBE строит частотные характеристики напряжений и токов, указанных пользователем.

Анализ нелинейных динамических цепей. Для расчета переходных процессов используются неявные методы численного интегрирования – *методы трапеций, Гауна второго порядка или неявный метод Эйлера*. По умолчанию используется метод трапеций. Максимальный шаг интегрирования выбирается пользователем или устанавливается автоматически. PSPICE автоматически выбирает шаг интегрирования, обеспечивающий оптимальное соотношение между точностью и временем расчета.

Интегрирование начинается с момента $t = 0$. Каждый из источников, действующих в схеме, может иметь свою форму. Если источники синусоидальные, их частоты могут быть разными.

Процедура расчета переходного процесса является многошаговой. На каждом шаге интегрирования автоматически определяется рабочая точка – токи и напряжения нелинейных компонентов. При определении рабочей точки нелинейной цепи напряжения и токи источников сигнала полагаются равными нулю, индуктивные элементы заменяются коротким замыканием, а емкостные – разрывом. *Расчет рабочей точки* ведется *итеративным методом Ньютона – Рафсона*. На каждой итерации нелинейные компоненты заменяются линеаризованными схемами замещения, соответствующими режиму этого компонента.

Анализ нелинейных динамических цепей является наиболее сложным видом анализа. В этом режиме используются все модули алгоритма. По результатам анализа программа PROBE строит графики напряжений и токов, указанных пользователем.

Многовариантный анализ. Режим Parametric Sweep. Помимо рассмотренных базовых методов анализа частотных и временных характеристик программа PSPICE дает проектировщику еще один мощный инструмент исследования электронных схем – *параметрический анализ*. В этом режиме пользователь имеет возможность проводить *многовариантный анализ частотных или временных характеристик*. Результатом такого анализа являются семейства кривых, наглядно показывающие, как влияет изменение того или иного параметра на характеристики цепи.

На каждом шаге вариации параметров могут выполняться различные виды анализа. Варьироваться могут напряжения и токи источников, температура компонентов, параметры моделей, глобальные параметры.

Статистический анализ. Элементы реальных цепей всегда имеют случайные отклонения от номинальных значений. Программа PSpice позволяет моделировать характеристики электронных цепей с учетом статистического разброса параметров компонентов. В ходе одного цикла статистического анализа цепь может моделироваться несколько сотен раз. При этом каждый раз моделирование осуществляется с новым набором параметров, задаваемых случайным образом. Например, если предусмотрен графический вывод испытаний с помощью программы Probe, максимальное количество испытаний $n = 400$. На одну диаграмму Probe могут быть выведены результаты 399 испытаний. Отклонения значений элементов от номинальных задаются с помощью генератора случайных чисел. Случайное значение параметра рассчитывается по формуле:

$$x = x_{\text{ном}}(1 + \zeta \Delta),$$

где $x_{\text{ном}}$ – номинальное значение параметра, указанное пользователем; Δ – относительный разброс параметра x ; ζ – центрированная случайная величина, принимающая значения на отрезке $(-1, +1)$.

В программе PSpice имеются генераторы случайных чисел с двумя законами распределения: UNIFORM – равномерное распределение на отрезке $(-1, +1)$; GAUSS – гауссовское распределение на отрезке $(-1, +1)$ с нулевым средним (M) и среднеквадратическим отклонением $\sigma = 0.25$. По умолчанию выбирается *равномерное распределение*.

Статистические испытания по методу Монте-Карло могут проводиться при расчете режима по постоянному току, анализе переходных процессов или расчете частотных характеристик. Проведение статистического анализа по методу Монте-Карло позволяет определить многие важные характеристики электронных цепей. Например, по величине отклонения частотных характеристик можно судить о чувствительности цепи.

При статистическом анализе предусматривается разнообразная статистическая обработка результатов моделирования. Возможны следующие *виды обработки*:

- расчет максимального отклонения текущей реализации от номинальной;
- расчет максимального значения в каждой реализации;
- расчет минимального значения в каждой реализации.

Контрольные вопросы

1. Определите функциональные возможности среды PSpice.
2. Разберите алгоритм функционирования SPICE.
3. Какой базовый набор элементов содержит PSpice?
4. В каких случаях выполняется нелинейный анализ?
5. Как выполняется моделирование цифровых и аналого-цифровых устройств?
6. Что реализует многовариантный анализ частотных или временных характеристик?
7. Какие задачи решает статистический анализ?

Литература

1. Разевиг, В. Д. Система схемотехнического моделирования и проектирования Design Center PSpice / В. Д. Разевиг. – М. : Родник Софт, 1996. – 272 с.

28 Среда многоуровневого компьютерного моделирования

Комплекс программ «Среда многоуровневого моделирования сложных технических управляемых систем» *предназначен* для автоматизированного решения задач исследования и функционального проектирования на основе автоматизированных экспериментов над многоуровневыми компонентными цепями СТУС.

Комплекс программ позволяет:

- формировать представленные в графическом виде многоуровневые КЦ СТУС, включающие в себя подцепи различных подсистем с неоднородными векторными связями, модели сценариев автоматизированных экспериментов, автоматизирующих решение задач исследования и функционального проектирования СТУС и представленных алгоритмическими подцепями на логическом уровне МКЦ, а также панели визуализации и интерактивного управления, представляющие собой наборы визуальных компонентов;
- производить анализ функционирования СТУС, представленных компонентными подцепями объектного уровня, сформированными в формате языка моделирования химико-технологических систем, между компонентами которой протекают информационные, мультифизические энергетические и многокомпонентные вещественные потоки;
- выполнять сценарии экспериментов, представленные на логическом уровне многоуровневой КЦ СТУС в виде алгоритмических подцепей и направленные на исследование и функциональное проектирование СТУС, в том числе на разработку и отладку сценариев функционирования устройства управления;
- осуществлять визуализацию результатов анализа КЦ СТУС и наблюдаемых характеристик реального объекта, а также осуществлять формирование и изменение воздействий пользователя на модель и на реальный объект;
- производить автоматизированное документирование результатов решения задач исследования и функционального проектирования СТУС с возможностью автоматизированного синтеза сценариев, на основе которых функционирует измерительно-управляющий контроллер.

Комплекс программ «Среда моделирования сложных технических управляемых систем» функционирует под управлением операционной системы *Windows*. Для его разработки использовались следующие языки и средства объектно-ориентированного программирования, формирования и обработки запросов к базам данных, представления, визуализации и документирования результатов моделирования и измерения:

- *объектно-ориентированный язык программирования Visual C++* с библиотекой классов и функций MFC, применяемый для непосредственной разработки основных моделей комплекса программ [1];
- *динамически загружаемая библиотека ftd2xx.dll* – библиотека функций для работы с FDTI-устройствами, подключаемыми к компьютеру по шине USB. В комплексе программ таким устройством является универсальный измерительно-управляющий контроллер X-Mega, функционирующий на основе сценариев, написанных на языке X-Robot [2–3].

Для автоматизации вычислительных и натурных экспериментов, направленных на решение задач исследования и функционального проектирования СТУС в составе комплекса программ, структура которого приведена на рисунке 28.1, входят:

- *многослойный редактор многоуровневой компонентной цепи сложной технической управляемой системы*, позволяющий в графическом виде формировать многоуровневые КЦ СТУС на трех взаимосвязанных слоях с применением трех подязыков языка МКЦ;
- *программно-алгоритмические средства языка моделирования СТУС*, представленные интерпретатором языка, осуществляющим построение КЦ исследуемой технологической системы на основе ее графической модели, представленной на объектном слое редактора МКЦ;
- универсального вычислительного ядра, реализующего представленный на рисунке 28.2 алгоритм анализа КЦ СТУС.

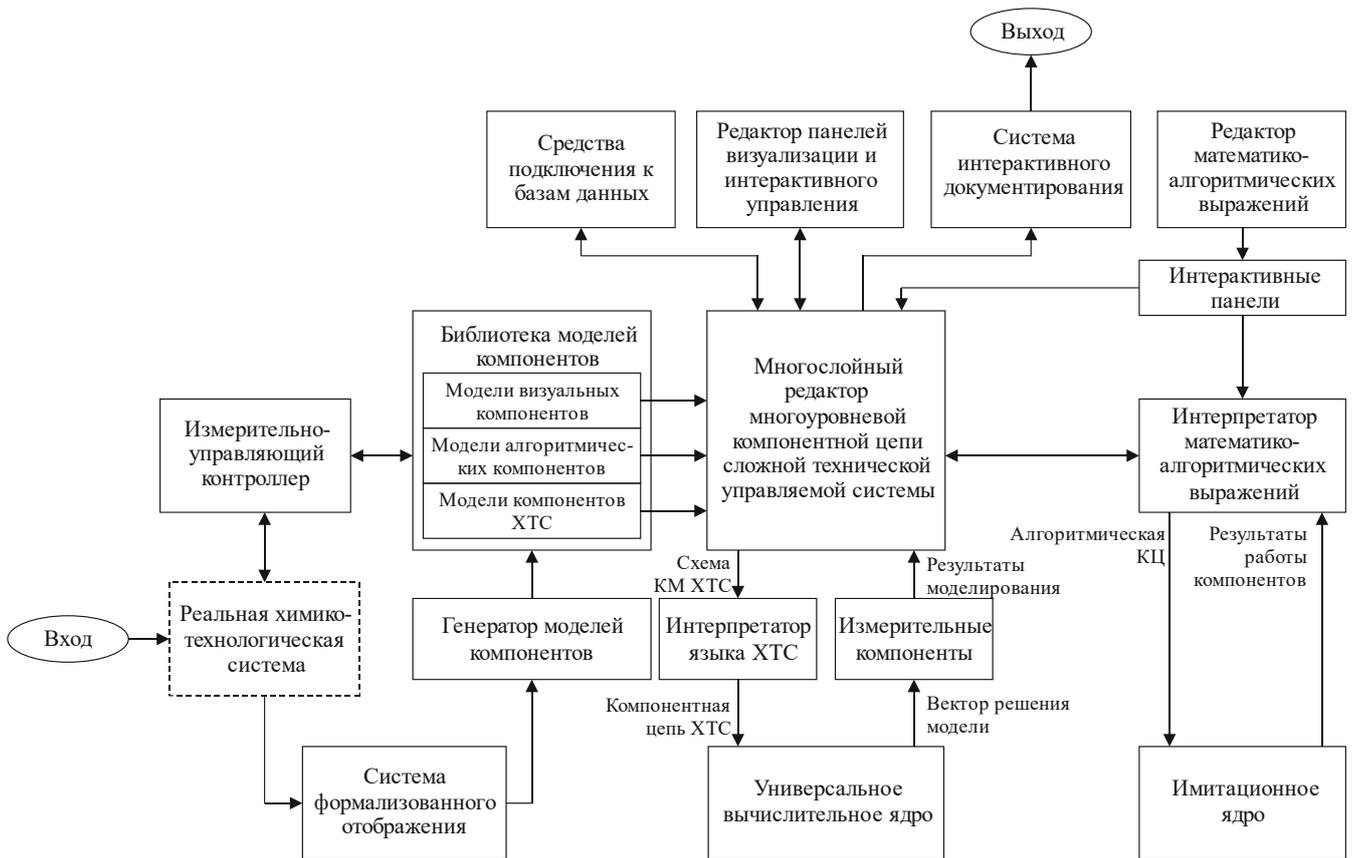


Рис. 28.1 – Структура комплекса программ «Среда автоматизации экспериментов над сложными техническими управляемыми системами»

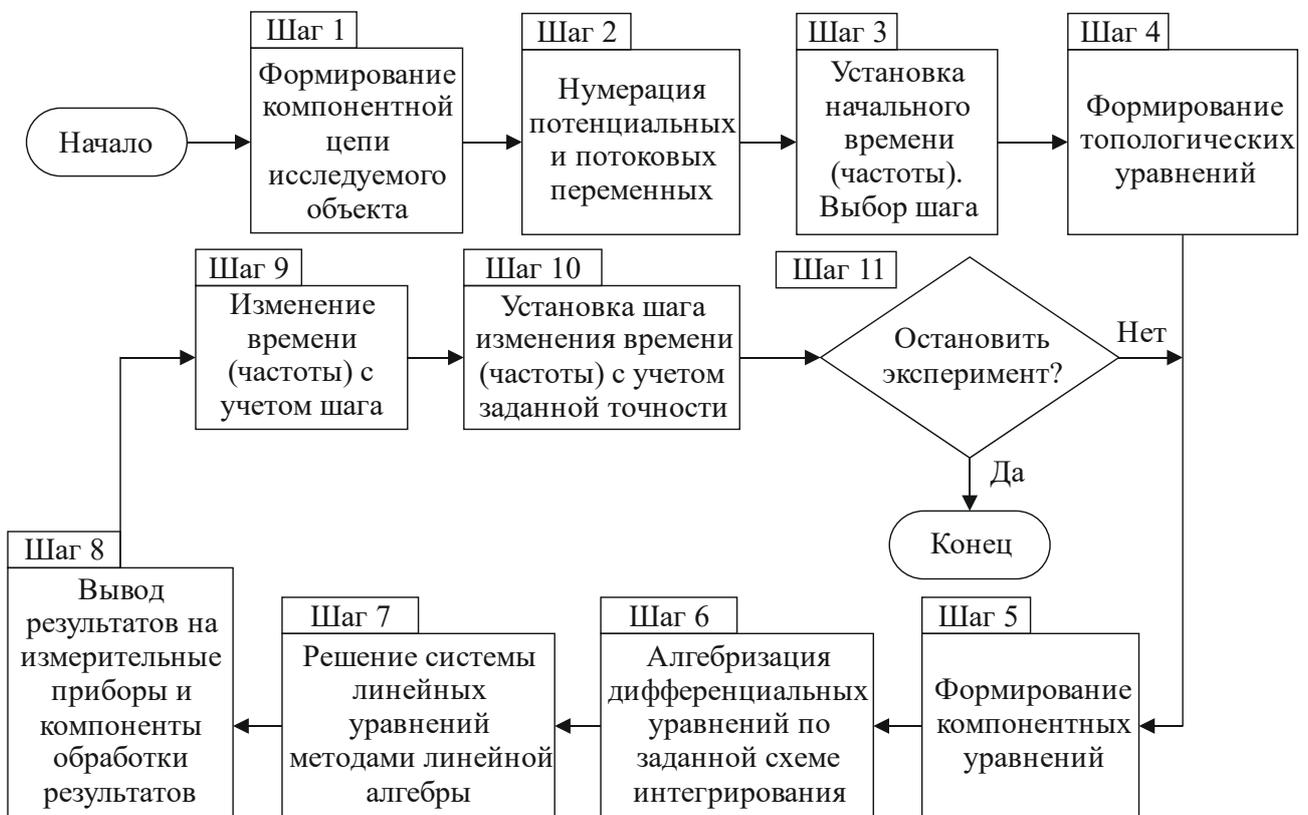


Рис. 28.2 – Алгоритм вычислительного эксперимента над КМ технического объекта в формате МКЦ

Последовательность действий вычислительного эксперимента, представленная на рисунке 28.2, легла в основу реализации универсального вычислительного ядра и состоит из следующих шагов:

Шаг 1. Формирование КЦ исследуемого объекта (системы) заключается в установке и соединении всех компонентов, являющихся моделями его элементов.

Шаг 2. Нумерация потенциальных и потоковых переменных заключается в преобразовании локального координатного базиса (ЛКБ) всех компонентов в глобальном координатном базисе (ГКБ) сформированной КЦ.

Шаг 3. Установка начального значения времени (частоты) и выбор шага производится согласно выбранному режиму анализа и используемым средствам визуализации результатов.

Шаг 4. Формирование топологических уравнений вида выполняется на основе информации о топологии цепи, полученной путем опроса топологического блока всех входящих в исследуемую КЦ компонентов. Она представляет собой данные об инцидентности ветвей каждого компонента соответствующим узлам цепи.

Шаг 5. Формирование компонентных уравнений вида осуществляется путем опроса вычислительного блока входящих в КЦ компонентов. На этом шаге производится линеаризация нелинейных алгебраических уравнений вида на основе схемы Ньютона:

$$\sum_i \frac{\partial f}{\partial V_i} \Big|_{V(t-h)} \cdot V(t) = f(V) \Big|_{V(t-h)}.$$

Шаг 6. Линеаризация дифференциальных уравнений осуществляется на основе заданной схемы интегрирования. В универсальном вычислительном ядре, являющемся программно-алгоритмической реализацией алгоритма ВЭ, реализованы следующие схемы алгебраизации дифференциальных уравнений.

Явная схема Эйлера позволяет осуществлять преобразование дифференциального уравнения к виду:

$$\sum_i \Psi_i \cdot V_i(t+h) = \sum_i \Psi_i \cdot V_i + h \cdot f(V(t)).$$

Неявная схема Эйлера предполагает, что в результате алгебраизации уравнение примет вид:

$$\sum_i \left(\Psi_i - h \cdot \frac{\partial f(V)}{\partial V_i} \Big|_{V=V(t)} \right) \cdot V(t+h) = \sum_i \Psi_i \cdot V_i(t) + h \cdot f(V(t)).$$

Согласно неявной схеме трапеций дифференциальное уравнение после линеаризации будет иметь вид:

$$\sum_i \left(\Psi_i - \frac{h}{2} \cdot \frac{\partial f}{\partial V_i} \Big|_{V=V(t)} \right) \cdot V(t+h) = \sum_i \Psi_i \cdot V_i(t) + \frac{h}{2} \cdot f \cdot V(t) .$$

Шаг 7. Решение системы линейных алгебраических уравнений может выполняться различными методами линейной алгебры. В настоящее время в качестве основного выбран метод Жордана с выбором главного элемента по строке.

Шаг 8. Вывод результатов на измерительные приборы и компоненты обработки результатов выполняется с помощью измерительных компонентов, осуществляющих выборку значений соответствующих переменных из общего решения КЦ. Обработка результатов производится основанными на численных методах алгоритмами. Полученные результаты передаются на визуальные компоненты. Также они могут быть использованы при решении задач параметрической оптимизации.

Шаг 9. Изменение времени (частоты) производится с учетом заданного или выбранного на основе заданной точности шага.

Шаг 10. Установка шага изменения времени (частоты) выполняется с учетом заданной точности на основе результатов двух или более предыдущих итераций полного вектора решения.

Шаг 11. Проверка останова эксперимента выполняется как на основе текущего значения времени (частоты), достигнутой точности решения нелинейных уравнений при постоянных входных воздействиях или по команде останова вычислительного эксперимента.

Измерительные компоненты, осуществляющие выборку соответствующих значений из вектора решения и далее их передачу с объектного уровня МКЦ СТУС на ее логический уровень (табл. 28.1):

- *программно-алгоритмические средства языка МАК*, реализованные в виде интерпретатора математико-алгоритмических выражений, имитационного ядра, в основу реализации которого положен алгоритм передачи сообщений, и интерактивных панелей, включающих в свой состав редактор математико-алгоритмических выражений. Помимо этого, они включают в себя средства подключения к базам данных и средства интерактивного документирования;

- средства языка виртуальных инструментов и приборов, представленные редактором панелей визуализации и интерактивного управления и библиотекой моделей визуальных компонентов;
- библиотека моделей компонентов (БМК), в которой реализованы инструментальные средства хранения и использования компонентов всех трех подязыков языка МКЦ, а также средства ее пополнения, представленные в виде генератора моделей компонентов и интерактивных панелей.

Таблица 28.1 – Компоненты-измерители языка моделирования СТУС

Измеритель	Отображение на объектном слое	Отображение на логическом слое	Математическая модель
Измеритель потенциальной переменной V_n			$V_{n2} = V_{n1}$
Измеритель поточковой переменной V_b			$V_{n3} = V_{b1}$
Измеритель разности потенциалов переменных dV_n			$V_{n3} = V_{n1} - V_{n2}$

Таким образом, комплекс программ моделирования химико-технологических систем, структура которого представлена на рисунке 28.1, позволяет автоматизировать решение задач исследования и функционального проектирования СТУС. Его интеграция с реальным объектом позволяет применять полученные с помощью модели результаты к реальной СТУС, тем самым проводя апробацию результатов и окончательную отладку сценариев управления до окончательного их программирования в контроллер.

Контрольные вопросы

1. Для чего предназначена среда многоуровневого моделирования МАРС?
2. Какие программно-инструментальные средства использовались для разработки СММ МАРС?
3. Охарактеризуйте состав БМК СММ МАРС.

4. Перечислите основные шаги алгоритма вычислительного эксперимента над компьютерной моделью.
5. Чем представлены средства языка для измерителей сигналов?

Литература

1. Мешков, А. В. Visual C++ и MFC / А. В. Мешков, Ю. В. Тихомиров. – СПб. : БХВ-СПб, 2001. – 468 с.
2. FTDI Chip Home Page [Электронный ресурс]. – Режим доступа: <http://www.ftdichip.com/>, свободный (дата обращения 20.02.2020).
3. Микроконтроллеры семейства AVR XMEGA [Электронный ресурс]. – Режим доступа: http://www.atmel.com/ru/ru/products/microcontrollers/avr/avr_xmega.aspx, свободный (дата обращения: 19.02.2020).

Список сокращений

- АЦП* – амплитудно-цифровые преобразователи
АЧХ – амплитудно-частотная характеристика
БД – база данных
БЗ – бетонные заводы
БИХ – бесконечная импульсная характеристика
БМК – библиотека моделей компонентов
БОД – блок обработки данных
ВАХ – вольт-амперная характеристика
ВК – вентильный комплект
ГДС – гибридные динамические системы
ГМК – генератор моделей компонентов
ГПН – генератор пилообразного напряжения
ДС – динамические системы
ИЭ – импульсный элемент
КА – конечные автоматы
КЛА – кусочно-линейный агрегат
КПФ – кусочно-постоянные функции
КС – карта состояний
КЦ – компонентная цепь
КЭ – конечные элементы
ЛПР – лицо, принимающее решения
ЛПФ – логарифмоподобные функции
ЛСН-системы – линейные стационарные непрерывные системы
МАК – моделирование алгоритмических конструкций
МКЦ – метод компонентных цепей
МКЭ – метод конечных элементов
ММ – математическая модель
ММК – метод Монте-Карло
ОДР – обобщенный дискретный регулятор
ОС – обратная связь
ОСС – операторно-структурные схемы
ПМ – передаточные матрицы
ПР – принятие решений

ПФ – передаточные функции

РИ – распределитель импульсов

РЦФ – рекурсивный цифровой фильтр

СКМ – среда компьютерного моделирования

СЛАУ – система линейных алгебраических уравнений

СМ – статистическое моделирование

СММ – среда многоуровневого компьютерного моделирования

СМО – системы массового обслуживания

СП – сети Петри

СППР – система поддержки принятия решений

СС – схема сравнения

СТУ – сложные технические управляемые системы

СУ – синхронизирующее устройство

ФИ – формирователь импульсов

ФЧХ – фазочастотная характеристика

ЦАП – цифро-аналоговый преобразователь

ЦВУ – цифровое вычислительное устройство

ЦФ – целевая функция

ЭЭС – экономико-экологическая система

Учебное издание

**Вячеслав Михайлович Дмитриев
Тарас Викторович Ганджа
Татьяна Евгеньевна Григорьева**

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СИСТЕМ

Курс лекций

Корректор А. Н. Миронова
Оригинал-макет Г. Д. Дурягиной

Подписано в печать 22.05.2020. Формат 60x84¹/₁₆.
Бумага офсетная. Гарнитура Times.
Усл. печ. л. 15,11.
Тираж 150 экз. Заказ № .

Издательство «Эль Контент»
634061, г. Томск, ул. Киевская, д. 57, оф. 27

ISBN 978-5-4332-0284-9



9 785433 202849