

**С.А. Лепешкин, В.А. Гончаренко, А.А. Шульгин**

# **Моделирование и проектирование систем**

**Учебное пособие**

**Часть 1**

## **Моделирование систем**

С.А. Лепешкин, В.А. Гончаренко, А.А. Шульгин

# Моделирование и проектирование систем

*Учебное издание «Моделирование и проектирование систем. Часть 1. Моделирование систем» утверждено в качестве учебного пособия и рекомендовано ученым советом факультета автоматизированных систем управления войсками Военно-космической академии имени А.Ф. Можайского для обучающихся по основным профессиональным образовательным программам высшего образования – программам специалитета по специальности «Применение и эксплуатация автоматизированных систем специального назначения»,  
протокол от 22 сентября 2016 года, № 2*

Часть 1

## Моделирование систем



Санкт-Петербург  
2016

Рецензенты:  
доктор технических наук, профессор **Ю.С. Мануйлов**,  
доктор технических наук, профессор **А.Г. Басыров**

**Моделирование и проектирование систем. Ч. 1. Моделирование систем:**  
учебное пособие / С.А. Лепешкин, В.А. Гончаренко, А.А. Шульгин. – СПб.:  
ВКА имени А.Ф. Можайского, 2016. – 244 с.

В предлагаемом учебном пособии изложен материал первого раздела «Моделирование систем» учебной дисциплины «Моделирование и проектирование систем», ориентированный на обучающихся по основным профессиональным образовательным программам высшего образования.

Рассматриваются общие вопросы теории моделирования, сущность и наиболее важные аспекты аналитического и имитационного моделирования, способы моделирования случайных величин, процессов и полей, основы моделирования боевых действий и систем военного назначения. Определен порядок планирования и проведения компьютерных экспериментов, а также наиболее распространенные методы обработки экспериментальных данных.

Предисловие, подразд. 3.3, 5.1, 5.2 и заключение разработал и подготовил к изданию кандидат технических наук С.А. Лепешкин, разд. 1, 2, 4 – кандидат технических наук, доцент В.А. Гончаренко, подразд. 3.1, 3.2, 5.3 – кандидат технических наук А.А. Шульгин.

© ВКА имени А.Ф. Можайского, 2016

---

Подписано к печ. 10.10.2016  
Гарнитура Times New Roman  
Уч.-печ. л. 31,00

Формат печатного листа 445×300/8  
Уч.-изд. л. 15,25  
Заказ 3313

Бесплатно

---

Типография ВКА имени А.Ф. Можайского

## Оглавление

Список сокращений.....	6
Предисловие.....	7
1. Основы теории моделирования .....	9
1.1. Введение в моделирование.....	9
1.1.1. Основные понятия моделирования и проектирования .....	9
1.1.2. История развития методов моделирования.....	11
1.1.3. Роль методов моделирования в разработке и анализе систем военного назначения .....	14
1.2. Общие вопросы теории моделирования .....	16
1.2.1. Понятие модели.....	16
1.2.2. Классификация моделей.....	18
1.2.3. Математические модели.....	21
1.2.4. Основные этапы моделирования .....	24
1.3. Особенности технологии моделирования .....	27
1.3.1. Построение концептуальной модели .....	27
1.3.2. Подготовка исходных данных .....	30
1.3.3. Разработка модели, выбор методов и средств моделирования .....	34
1.3.4. Проверка адекватности и корректировка модели.....	37
1.3.5. Планирование экспериментов с моделью и анализ результатов моделирования .....	40
1.4. Разработка и формализация моделей сложных систем.....	41
1.4.1. Автоматизированные системы как объекты моделирования.....	41
1.4.2. Выбор уровня детализации и декомпозиция моделей .....	45
1.4.3. Упрощение моделей и подбор параметров .....	48
1.4.4. Формализованные схемы описания моделей сложных систем.....	50
2. Аналитическое моделирование.....	57
2.1. Элементы вероятностных моделей сложных систем .....	57
2.1.1. Основы теории массового обслуживания.....	57
2.1.2. Обзор основных случайных процессов .....	60
2.1.3. Моделирование рабочей нагрузки системы .....	62
2.1.4. Показатели эффективности СМО.....	66
2.1.5. Законы сохранения в теории массового обслуживания.....	67
2.2. Анализ систем с простейшими потоками событий .....	71
2.2.1. Одноканальная система с отказами $M/M/1/0$ .....	71
2.2.2. Многоканальная система с отказами $M/M/n/0$ .....	72
2.2.3. Одноканальная система с неограниченной очередью $M/M/1$ .....	73
2.2.4. Многоканальная система с неограниченной очередью $M/M/n$ .....	74
2.2.5. Многоканальная система с ограниченной очередью $M/M/n/m$ .....	75

2.3. Анализ полумарковских систем массового обслуживания .....	76
2.3.1. Классификация методов исследования СМО .....	76
2.3.2. Метод фаз Эрланга .....	79
2.3.3. Метод вложенных цепей Маркова .....	81
2.4. Анализ систем с произвольными потоками событий .....	84
2.4.1. Метод интегральных уравнений Линдли .....	84
2.4.2. Приближённые методы расчёта систем .....	88
2.5. Основы теории сетей массового обслуживания .....	90
2.5.1. Основные понятия и классификация сетей массового обслуживания .....	90
2.5.2. Экспоненциальные сети обслуживания .....	92
2.5.3. Преобразование потоков в сетях массового обслуживания .....	95
2.5.4. Приближённые методы расчёта сетей массового обслуживания .....	98
3. Моделирование случайных величин, процессов и полей .....	101
3.1. Моделирование случайных величин .....	101
3.1.1. Общая характеристика методов моделирования случайных величин .....	101
3.1.2. Моделирование случайных величин с негауссовским распределением .....	101
3.1.3. Моделирование случайных величин с гауссовским распределением .....	109
3.1.4. Метод суперпозиции .....	111
3.2. Моделирование случайных процессов .....	112
3.2.1. Модели случайных процессов .....	112
3.2.2. Методы моделирования случайных процессов .....	122
3.3. Моделирование случайных полей .....	128
3.3.1. Основные понятия и подходы моделирования случайных полей ...	128
3.3.2. Математические модели поля изображения на основе стохастических дифференциальных уравнений в частных производных .....	135
4. Имитационное моделирование систем .....	141
4.1. Основы имитационного моделирования .....	141
4.1.1. Основные понятия имитационного моделирования .....	141
4.1.2. Организация и принципы разработки имитационных моделей .....	143
4.1.3. Алгоритм имитационного моделирования .....	146
4.1.4. Генерация случайных чисел .....	149
4.2. Подготовка и проведение имитационного моделирования .....	151
4.2.1. Испытания и исследование свойств имитационной модели .....	151
4.2.2. Планирование имитационных экспериментов .....	157
4.2.3. Обработка результатов имитационного моделирования .....	160

4.3. Программные средства имитационного моделирования .....	163
4.3.1. Инструментальные средства моделирования.....	163
4.3.2. Классификация специализированных программных средств моделирования.....	167
4.3.3. Обзор специализированных языков моделирования.....	171
4.4. Основы построения языка GPSS .....	175
4.4.1. Объекты языка GPSS .....	175
4.4.2. Основные блоки GPSS и построение простейших моделей.....	180
4.4.3. Объекты вычислительной категории .....	186
4.5. Технология моделирования в среде GPSS World .....	189
4.5.1. Разработка моделей в среде GPSS World .....	189
4.5.2. Окна GPSS.....	193
4.5.3. Этапы моделирования в GPSS World.....	194
5. Моделирование боевых действий и систем военного назначения.....	196
5.1. Теоретические основы моделирования боевых действий .....	196
5.1.1. Сущность, роль и значение моделирования боевых действий.....	196
5.1.2. Классификация моделей и методы моделирования боевых действий.....	198
5.1.3. Основные цели и задачи моделирования боевых действий. Общий порядок моделирования.....	202
5.1.4. Оценка точности моделей и достоверности результатов моделирования.....	207
5.2. Моделирование сложных систем военного назначения .....	210
5.2.1. Имитационно-моделирующий комплекс как основа построения автоматизированных учебно-тренировочных средств военного назначения.....	210
5.2.2. Моделирование автоматизированных систем радиовидения воздушно-космического базирования.....	217
5.3. Перспективы развития и применение моделирования.....	227
5.3.1. Моделирование на суперкомпьютерах, кластерах .....	227
5.3.2. Решение исследовательских задач .....	230
5.3.3. Искусственный интеллект и моделирование .....	232
Заключение.....	237
Список литературы .....	239

## СПИСОК СОКРАЩЕНИЙ

АК УТКП	– автоматизированный комплекс учебно-тренировочного командного пункта
АОС	– автоматизированная обучающая система
АС	– автоматизированная система
АСУ	– автоматизированная система управления
ВВТ	– вооружение и военная техника
ВС	– вычислительная система
ГСЧ	– генератор случайных чисел
ДНА	– диаграмма направленности антенны
ИВС	– информационно-вычислительная система
ИД	– исходные данные
ИМ	– имитационная модель
ИМК	– имитационно-моделирующий комплекс
ИПД	измерения параметров движения
ИС	– информационная система
КШУ	– командно-штабные учения
ОЗУ	– оперативное запоминающее устройство
ОУ	– орган управления
ПВФ	– пространственно-временные фильтры
ПЛС	– преобразование Лапласа – Стильтеса
ПР	– плотность распределения
РЛИ	– радиолокационное изображение
РЛС	– радиолокационная станция
РСА	– радиолокатор с синтезированной апертурой антенны
СеМО	– сеть массового обслуживания
СМВН	– система моделирования военного назначения
СМО	– система массового обслуживания
СПД	– система передачи данных
СЧА	– системный числовой атрибут
ТМО	– теория массового обслуживания
ТСО	– технические средства обучения
УКК	– учебные компьютерные классы
УТК ПВС	– учебно-тренировочные комплексы подготовки военных специалистов
ФНВ	– функция непрерывного времени
ФР	– функция распределения
ФЦА	– фазовый центр антенны
ЭВМ	– электронная вычислительная машина

## ПРЕДИСЛОВИЕ

Моделирование – один из мощнейших инструментов анализа, которыми располагают специалисты, занимающиеся исследованием, разработкой и эксплуатацией сложных систем, особенно – автоматизированных систем специального назначения. Идея моделирования проста, и интуитивно понятен выигрыш при её реализации на практике. Каждый инженер-исследователь должен уметь использовать современные методы моделирования.

Объектами профессиональной деятельности выпускников, освоивших программу специалитета по специальности «Применение и эксплуатация автоматизированных систем специального назначения», являются, в том числе, методы и средства моделирования, проектирования, экспериментального исследования и управления автоматизированными системами специального назначения.

Основной целью учебной дисциплины «Моделирование и проектирование систем» является формирование у обучающихся систематизированных знаний в сфере моделирования и проектирования автоматизированных систем специального назначения, оценки эффективности их функционирования и применения, необходимых в профессиональной деятельности военного специалиста.

Эффективное освоение материалов учебной дисциплины «Моделирование и проектирование систем» возможно на основе знаний, умений и навыков, полученных обучающимися в ходе изучения дисциплин: математический анализ, теория вероятностей и математическая статистика, основы теории управления, основы теории случайных процессов и обработки информации в АСУ, информатика и программирование.

Актуальность изучения вопросов моделирования систем в рамках основной профессиональной образовательной программы высшего образования по специальности обусловлена необходимостью формирования у обучающихся способностей:

- применять методы математического анализа и моделирования, теоретического и экспериментального исследования;
- использовать инструментальные средства компьютерного моделирования для решения различных исследовательских и профессиональных задач;
- создавать и применять математические модели объектов и процессов, выбирать методы их исследования и разрабатывать алгоритмы их реализации;
- решать задачи анализа и синтеза элементов автоматизированных систем специального назначения;
- разрабатывать программы экспериментальных исследований, включая выбор технических средств и обработку результатов.



Учебное пособие «Моделирование и проектирование систем. Часть 1. Моделирование систем» рекомендуется курсантам при освоении первого раздела дисциплины «Моделирование и проектирование систем». Также оно окажет неоценимую помощь при выполнении курсового и дипломного проектирования, поскольку моделирование процессов и систем является неотъемлемой частью решения любой исследовательской задачи, ставящейся в рамках курсовых и выпускных квалификационных работ. Целесообразно использование учебного пособия и адъюнктами при проведении диссертационных исследований.

Первый раздел учебного пособия носит вводный характер. В нём сформулированы центральные понятия теории моделирования (модель, система, анализ, синтез и т. д.), дана классификация моделей, выделены основные этапы моделирования, отмечены особенности технологии моделирования, определены требования к разработке и формализации моделей сложных систем.

Во втором разделе рассмотрены вопросы аналитического моделирования и современные подходы к процессу разработки и анализа систем и сетей массового обслуживания.

В третьем разделе изложена сущность моделирования случайных величин, процессов и полей, применяемых при построении и исследовании сложных систем. Рассмотрены основные модели, учитывающие вероятностный характер реальных событий и процессов, предложены и обоснованы методы моделирования случайных величин, процессов и полей с заданными характеристиками.

Вопросы имитационного моделирования систем подробно разобраны в четвёртом разделе. В нём проиллюстрированы широкие возможности имитационного моделирования, подчеркнута важность овладения его принципами и методами, приведены этапы подготовки и проведения имитационного моделирования. Проведён обзор наиболее распространённых программных средств имитационного моделирования, а основы технологии моделирования в среде GPSS World рассмотрены более подробно.

Пятый раздел посвящён различным аспектам моделирования в военной сфере. Изложены теоретические основы моделирования боевых действий, разобраны примеры моделирования сложных систем военного назначения, рассмотрены перспективы развития моделирования на современном этапе с использованием суперкомпьютеров и систем искусственного интеллекта, а также возможности применения методов моделирования при решении научных задач в интересах Вооружённых Сил Российской Федерации.

Авторы уверены, что качественное и полное освоение материалов учебного пособия «Моделирование и проектирование систем. Часть 1. Моделирование систем» является залогом успешного становления будущего инженера по эксплуатации автоматизированных систем специального назначения и основой дальнейшего совершенствования военного специалиста.

# 1. Основы теории моделирования

## 1.1. Введение в моделирование

### 1.1.1. Основные понятия моделирования и проектирования

Рассмотрим взаимосвязь понятий «проектирование» и «моделирование».

**Проектирование** – процесс определения архитектуры, компонентов, интерфейсов и других характеристик системы или её части. Для решения задач проектирования разрабатывается **проект**.

**Проект** (от лат. *projectus* – выдвинутый; буквально – брошенный вперёд) – это многозначное понятие, которое в зависимости от контекста может означать:

- замысел, план, прототип, прообраз какого-либо объекта;
- предварительный текст документа;
- совокупность документов (расчётов, чертежей) для создания какого-либо сооружения или изделия.

Проектирование современных сложных технических систем было бы невозможно без предварительного изучения свойств проектируемого объекта с помощью физических или абстрактных аналогов системы. Такие аналоги называют моделями. Таким образом, **модель** – *это физическая или абстрактная система, адекватно представляющая объект исследования*.

Сам процесс исследования объектов познания на их моделях называют **моделированием**. Таким образом, моделирование – это наиболее распространенный способ исследования различных процессов и систем. Целью такого исследования как раз и является создание нового объекта или модернизация существующего, что обозначается понятием «проектирование».

**Моделирование** (в широком смысле) является **основным методом** исследований во всех областях знаний и научно обоснованным методом оценок характеристик сложных систем, используемым для принятия решений в различных сферах деятельности.

**Система** (от др.-греч. σύστημα – целое, составленное из частей) – множество элементов, находящихся в отношениях и связях друг с другом, которое образует определённую целостность, единство.

Системы информатики и вычислительной техники, автоматизированные системы обработки информации и управления, информационные системы относятся к **классу больших систем**, этапы проектирования, внедрения, эксплуатации и эволюции которых в настоящее время невозможны без использования различных видов моделирования.

**Существующие и проектируемые системы** можно эффективно исследовать с помощью **математических моделей** (аналитических и имитационных), реализуемых на современных ЭВМ, которые в этом случае выступают в качестве инструмента экспериментатора с моделью системы.

При **проектировании сложных систем** и их подсистем возникают многочисленные задачи, требующие оценки количественных и качественных закономерностей процессов функционирования таких систем, проведения структурного алгоритмического и параметрического их синтеза.

При **прогнозировании развития сложных систем** роль моделирования очень высока, так как это единственная возможность ответить на многочисленные вопросы о путях дальнейшего эффективного развития системы и выбора из них наиболее оптимального.

Для **сложных систем** необходимо учитывать следующие **особенности**:

- сложность структуры и стохастичность связей между элементами;
- неоднозначность алгоритмов поведения при различных условиях;
- большое количество параметров и переменных;
- неполноту и недетерминированность исходной информации;
- разнообразие и вероятностный характер воздействий внешней среды.

Ограниченность возможностей экспериментального исследования больших систем делает актуальной разработку **методики их моделирования**, которая позволила бы в соответствующей форме представить процессы функционирования систем, описание протекания этих процессов с помощью математических моделей, получение результатов экспериментов с моделями по оценке характеристики исследуемых объектов.

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

1. На стадии **макропроектирования** должна быть разработана обобщённая модель процесса функционирования сложной системы, позволяющая разработчику **получить ответы** на вопросы об эффективности различных стратегий управления объектом при его взаимодействии со внешней средой. Стадию внешнего проектирования можно разбить на **анализ** и **синтез**.

1.1. При **анализе** изучают объект управления, строят модель воздействий внешней среды, определяют критерии оценки эффективности, имеющиеся ресурсы, необходимые ограничения. Конечная цель стадии анализа – построение модели объекта управления для оценки его характеристик.

1.2. При **синтезе** решаются задачи выбора стратегии управления на основе модели объекта моделирования, т. е. сложной системы.

2. На стадии **микропроектирования** разрабатывают модели с целью создания эффективных подсистем. Используемые методы и средства моделирования

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

Таким образом, **проектирование** объекта – это процесс создания объекта и его модели; **моделирование** объекта – это способ оценки результата проектирования; моделирования без проектирования не существует.

### 1.1.2. История развития методов моделирования

В деятельности человека моделирование как способ отражения объектов реальной действительности или мышления используется с глубокой древности.

Исторически первыми моделями как заместителями некоторых объектов были языковые знаки, возникшие в ходе развития человечества и постепенно превратившиеся в **разговорный язык**. Слово было первой моделью реального объекта.

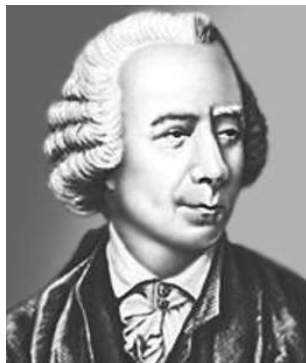
Следующим этапом развития моделирования можно считать возникновение **знаковых** числовых обозначений. Сведения о результатах счёта первоначально сохранялись в виде зарубок. Постепенное совершенствование этого метода привело к изображению чисел в виде **цифр** как системы знаков. Можно предположить, что именно зарубки были прототипом римских цифр.

По существу, моделирование как форма отражения действительности зарождается в античную эпоху одновременно с возникновением научного познания. Значительное развитие моделирование получило в **Древней Греции**. В V–III веках до нашей эры в Греции была создана геометрическая модель Солнечной системы. Греческий врач Гиппократ для изучения человеческого глаза воспользовался его физической моделью – глазом быка.

Однако в отчётливой форме моделирование начинает широко использоваться в эпоху Возрождения. Брунеллески, Микеланджело и другие итальянские архитекторы и скульпторы пользовались моделями проектируемых ими сооружений; в теоретических же работах Галилео Галилея и Леонардо да Винчи модели не только используются, но и выясняются пределы применимости метода.



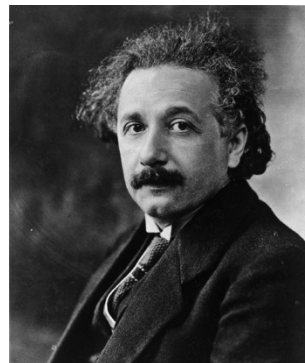
Исаак Ньютон  
(1642–1727)



Леонард Эйлер  
(1707–1783)



Джон Дальтон  
(1766–1844)



Альберт Эйнштейн  
(1879–1955)

Начало научного осмысления метода моделирования относится к XVII–XVIII вв. Классические примеры математических моделей:

- законы движения и всемирного тяготения Ньютона (XVII в.);
- полиномы Эйлера (XVIII в.);
- закон парциального давления Дальтона (XIX в.);
- теория относительности Эйнштейна (XX в.).

Исключительно большую методологическую роль сыграли в этом отношении работы Кельвина, Дж. Максвелла, Ф.А. Кекуле, А.М. Бутлерова и других физиков и химиков – именно эти науки стали, можно сказать, классическими «полигонами» методов моделирования. Появление же первых электронных вычислительных машин (Дж. Нейман, 1947) и формулирование основных принципов кибернетики (Н. Винер, 1948) привели к поистине универсальной значимости новых методов – как в абстрактных областях знания, так и в их приложениях. Моделирование ныне приобрело общенаучный характер и применяется в исследованиях живой и неживой природы, в науках о человеке и обществе.

Рука об руку с методами моделирования развивались и методы **проектирования систем**. В античные времена проектирование рассматривалось как «наука архитектора». Деятельность архитектора была связана не только с возведением зданий, но и с созданием строительных и военных машин.

Первые сооружения, которые были спроектированы и построены человеком, как ни странно, дошли до наших дней. Несомненно, это пирамиды в Гизе. Архитектура и проектирование Древнего Египта достигло своего расцвета при **фараоне Сезострисе**. Форма и пропорции пирамид – это не случайные величины, они геометрически выверены, т. е. можно утверждать, что пирамиды были первыми объектами, имеющими проектные данные.

Ещё около двух тысячелетий назад знаменитый представитель александрийской школы **Герон** создавал грузоподъемные и военные машины, турбины и даже простейшие автоматы для раздачи воды.



(I век до н. э.)

Описание системы знаний и принципов организации этой науки представлено в труде римского архитектора и механика **Марка Витрувия**, жившего 2 тысячи лет назад в эпоху Цезаря и Августа.

Трактат «Десять книг об архитектуре» (лат. *De architectura libri decem*) является единственной сохранившейся античной работой об архитектуре и – по свидетельству самого Витрувия – единственной на тот момент книгой об архитектуре на латыни.

Важнейшим этапом в инженерном деле стало применение масштабных чертежей. Этот способ развился в XVII веке и оказал силь-

нейшее влияние на дальнейшую историю проектирования. Благодаря ему появилась возможность разделить инженерный труд на собственно разработку идеи и её техническое воплощение.

Последующие столетия дали человечеству целый ряд гениальных сложных проектов в различных областях человеческой деятельности – в коммуникациях (телефон, телеграф), в транспорте (автомобили, поезда, самолеты), в энергетике (электричество, нефте- и газодобыча), сохранении статической и динамической информации (фотография, кино, телевидение) и т. д.

История проектирования автоматизированных информационных систем насчитывает не так много лет и связана прежде всего с появлением и развитием электронной вычислительной техники. Собственно, сами ЭВМ – также сложные системы, разработка которых потребовала консолидации значительных человеческих и материальных ресурсов в проектных командах.

Первые **информационные системы** появились в 1950-х годах. В эти годы они выполняли обработку счетов и расчёт зарплаты, а реализовывались на электромеханических бухгалтерских счётных машинах. Это приводило к некоторому сокращению затрат и времени на подготовку бумажных документов.

1960-е годы знаменуются изменением отношения к информационным системам. Информация, полученная из них, стала применяться для периодической отчётности по многим параметрам. Для этого организациям требовалось компьютерное оборудование широкого назначения, способное обслуживать множество функций, а не только обрабатывать счета и считать зарплату, как было ранее. Информационные системы становятся **автоматизированными**. Первое поколение автоматизированных ИС (1960–1970-е годы) строилось на базе вычислительных центров по принципу «одно предприятие – один центр обработки».

В 1970-х – начале 1980-х годов ИС начинают широко использоваться в качестве средства управленческого контроля, поддерживающего и ускоряющего процесс принятия решений. Второе поколение АИС (1970–1980-е годы) характеризуется переходом к **децентрализации** ИС. Информационные технологии проникают в отделы, службы предприятия. Появились пакеты и децентрализованные базы данных, стали внедряться двух-, трёхуровневые модели организации систем обработки данных.

Для третьего поколения АИС (1980–1990-е годы) характерен массовый переход к **распределённой сетевой обработке** на базе персональных компьютеров с объединением разрозненных рабочих мест в единую ИС.

Четвёртое поколение АИС (2000–2010-е годы) характеризуется **сочетанием централизованной обработки** на верхнем уровне с **распределённой обработкой** на нижнем. Наблюдается тенденция к возврату на крупных и средних предпри-



Труд Витрувия

ятях к использованию в ИС мощных ЭВМ в качестве центрального узла системы и дешёвых сетевых терминалов (рабочих станций).

*Современные ИС на предприятиях создаются на основе сетей ЭВМ*, новых технологий принятия управленческих решений, новых методов решения профессиональных задач конечных пользователей и т. д. Конечной целью как разработчиков, так и пользователей является создание целостных технологических систем, полностью охватывающих информационное производство со всеми основными и вспомогательными процессами на всех уровнях управления.

В функционально-прикладном аспекте подобные тенденции характеризуются расширением сфер применения новых инструментально-программных средств – сетевых средств, СУБД, экспертных систем и др. Существенно меняется как содержимое, так и состав функциональной структуры АИС предприятия на всех уровнях.

### **1.1.3. Роль методов моделирования в разработке и анализе систем военного назначения**

Роль методов моделирования и проектирования в разработке и испытаниях вооружения и военной техники также чрезвычайно высока.

Имитационное моделирование издавна используются в военном деле. **Военные игры** (манёвры, учения, командно-штабные учения и т. д.) проводятся для проигрывания (имитации) предстоящих операций и относятся к имитационному моделированию. Так, в РВСН при проведении командно-штабных военных игр широко используются **штабные математические модели** и другие, отражающие связь эффективности боевых действий с факторами, её определяющими.

В связи с бурным развитием вычислительной техники широкое распространение получили военные игры с использованием ЭВМ. Имитационное исследование, проводимое с использованием имитационных моделей, является основной формой системного анализа эффективности боевых действий.

События при имитации разворачиваются **во времени**, как правило, в том порядке, в каком они следуют в реальной системе, но в изменённой временной шкале. Действие случайных факторов учитывается с помощью специальных датчиков случайных чисел. В определённом месте процесс имитации может быть **приостановлен** для проведения, например, операционной военной игры, экспертного опроса или натурного эксперимента. Результаты игры, экспертизы или эксперимента могут быть использованы для продолжения имитации процесса на ЭВМ.

К настоящему времени наиболее распространено моделирование процессов вооружённой борьбы (боя, удара, сражения, операции и т. п.) с целью обоснования принимаемых решений в области **управления войсками и оружием** при подготовке и ведении боевых действий, строительстве вооружённых сил, разработке программ развития вооружений, оперативной подготовке штабов и т. д.

При изучении боевых действий видов ВС и родов войск метод моделирования является практически единственным методом познания и выработки военно-технических решений.

К настоящему времени создан большой класс моделей одиночных, групповых и массированных ударов группировок войск в различных формах боевого применения, предназначенных в основном для исследования эффективности боевых действий в широком диапазоне возможных условий обстановки.

Особое значение имеют задачи **планирования ударов** по противнику (в частности, задача целераспределения), решаемые только с использованием метода моделирования. Не менее важную роль играет моделирование при **выборе рационального состава** и структурно-функционального облика системы вооружения вооружённых сил.

Примером современного моделирующего комплекса военного назначения, объединяющего в себе разнообразные модели, реальную технику и участников виртуальных учений, является система SIMNET.

Напрашивалась идея о повышении эффективности использования моделей, снижающей стоимость их разработки и эксплуатации. Первой её реализацией занялся капитан военно-воздушных сил США Джек Торп, предложивший в 1978 году проект масштабируемой системы на базе симуляторов полёта для подготовки пилотов. Система представляла собой управляемую компьютером базу видеоматериалов, используемых в тогдашних имитаторах полёта, которая в параллельном режиме могла использоваться множеством обучаемых.

Успехи проектов Торпа и их однозначная экономическая выгода сподвигли в 1983 году военное исследовательское агентство DARPA к развитию этих разработок. Кроме команды Торпа к исследованиям были привлечены компании Delta Graphics и BBN Technologies.

Усилиями специалистов этих компаний к середине 1985 года была разработана концепция и прототип сети SIMNET – многопользовательской распределённой моделирующей системы, обеспечивающей отработку боевых ситуаций в реальном масштабе времени. В составе SIMNET в едином модельном пространстве работали симуляторы танков, самолётов и вертолётов. Эффективность систем распределённого моделирования, доказанная сетью SIMNET, стимулировала дальнейшее развитие этого направления имитационного моделирования.

Протокол DIS, разработанный преимущественно для военных моделирующих систем, потребовал существенной переработки. Её результатом стала архитектура, описывающая принципы организации любых систем распределённого моделирования. Её инвариантный характер отражается в названии HLA (High Level Architecture) – высокоуровневая архитектура (рис. 1.1).





Рис. 1.1. Архитектура HLA.  
Единый фундамент виртуальных полигонов

В основе идеологии HLA лежит принцип объединения множества объектов, участвующих в процессе распределённого моделирования, в динамически формируемую сущность, именуемую **федерация** (*federation*). Соответственно входящие в состав федерации объекты называются **федератами** (*federate*).

## 1.2. Общие вопросы теории моделирования

### 1.2.1. Понятие модели

Понятие модели в самом широком смысле можно сформулировать следующим образом.

**Модель** (фр. *modèle*, от лат. *modulus* – мера, аналог, образец) – это система, исследование которой служит средством для получения информации о другой системе, это упрощённое представление реального устройства и/или протекающих в нём процессов, явлений.

Собственно, процесс представления объекта исследований адекватной ему моделью и проведения экспериментов с моделью с целью получения информации об объекте, называется **моделированием**.

**Моделирование** – это построение и изучение моделей реально существующих объектов, процессов или явлений с целью получения объяснений этих явлений, а также для предсказания явлений, интересующих исследователя.

Таким образом, под моделированием понимают имитацию поведения некоторых существующих или предполагаемых систем либо отдельных аспектов

этого поведения с помощью модели, которая позволяет определять некоторые численные значения величин, характеризующих поведение системы путём измерения соответствующих величин на модели.

При этом предполагается, что элементы модели с определённой точки зрения являются *подобными элементам системы*.

В общем случае исследуемый объект-оригинал имеет некоторое множество параметров  $S_0$  и характеризуется определёнными свойствами. Количественной мерой свойств системы служит множество характеристик  $Y_0$ . Система проявляет свои свойства под влиянием внешних воздействий  $X$ .

Множество параметров системы и их значений отражает внутреннее содержание – структуру и принципы функционирования и существования. Характеристики системы – это в основном её внешние признаки, которые важны при взаимодействии с другими системами.

**Характеристики системы** находятся в функциональной зависимости от её **параметров**. Очевидно, что каждая характеристика системы  $y_0 \in Y_0$  определяется в основном или полностью ограниченным подмножеством параметров  $\{s_{ok}\} \in S_0$ . Остальные параметры не влияют или практически не влияют на значение данной характеристики системы.

**Модель** – это тоже система со своими множествами **параметров**  $S_m$  и **характеристик**  $Y_m$ . Оригинал и модель сходны по одним параметрам и различны по другим. Замещение одного объекта другим правомерно, если интересующие исследователя характеристики оригинала и модели определяются однотипными подмножествами параметров и связаны одинаковыми зависимостями с этими параметрами.

При одинаковых **внешних воздействиях**  $\{x_{on}\}$  на определённом временном интервале  $T$  для оригинала и модели характерны зависимости:

$$y_{ok} = f(\{s_{oi}\}, \{x_{on}\}, T);$$

$$y_{mk} = f(\{s_{mi}\}, \{x_{mn}\}, T).$$

Любая модель описывает реальный объект лишь с некоторой степенью приближения к действительности.

Перечислим основные свойства модели:

- 1) **конечность** – модель отображает оригинал лишь в конечном числе его отношений и, кроме того, ресурсы моделирования конечны;
- 2) **упрощённость** – модель отображает только существенные стороны объекта;
- 3) **приблизительность** – действительность отображается моделью грубо или приблизительно;
- 4) **адекватность** – модель успешно описывает моделируемую систему;
- 5) **информативность** – модель должна содержать достаточную информацию о системе в рамках гипотез, принятых при построении модели.

### 1.2.2. Классификация моделей

Модели объектов, модели деятельности целесообразно разделить по **направленности основных информационных потоков**, циркулирующих между субъектом и окружающим миром (рис. 1.2).

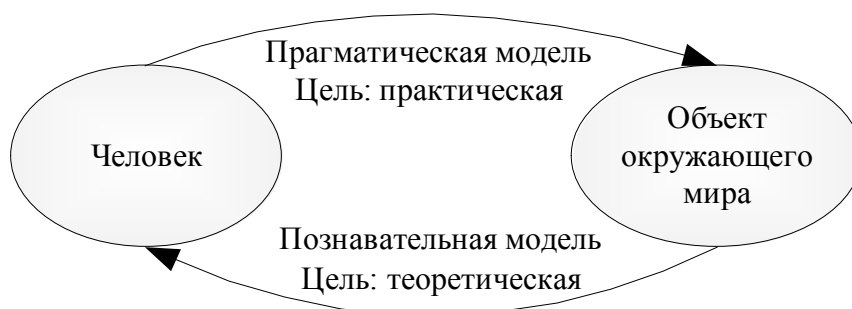
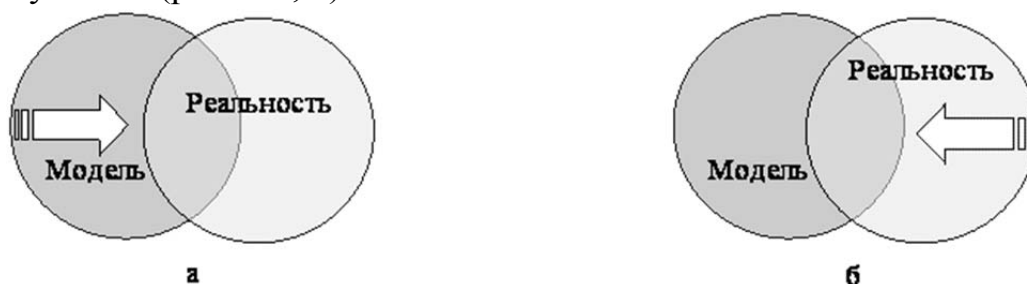


Рис. 1.2. Направленность информационных потоков между человеком и внешним миром

**Познавательная модель** – это форма организации и представления знаний, средство соединения новых знаний с уже имеющимися (рис. 1.3, а).

**Прагматическая модель** – это средство управления, средство организации практических действий, способ представления образцово правильных действий или их результата (рис. 1.3, б).



а) при установлении расхождений между моделью и реальностью модель «подгоняется» под реальность

б) при обнаружении расхождений между моделью и реальностью реальность «подгоняется» под модель

Рис. 1.3. Познавательная (а) и прагматическая (б) модели

**Познавательные модели отражают существующее, а прагматические – несуществующее, но желаемое и (возможно) осуществимое.**

Каждая модель характеризуется тремя признаками:

- **принадлежностью** к определённому классу задач (по классам задач);
- **указанием класса объектов** моделирования (по классам объектов);
- **способом реализации** (по форме представления и обработки информации).

Рассмотрим классификацию моделей (рис. 1.4). В основу классификации положена степень абстрагирования модели от оригинала. Предварительно, исходя из определения модели, все модели **по способу реализации** можно подразделить на две группы: **материальные** (реальные) и **абстрактные** (идеальные).

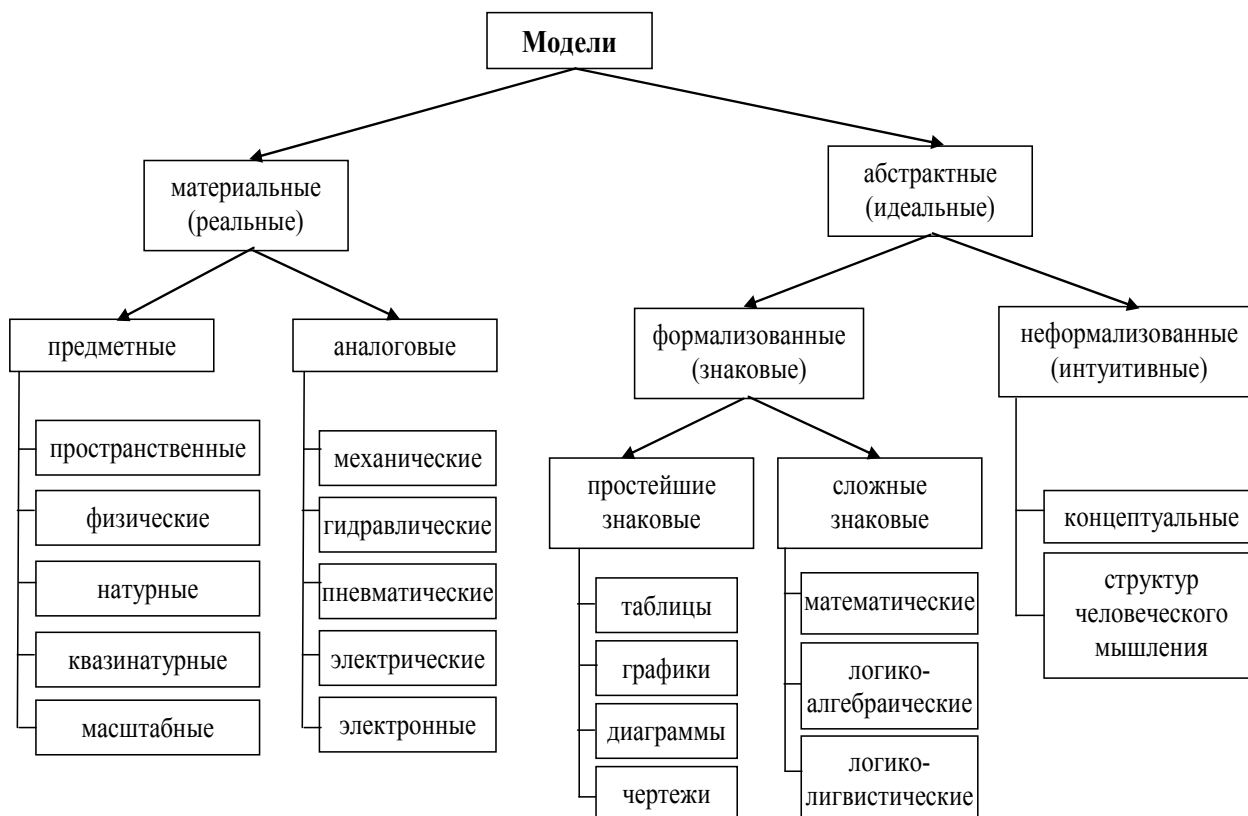


Рис. 1.4. Классификация моделей по форме представления и обработки информации

Соответственно, и развитие моделирования как одного из основных методов науки связано с двумя направлениями:

- материальным (реальным) моделированием;
- абстрактным (идеальным) моделированием.

**Материальные модели** образуются из совокупности материальных объектов той же или иной природы, что и оригинал (например, макет), и могут быть *предметными* и *аналоговыми*.

Когда модель и прототип имеют сходное геометрическое построение или одинаковую природу протекающих физических процессов, или и то, и другое, модель называют **предметной**.

Когда обеспечивается только процессуальное подобие (сходный с оригиналом процесс функционирования), но при различной физической природе оригинала и модели, говорят об **аналоговых моделях**. При этом процессы описываются одинаковыми математическими соотношениями.

Видами **предметного моделирования** являются:

- **пространственное моделирование** (обеспечение геометрического подобия, например, макеты, макетные архитектурные модели);
- **физическое моделирование** (обеспечение подобия физических свойств) – воспроизведение изучаемого процесса с сохранением его физической природы. Примером физического моделирования является построение и испытание моделей самолетов (в частности, продувка деталей) в аэродинамических трубах, исследование двигателей на стенде, различного вида тренажёры;

– **масштабное моделирование** – это моделирование систем той же физической природы, что и оригинал, но отличающихся от неё масштабами. При проектировании ВС может использоваться для анализа вариантов компоновочных решений по конструкции системы и её элементов.

– **натурное моделирование** (в качестве модели – сам прототип, при испытаниях, настройке, исследовании свойств). Натурные модели – это реальные исследуемые системы (макеты и опытные образцы). Они адекватны с системой-оригиналом, обеспечивают высокую точность и достоверность результатов моделирования. Процесс проектирования ВС завершается натурным испытанием опытных образцов.

– **квазинатурное моделирование** основано на совокупности натуральных и математических моделей. Этот вид моделей используется тогда, когда математическая модель части системы не является удовлетворительной или часть системы должна быть исследована во взаимодействии с другими частями, которых ещё не существует.

В качестве **аналоговых моделей** наиболее часто используются **электрические** и **электронные** системы, в которых сила тока и напряжения являются аналогами физических величин другой природы, а также **механические**, **гидравлические** и **пневматические** системы.

Особенностью аналоговых моделей является их **гибкость** и **простота** адаптации к изменению количественных параметров и характеристик объекта.

Аналоговые модели используют при исследовании средств вычислительной техники на уровне **логических элементов** и электрических цепей, а также на **системном уровне**, когда функционирование системы описывается, например, дифференциальными или алгебраическими уравнениями.

**Достоинством** материальных моделей является достаточно полное воспроизведение свойств оригинала, наглядность.

**Недостатками** материальных моделей являются:

- необходимость **переделки** модели при изменении параметров оригинала;
- **высокие затраты** на материальные модели;
- отсутствие возможности смоделировать **большие** и **сложные** системы.

Из-за дороговизны и громоздкости материальное моделирование имеет ограниченную область применения.

**Абстрактные модели** – это описание объекта исследований на некотором языке, т. е. компонентами такой модели являются не *физические элементы*, а *понятия* (например, словесные описания, чертежи, схемы, графики, таблицы, программы (алгоритмы), математические описания). В теории вычислительных систем используются преимущественно абстрактные модели. Абстрактные модели условно можно разбить на **формализованные** и **неформализованные**.

К **неформализованным моделям** в первую очередь относится **концептуальная модель** – это абстрактная модель, выявляющая причинно-следственные связи, присущие исследуемому объекту и существенные в рамках определённого исследования для получения требуемых результатов.

**Простейшими** формализованными моделями можно также считать таблицы, диаграммы, графики и чертежи.

Основным видом **формализованных моделей** является **математическая модель** – **абстрактная модель, представленная на языке математических отношений**. Она имеет форму функциональных зависимостей между параметрами, учитываемыми соответствующей концептуальной моделью. Эти зависимости **конкретизируют** причинно-следственные связи, выявленные в **концептуальной модели**, и характеризуют их **количественно**.

Кроме того, иногда выделяют **логико-алгебраические модели** (на основе соединения традиционной математики с математической логикой) и **логико-лингвистические модели** (сочетание математической логики и лингвистики).

### 1.2.3. Математические модели

**Математическое моделирование** – это способ исследования процессов путём изучения явлений, имеющих различное физическое содержание, но описываемых одинаковыми математическими зависимостями.

**Математическая модель** – это **формализованное описание системы с помощью абстрактного языка (в частности, с помощью математических соотношений), отражающее процесс функционирования системы**.

**Математическая модель** – это **совокупность соотношений (формул, уравнений, неравенств, логических условий, операторов и т. п.), определяющих характеристики состояний системы и выходные сигналы в зависимости от параметров системы, входных сигналов и начальных условий**.

Для составления математической модели используются любые математические средства – алгебраическое, дифференциальное и интегральное исчисление, теорию множеств, теорию алгоритмов и т. д.

Условно классификацию математических моделей можно представить в виде, представленном в табл. 1.1.

Математические модели можно разделить на **детерминированные** и **вероятностные** (стохастические). Первые устанавливают однозначное соответствие между параметрами и характеристиками модели, а вторые – между статистическими значениями этих величин. Выбор вида модели обусловливается степенью необходимости учёта случайных факторов.

Среди математических моделей по методу их исследования можно выделить **аналитические, имитационные и аналитико-имитационные модели**.

## Классификация математических моделей

Признаки классификации	Виды математических моделей
1. Принадлежность к иерархическому уровню	<ul style="list-style-type: none"> <li>▪ модели микроуровня;</li> <li>▪ модели макроуровня;</li> <li>▪ модели метауровня</li> </ul>
2. Характер взаимоотношений со средой	<ul style="list-style-type: none"> <li>▪ открытые (непрерывный обмен);</li> <li>▪ закрытые (слабая связь)</li> </ul>
3. Характер отображаемых свойств объекта	<ul style="list-style-type: none"> <li>▪ структурные;</li> <li>▪ функциональные (поведенческие)</li> </ul>
4. Способ представления свойств объекта	<ul style="list-style-type: none"> <li>▪ аналитические;</li> <li>▪ имитационные;</li> <li>▪ аналитико-имитационные</li> </ul>
5. Способ получения модели	<ul style="list-style-type: none"> <li>▪ теоретические;</li> <li>▪ эмпирические</li> </ul>
6. Причинная обусловленность	<ul style="list-style-type: none"> <li>▪ детерминированные;</li> <li>▪ вероятностные</li> </ul>
7. По отношению ко времени	<ul style="list-style-type: none"> <li>▪ динамические;</li> <li>▪ статические</li> </ul>
8. По типу уравнений	<ul style="list-style-type: none"> <li>▪ линейные;</li> <li>▪ нелинейные</li> </ul>
9. По множеству значений переменных	<ul style="list-style-type: none"> <li>▪ непрерывные;</li> <li>▪ дискретные;</li> <li>▪ дискретно-непрерывные</li> </ul>
10. По назначению	<ul style="list-style-type: none"> <li>▪ технические;</li> <li>▪ экономические;</li> <li>▪ социальные и т. д.</li> </ul>
11. По полноте	<ul style="list-style-type: none"> <li>▪ полные;</li> <li>▪ упрощённые</li> </ul>

**Аналитическая модель** описывает моделируемую систему в виде алгебраических, интегральных, дифференциальных, конечно-разностных и иных соотношений и логических условий.

Аналитические модели могут быть исследованы одним из трёх методов:

- **аналитически**, когда получают в общем виде явные зависимости для искомых характеристик;
- **численно**, когда нельзя найти общее решение для уравнений, а имеются только частные численные решения для конкретных начальных условий и количественных параметров модели;
- **качественно**, когда, не имея решения в явном виде можно, тем не менее, найти некоторые свойства решения (ограниченность, единственность, возвратность, эргодичность, устойчивость и т. д.).

Процесс аналитического моделирования может использовать четыре вида моделей: концептуальную, собственно аналитическую, алгоритмическую и программную (рис. 1.5). Концептуальная модель не является обязательным элементом аналитического моделирования, но может вводиться для выяснения целесообразности использования того или иного математического аппарата для построения аналитической модели.

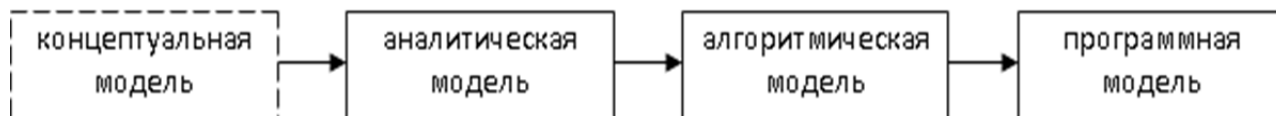


Рис. 1.5. Схема аналитического моделирования

Алгоритм численных исследований и программная модель могут опираться на использование пакетов прикладных программ.

**Достоинство:** возможность получения на её основе фундаментальных результатов, которые могут быть распространены на различные случаи использования системы и на другие системы данного класса.

**Недостаток:** существенная идеализация системы в целом и её элементов.

**Имитационная модель** — это совокупность описания системы и внешних воздействий, алгоритмов функционирования системы или правил изменения состояния системы под влиянием внешних и внутренних возмущений. Эти алгоритмы и правила позволяют имитировать процесс функционирования системы и производить измерения интересующих характеристик. Имитационные модели могут быть созданы для более широкого класса объектов и процессов, чем аналитические и численные модели.

В отличие от аналитического моделирования, имитационное моделирование требует обязательного использования ЭВМ (рис. 1.6). Также обязательна концептуальная модель, которая заменяет аналитическую модель и алгоритм численных исследований. Так как для нахождения объективных характеристик требуется многократный прогон отдельных реализаций и дальнейшая статистическая обработка полученных статистических данных, исследование сложных систем с помощью имитационных моделей называют **статистическим моделированием**.



Рис. 1.6. Схема имитационного моделирования



**Достоинства:** возможность отражения адекватным образом различных свойств элементов системы.

**Недостатки:** принципиальная необходимость осуществления многократных экспериментов, состоящих в имитации процессов в системе и требующих значительного машинного времени.

Для преодоления недостатков аналитического и имитационного моделирования используют **аналитико-имитационное моделирование** (рис. 1.7).

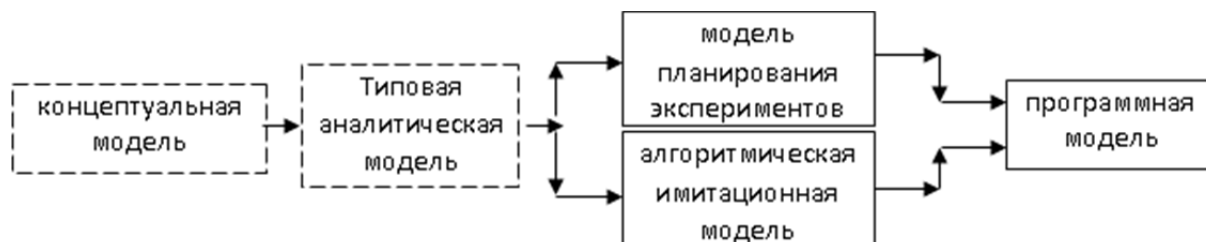


Рис. 1.7. Схема аналитико-имитационного моделирования

**Поведенческие** (функциональные) модели описывают объект как бы снаружи, внутренняя структура объекта не раскрывается. Это модели типа **чёрный ящик**.

**Структурные** модели отражают внутреннее строение объекта, его структуру. Структурные модели кажутся более привлекательными, чем поведенческие, потому что на небольшом наборе базовых элементов (и соответствующих им поведенческих моделей) можно без особых хлопот моделировать весьма широкий класс цифровых узлов и устройств. Кроме того, при построении структурных моделей совсем не обязательно располагать информацией о поведении моделируемого объекта. Нужна лишь его структура, а детали поведения объекта будут выявлены позднее, в ходе модельных экспериментов.

**К сожалению, структурные модели хороши, пока они захватывают не более двух, реже трёх уровней описания объекта.**

Попытка декомпонзировать сложный объект до уровня элементов или до уровня компонентов (транзисторов, резисторов и прочих радиодеталей) приводит к лавинообразному росту числа структурных примитивов, и риску «утонуть в деталях». Достаточно вспомнить, что **микропроцессор Pentium** содержит около трёх миллионов транзисторов, отчего пропадает всякое желание моделировать его на **транзисторном уровне**.

#### 1.2.4. Основные этапы моделирования

Для моделирования необходимо создать модель и провести её исследование. Перед созданием модели требуется конкретизировать цели моделирования. После исследования надо выполнить анализ результатов моделирования. Процесс создания модели проходит несколько стадий. Он начинается с изучения системы  $S_0$  и внешних воздействий  $X$  и завершается разработкой или выбором математической модели или программы. Некоторые математические модели могут быть исследованы без применения средств вычислительной техники.

**Моделирование ВС предполагает наличие следующих этапов [2]:**

- 1) формулирование цели моделирования;
- 2) разработка концептуальной модели;
- 3) подготовка исходных данных;
- 4) разработка математической модели;
- 5) выбор метода моделирования;
- 6) выбор средств моделирования;
- 7) разработка программной модели;
- 8) проверка адекватности и корректировка модели;
- 9) планирование машинных экспериментов;
- 10) моделирование на ВС;
- 11) анализ результатов моделирования.

Перечисленные этапы вытекают из общей методологии моделирования систем. При моделировании различных систем трудоёмкости одних и тех же этапов могут быть разными. В процессе моделирования конкретной системы могут иметь место некоторые изменения технологии. В частности, может быть заранее предопределён метод моделирования или выбрано конкретное средство моделирования.

Следует обратить внимание на первоочередную необходимость постановки, формулирования цели моделирования. Важность корректного выполнения этого этапа определяется тем, что все последующие этапы проводятся с ориентацией на определённую цель моделирования.

Для одной и той же системы  $S_0$  можно составить множество моделей  $\{S_m\}$ . Они будут отличаться степенью детализации и учёта тех или иных особенностей и режимов функционирования, ориентироваться на исследование определённого свойства или группы свойств системы. Поэтому перед разработкой модели следует сформулировать цели исследования.

При создании или модернизации любой системы встает задача определения её эффективности. Сделать это надо ещё до появления новой системы. Если может быть создано несколько вариантов системы, то из них надо выбрать наилучший. Решение этих задач и является основной целью моделирования. Однако понятие эффективности неоднозначно. Его надо конкретизировать до моделирования.

На создание и эксплуатацию технической системы расходуются **материальные, энергетические, трудовые и временные ресурсы**. В принципе можно рассчитать стоимость создания и эксплуатации системы.

Гораздо сложнее определить степень полезности системы (эффект) с учётом влияния на общество, человека и окружающую среду в ближайшем и отдалённом будущем. Часто не представляется возможным оценить в определённых

единицах общий эффект системы. В этом случае пользуются понятием технико-экономической эффективности, которое учитывает затраты и измеряемые характеристики системы:  $E = E(Y_0)$ .

Элементы множества характеристик  $y_{ok} \in Y_0$  являются частными показателями качества системы – это производительность, надёжность, стоимость, масса, габаритные размеры.

Если известны обратные зависимости, то **задача проектирования** тривиальна – можно вычислить параметры. В противном случае используется либо **однокритериальная**, либо **многокритериальная** оценки:

1. Ограничиваются оценкой эффективности по одному частному показателю, а по остальным накладывают ограничения.

2. Неизвестный вид функции искусственно представляется в форме обобщённого или интегрального критерия, который связывает простой зависимостью показатель эффективности со всеми учитываемыми характеристиками системы.

Определение цели моделирования заключается в первую очередь в выявлении вида критерия эффективности  $E$  исследуемой системы, а это предусматривает задание конечного множества характеристик  $\{y_i\}$ , их весовых коэффициентов и допустимых пределов изменений.

Ответ на эти вопросы может быть дан только в результате анализа метасистемы, в которую исследуемая система входит в качестве элемента, или, другими словами, в результате анализа взаимодействия системы со внешней средой.

В дальнейшем из множества характеристик  $\{y_i\}$  следует исключить те, которые могут быть определены без моделирования. В результате остается только подмножество  $\{y_{ok}\} \subset Y_0$ .

По результатам анализа метасистемы должны быть конкретизированы внешние воздействия  $\{x_{on}\} \subset X$ , под влиянием которых или при наличии которых функционирует система  $S_0$ , и исследуемый временной интервал  $T$  работы системы.

Под **разработкой математической модели** подразумевается создание полностью формализованного описания динамики функционирования системы.

Проверка адекватности указана в качестве одного из этапов моделирования. Однако на адекватность модели оказывает влияние качество выполнения практически всех этапов. Поэтому проверка адекватности должна проводиться в том или ином виде начиная от разработки концептуальной модели и кончая анализом результатов моделирования.

Результаты машинного моделирования должны быть проанализированы с целью проверки их достоверности и выработки рекомендаций о способах повышения качества исследуемой системы. На всех этапах моделирования нужно обращать внимание на документирование принимаемых решений, допущений, ограничений и выводов.

Если целью моделирования является не просто фиксация свойств системы, например, анализ её работоспособности, но её оптимизация, то перед моделированием следует выявить те параметры, которые исследователь может изменять.

Часто **целью моделирования** является определение зависимости одной или нескольких характеристик от тех или иных параметров системы.

### 1.3. Особенности технологии моделирования

#### 1.3.1. Построение концептуальной модели

В разработке модели условно выделяют четыре этапа описания: концептуальный, математический, алгоритмический и программный.

**Концептуальная (содержательная) модель** – это абстрактная модель, определяющая состав и структуру системы  $S_0$ , свойства элементов и причинно-следственные связи, присущие системе и существенные для достижения цели моделирования.

**Выделяют следующие шаги разработки концептуальной модели [2]:**

- 1) определение и ориентация;
- 2) стратификация;
- 3) детализация;
- 4) локализация;
- 5) структуризация и управление;
- 6) выделение процессов (или отражение состояний).

Раскроем эти понятия.

#### **Определение и ориентация**

*В словесной форме* приводятся сведения о **природе** и **параметрах** элементарных явлений исследуемой системы, о виде и степени взаимодействия между ними, о месте и значении каждого элементарного явления в общем процессе функционирования системы.

Модель ориентируется на выявление определённых свойств системы в соответствии с целями моделирования.

Для этого исследователь делает **мысленный срез системы** в плоскости той метасистемы, в качестве элемента которой представляет интерес система  $S_0$  (выполняет *M*-ориентацию). Затем выявляются основные признаки ориентированной модели.

Основная проблема при создании модели – нахождение компромисса между простотой модели и её адекватностью с исследуемой системой. При создании ориентированной модели уточняются множества полезных и возмущающих внешних воздействий.

## Стратификация

Следующий шаг – выбор уровня детализации модели. Модель представляется в виде совокупности частей (подсистем). Каждая часть системы также состоит из совокупности элементов. С учётом этого проблема уровня детализации может быть разрешена путём построения иерархической последовательности моделей.

Система представляется семейством моделей, каждая из которых отображает её поведение на различных уровнях детализации. На каждом уровне существуют характерные особенности системы, переменные, принципы и зависимости, с помощью которых описывается поведение системы.

Уровни детализации называются *стратами*, а процесс выделения уровней – *стратификацией*. Обычно в модель включаются элементы одной страты. Но иногда представляет интерес построение модели из элементов разных страт.

В модель должны войти все те параметры системы  $S_{ok}$ , которые допускают варьирование в процессе моделирования и обеспечивают определение интересующих характеристик  $Y_{ok}$  при конкретных внешних воздействиях на заданном интервале функционирования системы. Остальные параметры нужно исключить из модели.

## Детализация

Детализация системы должна производиться до такого уровня, чтобы для каждого элемента были известны или могли быть получены зависимости параметров выходных воздействий элемента, существенных для функционирования системы и определения её выходных характеристик, от параметров воздействий, которые являются входными для этого элемента.

Модель большой размерности с большим числом параметров следует упростить с помощью изоморфных преобразований без снижения адекватности, в том числе путём декомпозиции системы на подсистемы, интеграции элементарных операций и интеграции элементов, исключения или усечения второстепенных процессов.

## Локализация

Локализация осуществляется путём представления внешней среды в виде генераторов внешних воздействий, включаемых в состав модели в качестве элементов (рис. 1.8). Различают *генераторы рабочей нагрузки* (исходных объектов преобразования), *генераторы дополнительных обеспечивающих объектов*, *генераторы управляющих и возмущающих воздействий*.

Результаты функционирования системы поступают на приёмники выходных воздействий, которые обычно в состав модели не включаются.

К результатам относят основные продукты преобразования, побочные продукты преобразования, информацию о состоянии системы и управляющие воздействия на другие системы.

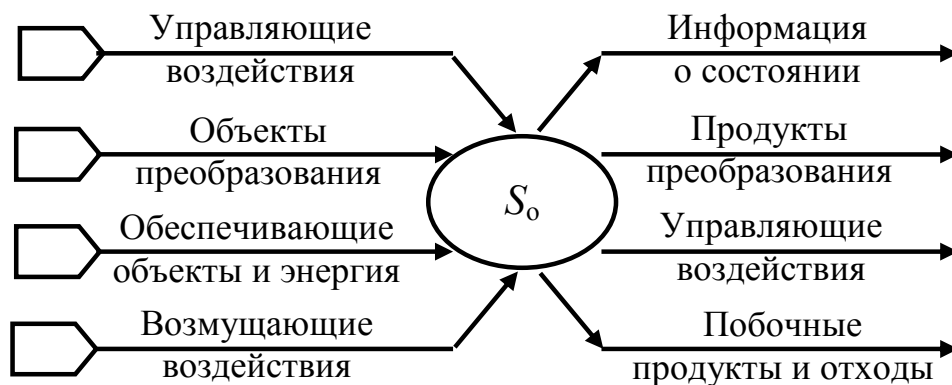


Рис. 1.8. Локализованная модель системы

## Структуризация

Построение структуры модели завершается указанием связей между элементами. Различают вещественные и информационные связи.

*Вещественные* – отражают возможные пути перемещения продукта преобразования (например, программы, задания) от одного элемента к другому.

*Информационные* связи передают управляющие воздействия и информацию о состоянии. В простых моделях информационные связи могут отсутствовать.

В самой структуре заложен *принцип структурного управления* (логические элементы, АВМ). В более сложных моделях имеются и управляющие, и информационные связи. Они работают в соответствии с *программным принципом управления*.

## Выделение процессов

Всё вышеперечисленное отражает статику системы – состав, структуру. Модель дополняется также описанием работы системы.

Функционирование системы заключается в выполнении технологического процесса преобразования информации (вещества, энергии). В сложных системах – сразу несколько технологических процессов.

*Технологический процесс* – это определённая последовательность отдельных элементарных операций. Часть операций может выполняться параллельно разными элементами.

Технологический процесс задаётся *алгоритмом*. Совокупность алгоритмов управления процессами  $A_0$  совместно с параметрами входных воздействий  $X_0$  и элементов  $S_0$  отражают *динамику функционирования системы*.

Описание динамики работы системы определяет множество характеристик системы  $Y$ :

$$Y = \Phi(X, S, A, T),$$

где  $\Phi$  – множество операторов вычисления выходных характеристик;

$X$  – параметры входных воздействий;

$S$  – параметры элементов;

$A$  – алгоритм управления;

$T$  – модельное время.

## Отражение состояний

Для систем со *структурным принципом управления* выходные характеристики описываются по-другому. Для каждого элемента выбирается один или несколько параметров, изменяющихся в ходе работы и отражающих состояние элемента в текущий момент времени  $z(t)$ .

Множество таких параметров отражает состояние системы  $Z(t)$ . Функционирование системы может быть представлено в виде последовательной смены состояний. Множество возможных состояний системы  $\{Z\}$  называют пространством состояний. Текущее состояние системы отражается в виде координаты точки в  $n$ -мерном пространстве состояний, а вся реализация процесса функционирования системы за время  $T$  – в виде некоторой траектории.

Если известно начальное состояние  $Z_0 = Z(t_0)$ , то состояние системы в момент  $t$  при входных воздействиях  $X$ , множестве состояний  $Z$ , начальном состоянии  $Z_0$  в любой момент времени определяется как

$$Z(t) = H(X, Z_0, Z, t).$$

Выходные характеристики определяют по формуле

$$Y = G(Z, T).$$

### 1.3.2. Подготовка исходных данных

Подготовка исходных данных – самый трудоёмкий и ответственный этап моделирования, так как от точности и полноты исходных данных зависит достоверность результатов моделирования.

Данный этап может включать такие шаги как сбор фактических данных, подбор законов распределений для случайных величин, аппроксимация функций, выдвижение гипотез, классификация исходных данных и т. д.

**Сбор фактических данных.** Достоверность результатов моделирования однозначно зависит от точности и полноты исходных данных. Сбор ИД может осуществляться по части существенных параметров параллельно с разработкой концептуальной модели. Затем определяются остальные параметры.

При сборе исходных данных возникает ряд проблем [19, 22, 23]. Информация, полученная при сборе и обработке фактических данных, является основой для определения конкретных значений параметров для модели. Достоверность результатов моделирования однозначно зависит от точности и полноты исходных данных. Большинство учитываемых в модели параметров по своей природе являются случайными. В целях упрощения модели их часто заменяют детерминированными средними величинами. Это допустимо, когда разброс случайных параметров небольшой или для достижения целей моделирования достаточно расчёта модели по средним значениям. Однако наличие случайных отклонений параметров системы приводит не только к *случайным отклонениям результи-*

рующих характеристик системы от среднего, но и к изменению самого этого среднего. Значит, при подмене в расчётах случайных параметров детерминированными результаты расчёта подвергаются не только рассеиванию, но и смещению относительно истинных значений, что приводит к возрастанию погрешности результатов анализа функционирования объекта.

Процесс сбора информации о системе может зависеть от множества ограничений – времени, возможностей, сложности сбора. Поэтому по количеству отсутствующей информации об исследуемом объекте, внешних условиях и т. д. различают следующие **уровни неопределённости** [19, 22, 23, 54]:

1. *Неизвестность* – соответствует начальной стадии изучения объекта и характеризует состояние практически полного отсутствия информации.

2. *Недостоверность* – соответствует первым этапам сбора информации, когда процесс сбора временно приостановлен или не хватает для этого ресурсов (возможностей). При этом не исключается возможность дальнейшего результативного изучения объекта.

3. *Неоднозначность* – ситуация, когда даже при наличии всей требуемой достоверной информации полностью определённое описание не получено и не может быть получено, поскольку неопределённость свойственна самой сущности рассматриваемой ситуации.

Как правило, первый уровень неопределённости сохраняется недолго, и основные проблемы с ИД возникают на втором уровне. **По характеру недостоверности ИД** выделим следующие её виды:

- *неполнота* ИД, когда собраны ещё не все возможные сведения, характеризующие исследуемую систему;
- *недостаточность* ИД, когда имеется ещё не вся необходимая информация для того, чтобы делать уверенные выводы;
- *недоопределённость* ИД, когда заданы (получены) не точные значения (описания) параметров (элементов) системы, а лишь множества (диапазоны) их возможных значений;
- *неадекватность* ИД, возникающая при замещающем описании параметров (элементов) системы по аналогии с существующими системами, либо при намеренном искажении информации с целью упрощения модели системы.

В процессе изучения системы уменьшается степень недостоверности исходных данных. Дальнейший сбор, обработка и интерпретация ИД о системе приводит либо к *полной определённости* (однозначности), либо к *неоднозначности*, когда вся возможная информация о системе уже собрана, но нельзя получить полностью определённое её описание.

Упрощение модели исследуемой системы (например, замена случайных или нечётких величин постоянными) в силу сложности модели или отсутствия достоверной информации ведёт к потере части имеющейся информации и увели-



чению погрешности получаемых результатов, поэтому необходим разумный компромисс между простотой модели и её адекватностью исследуемой системе.

Виды неопределённости можно классифицировать и **по источникам (факторам) возможной неопределённости** (точнее, неоднозначности, которая может иметь место на любом уровне неопределённости):

- физическая неопределённость, обусловленная внешней по отношению к системе средой;
- субъективная неопределённость, обусловленная субъективным восприятием человеком окружающего мира;
- логическая неопределённость, обусловленная спецификой человеческих знаний, трудно поддающихся однозначному толкованию, структуризации и алгоритмизации.

При этом два последних вида могут относиться как к знаниям конкретного человека, так и к знаниям, накопленным человечеством в искусственных носителях информации.

Наиболее существенной по своему влиянию на объект исследования является **внешняя среда** – основной источник неопределённости, под которой понимается совокупность элементов окружающего систему мира, не входящих в её состав, но оказывающих на неё то или иное воздействие. Кроме того, к среде часто относят недостаточно изученные элементы и факторы, включая и те, которые имеют внутрисистемный характер.

Неопределённость, связанная с трудностью выделения системы из внешней среды, проявляется в неструктурируемости (неформализуемости) системы, т. е. в невозможности построения соответствующей математической модели, и относится к субъективной неопределённости. Широкая подгруппа лингвистической неопределённости, обусловленная нечёткостью, неоднозначностью слов и фраз естественного языка, также относится к последней.

Необходимо отметить ещё некоторые обстоятельства, не учтённые в классификации. **Во-первых**, в чистом виде тот или иной вид неопределённости, как правило, не проявляется. Так, учёт физической неопределённости может усложниться появлением лингвистической неопределённости (нечёткости) или неточности в описании вероятностных распределений. Поливероятностная ситуация возникает тогда, когда об исходном распределении имеются неполные сведения (несколько правдоподобных гипотез о распределении, система его ограничений и др.). Таким образом, различные виды неопределённости могут *накладываться* один на другой.

**Во-вторых**, классификация не касается вопроса, какие элементы задачи исследования имеют неопределённое описание: постановка задачи, цель исследования, критерии (условия) оценивания (выбора), состояния среды (системы), ограничения (требования к системе), множество альтернатив и т. д.

Например, неопределённость описания целей отражается в многокритериальности оценивания (выбора) альтернатив и может иметь случайный или нечёткий характер. Неопределённость описания среды может проявляться в виде и физической, и лингвистической неопределённости.

**Подбор законов распределений.** Для случайных параметров организуется сбор статистики и её обработка. Затем на основе обработанной статистики подбирается теоретический закон распределения. Это даёт возможность создать аналитическую модель или упростить процесс имитационного моделирования.

Подбор закона распределения при этом состоит в следующем. По численным значениям строится гистограмма относительных частот событий (эмпирическая плотность распределения). Затем она аппроксимируется плавной кривой. Кривая сравнивается с разными законами распределений. При наилучшем совпадении выбирается один из законов. По эмпирическим значениям вычисляются параметры распределения. Затем оценивают степень совпадения распределений по одному из критериев согласия (Пирсона, Колмогорова, Смирнова, Фишера, Стьюдента).

**Аппроксимация функций.** Для каждого элемента системы существует связь между параметрами входных воздействий и выходными характеристиками. Если вид функции неочевиден, вносится гипотеза о характере функциональной зависимости, и функция аппроксимируется определённым математическим уравнением. Поиск математических зависимостей между двумя и более переменными по собранным данным осуществляется методами регрессионного, корреляционного и дисперсионного анализа.

Для описания элемента системы исследователем задаётся вид уравнения. Константы выбранного уравнения вычисляются методами регрессионного анализа, чтобы обеспечить наилучшее приближение кривой к экспериментальным данным. Корреляционный анализ помогает ответить на вопрос, насколько точно согласуется выбранная зависимость с опытными данными.

**Выдвижение гипотез.** Для новых элементов или условий функционирования может отсутствовать возможность сбора фактических данных. Для них выдвигаются гипотезы об их возможных значениях экспертами, лучше от группы специалистов.

**Классификация исходных данных.** На заключительном шаге сбора и обработки данных проводится классификация полученных исходных данных на внешние/внутренние, постоянные/переменные, непрерывные/дискретные, линейные/нелинейные, стационарные/нестационарные, детерминированные/стохастические и т. д. Это позволит в дальнейшем существенно упростить разработку и отладку математической модели системы.

### 1.3.3. Разработка модели, выбор методов и средств моделирования

#### Разработка математической модели

На основе концептуальной модели и исходных данных строится *математическая модель* с целью:

- формализованного описания структуры и процесса функционирования системы;
- представления процесса функционирования системы в доступном для аналитического исследования виде.

Единой методики создания математической модели нет ввиду разнообразия классов систем. Системы могут быть статическими или динамическими, со структурным или программным управлением, с постоянной или переменной структурой, с жёстким или гибким программным управлением. Кроме того, по характеру воздействий и внутренних состояний системы подразделяются на непрерывные и дискретные, линейные и нелинейные, стационарные и нестационарные, детерминированные и стохастические. Поэтому может быть получено и такое же многообразие моделей. Рассмотрим кратко некоторые из них.

Наиболее общими формализованными моделями представления функционирования ВС служат *агрегативные системы*, с помощью которых можно построить *непрерывные и дискретные, детерминированные и стохастические* модели. В основе такого представления лежит понятие *агрегата* как абстрактной схемы функционирования сложной системы, с помощью которого моделируются функциональные элементы ВС.

Дальнейшая конкретизация структуры пространств состояний, входных и выходных воздействий, операторов преобразования входных переменных в выходные и текущих состояний в последующие приводит к *кусочно-линейным агрегатам* и к более специализированным моделям, подчинённым конкретным целям моделирования: моделям теории расписаний, автоматным моделям, моделям типа сетей Петри, стохастическим сетевым моделям, моделям массового обслуживания и т. д.

Применение *моделей теории расписаний* целесообразно при наличии детерминированных потоков задач и времён их решения для оптимизации планирования работ в АС, маршрутизации задач на сетевой структуре АС, коррекции рассчитанных расписаний. Случайный характер процессов поступления и обслуживания задач в АС ограничивает возможности применения детерминированных моделей теории расписаний.

Под *системой массового обслуживания* будем понимать сложную систему, в которой выполняется последовательность операций по обслуживанию потоков определённого рода требований (заявок) путём предоставления им на какое-то время некоторых ресурсов. Основными элементами модели обслужива-

ния являются обслуживающие приборы, очереди заявок на обслуживание и источники заявок.

Будем полагать, что модель обслуживания задана, если известна зависимость  $F$  выходных характеристик системы  $Y$  от входных данных  $X$ :  $Y = F(X)$ . Часто такая зависимость известна в неявном виде – в виде уравнений  $\Phi(Y, X) = 0$ , которые могут быть неразрешимыми аналитически в замкнутом виде.

Обычно входными данными могут быть характеристики входного потока заявок и элементов модели, производящих обслуживание, а в качестве выходных характеристик выступают характеристики выходного потока, времени ожидания обслуживания, времени пребывания заявок в системе и т. д.

**Обслуживающие приборы** моделируют различные ресурсы, запрашиваемые заявками, которыми могут выступать различные технические устройства ВС: процессоры, память, каналы ввода-вывода и др. Один или несколько приборов в совокупности с очередью к ним образуют **узел обслуживания**.

Длительность обслуживания определяется отношением трудоёмкости заявки (количества работы по обслуживанию одного сообщения) к быстродействию прибора. Быстродействие выражается количеством работы, выполняемой прибором в единицу времени.

**Очереди заявок** возникают при задержке обслуживания. В качестве накопителя очереди выступают запоминающие устройства ЭВМ, поэтому при практических расчётах ёмкость накопителя можно считать неограниченной.

Порядок постановки заявок в очередь, как и порядок выбора заявок из очереди на обслуживание, определяется **дисциплиной обслуживания**.

**Источники заявок** генерируют **конечные** или **бесконечные** случайные потоки заявок. Заявки могут быть **однородными** (одного типа) или **неоднородными** (разных типов). Поток заявок поступает на вход узла обслуживания.

Моделями, наиболее полно и адекватно отражающими особенности процесса функционирования ВС и позволяющими получать основные характеристики его производительности, являются **сети массового обслуживания** (СеМО), или стохастические сетевые модели. Они строятся на основе моделей массового обслуживания и примыкающих к ним по математическому аппарату моделей теории случайных процессов (марковских и полумарковских процессов), теории графов, теории надёжности и теории восстановления, статистических моделей и др.

СеМО используются для описания систем с дискретными множествами состояний, входных и выходных воздействий, функционирующих в непрерывном времени. Стохастическая сетевая модель состоит из нескольких узлов, в ней дополнительно известны характеристики взаимосвязей между узлами, определяющие возможные пути следования заявок. Источниками заявок для узлов могут быть как другие узлы сети, так и внешние источники. Сетевая модель с внешними источниками называется открытой.

**Непрерывные детерминированные системы.** Если в модели не учитывается воздействие случайных факторов, а операторы переходов и выходов непрерывны, то зависимости могут быть представлены в виде дифференциальных уравнений. При линейности систем операторы переходов и выходов однородны и аддитивны, тогда возможно аналитическое решение.

**Автоматные модели** применимы для анализа систем, состояния которых определены в дискретные моменты времени.

Классические модели теории автоматов используются при анализе и синтезе дискретных устройств, а также микропрограммных автоматов и алгоритмов программ.

**Сети Петри** используются, в основном, при алгоритмическом анализе различных свойств ВС для отображения динамики его функционирования и параллельно протекающих в ВС процессов, для выявления и разрешения конфликтных ситуаций между требованиями задач на ресурсы, для решения задач планирования и оперативного принятия решений.

### **Выбор методов моделирования**

Разработанная математическая модель может быть исследована различными методами – аналитическими, численными, качественными или имитационными.

С помощью **аналитических методов** анализа можно провести наиболее полное исследование модели. Это удаётся лишь для сравнительно простых систем. Применение аналитических методов для более сложных систем связано с большей по сравнению с другими методами степенью упрощения реальности и абстрагирования. Поэтому аналитические методы исследования используются обычно для первоначальной грубой оценки характеристик всей системы или отдельных её подсистем.

Аналитические методы можно разделить на **точные, приближённые и эвристические**. Точный метод имеет чёткое доказательство, что полученное решение является оптимальным. Если же уверены, что решение отклоняется от оптимального не более чем на определённую величину, и эта величина задаётся ДО начала решения, то это приближённый метод. Если же величина погрешности вычисляется (оценивается) ПОСЛЕ получения ответа, то это эвристический метод решения.

**Методы численного анализа** применяются для получения конечных результатов в числовой форме, когда невозможно получить аналитические зависимости характеристик от параметров в явном виде. Численные методы по сравнению с аналитическими позволяют решать значительно более широкий круг задач.

Если полученные уравнения не удаётся решить аналитическими или численными методами, то прибегают к **качественным методам**. С помощью этих методов можно оценивать отклонения значений различных характеристик систе-

мы при вариации параметров, выявлять условия, при которых исследуемый процесс обладает или не обладает требуемыми свойствами, проводить сравнительную оценку различных процессов [24].

**Имитационное моделирование** является наиболее универсальным методом исследования систем и количественной оценки характеристик их функционирования. При имитационном моделировании динамические процессы системы-оригинала подменяются процессами, имитируемыми в абстрактной модели. В процессе имитации фиксируют определённые события и состояния или измеряют выходные воздействия, по которым вычисляют характеристики качества функционирования системы.

Имитационное моделирование позволяет рассматривать процессы, происходящие в системе, практически на любом уровне детализации.

### **Выбор средств моделирования**

После выбора метода моделирования нужно выбрать технические и программные **средства моделирования**.

К техническим средствам моделирования относят:

- **универсальные вычислительные системы** для аналитического моделирования без предъявления особых требований, для имитационного моделирования – высокопроизводительные ВС с наличием большой ОП;
- **специальные технические средства** аналитического моделирования – аналоговые вычислительные машины, используемые для исследования непрерывных детерминированных систем;
- **специализированные вычислительные системы** – стохастические машины, машины имитационного моделирования, гибридные моделирующие комплексы, распределённые системы моделирования.

К программным средствам моделирования относят:

- процедурно-ориентированные алгоритмические языки;
- проблемно-ориентированные языки моделирования;
- автоматизированные системы моделирования.

### **1.3.4. Проверка адекватности и корректировка модели**

В сложных системах, к которым относятся и информационно-вычислительные системы, модель лишь частично отражает реальный объект (или процесс). Поэтому необходимо проводить проверку степени соответствия (или адекватности) модели и реального процесса.

Проверка адекватности модели состоит в анализе её соразмерности исследуемой системе, а также равнозначности системе.

**Адекватность нарушается** по следующим причинам:

- идеализация внешних условий и режимов функционирования;

- исключение тех или других параметров;
- пренебрежение некоторыми случайными факторами;
- отсутствие точных сведений о внешних воздействиях, определённых нюансах структуры системы;
- принятия аппроксимаций, интерполяций, предположений и гипотез.

Проверку можно проводить путём сравнения выходных характеристик модели  $y_m$  или предвиденного поведения модели с фактическими характеристиками объекта  $y_0$  при изменении значений внешних факторов  $\omega_s$ , а также (при возможности) параметров системы  $A$  в широком диапазоне колебаний.

В качестве меры адекватности модели служат либо отклонение некоторой характеристики  $y_0$  оригинала и  $y_m$  модели, либо некоторая вероятностная оценка:

а) абсолютное отклонение  $\Delta y = |y_0 - y_m| \leq \Delta y^{\text{доп}}$ ;

б) относительное отклонение  $\varepsilon_y = |(y_0 - y_m)/y_0| \leq \varepsilon_y^{\text{доп}}$ ;

в) вероятностная оценка  $P_\Delta: P(\Delta y < \Delta y^{\text{доп}}) \geq P_\Delta$  или  $P_\varepsilon: P(\varepsilon_y < \varepsilon_y^{\text{доп}}) \geq P_\varepsilon$ .

где  $\Delta y^{\text{доп}}$  и  $\varepsilon_y^{\text{доп}}$  – допустимые отклонения, которые задаются экспертом и определяют заданную степень адекватности модели.

При вероятностной оценке можно считать, что модель адекватна системе, если вероятность того, что отклонения  $\Delta y$  или  $\varepsilon_y$  не превышают предельной величины  $\Delta y^{\text{доп}}$  и  $\varepsilon_y^{\text{доп}}$  соответственно больше допустимой вероятности  $P_\Delta$  или  $P_\varepsilon$ .

Если величины  $\Delta y$  или  $\varepsilon_y$  превышают  $\Delta y^{\text{доп}}$  и  $\varepsilon_y^{\text{доп}}$ , то это свидетельствует о том, что упущены некоторые важные факторы и модель требует корректировки.

Практическое использование данного критерия адекватности зачастую невозможно по следующим причинам:

- 1) для проектируемых или модернизируемых систем отсутствует информация о значении характеристики  $y_0$ ;
- 2) система оценивается не по одной, а по множеству характеристик, у которых может быть разная величина отклонения;
- 3) характеристики могут быть случайными величинами и функциями, а часто и нестационарными;
- 4) отсутствует возможность априорного точного задания предельных отклонений  $\Delta y^{\text{доп}}$  или  $\varepsilon_y^{\text{доп}}$  и допустимых вероятностей  $P_\Delta$  или  $P_\varepsilon$ .

Несмотря на это, проверять адекватность необходимо, иначе по неверным результатам моделирования будут приняты неправильные решения. На практике оценка адекватности обычно проводится путём экспертного анализа разумности результатов моделирования. Можно выделить следующие виды проверок:

- 1) проверка моделей компонентов;
- 2) проверка модели внешних воздействий (аппроксимации и гипотезы необходимо оценить математическими методами);

- 3) проверка концептуальной модели функционирования системы;
- 4) проверка формализованной и математической модели;
- 5) проверка способов измерения и вычисления выходных характеристик;
- 6) проверка программной модели.

### **Корректировка модели**

Если по результатам проверки адекватности выявляется недопустимое расхождение модели и системы, возникает необходимость в **корректировке** или калибровке модели.

Возможны корректировки концептуальной модели, математической модели и, соответственно, метода решения. При этом выделяются следующие типы изменений: глобальные, локальные и параметрические.

**Глобальные:** в случае методических ошибок в концептуальной или математической моделях. В результате устранения – новая модель.

**Локальные:** уточнение параметров или алгоритмов, замена моделей компонентов системы и внешних воздействий.

**Параметрические:** изменение калибровочных параметров.

Стратегия корректировки модели должна быть направлена на первоочередное введение глобальных изменений, затем – локальных и параметрических изменений. Для выработки тактики параметрических изменений большое значение имеет анализ чувствительности модели к вариациям её параметров.

Корректировка может потребовать проведения дополнительных исследований на объекте, наборе необходимых данных, уточнения набора переменных и структуры модели.

Можно выделить такие варианты корректировок математической модели:

- расширение набора внешних факторов, управляющих переменных и выходных характеристик модели;
- переход от линейных зависимостей  $f(X, Y, A, \omega)$  к нелинейным или повышению степени нелинейности;
- расширение набора ограничений или их комбинаций.

Корректировка может повторяться многократно до тех пор, пока не будет достигнуто удовлетворительное соответствие между выходными характеристиками объекта и модели.

Завершается этап проверки адекватности и корректировки модели определением и фиксацией **области пригодности модели**. Под областью пригодности понимается множество условий, при соблюдении которых точность результатов моделирования находится в допустимых пределах.



### 1.3.5. Планирование экспериментов с моделью и анализ результатов моделирования

#### Планирование экспериментов

Планирование экспериментов с моделью можно разделить на **стратегическое** и **тактическое**.

При ограниченных вычислительных и временных ресурсах обычно не представляется возможным провести все эксперименты.

**Стратегическое планирование** состоит в необходимости выбора определённых сочетаний параметров и последовательности проведения экспериментов. Выявляются характеристики качества и параметры, с помощью которых предполагается управлять качеством функционирования системы.

Параметры в теории планирования экспериментов называют **факторами**. Выделяют возможные значения количественных параметров и варианты функциональных параметров.

Если число факторов велико, то используют факторный анализ для составления плана. Особое значение – при исследовании нестационарных систем.

**Тактическое планирование** – совокупность методов уменьшения длительности машинного эксперимента при обеспечении статистической достоверности результатов имитационного моделирования.

Существует три основных метода уменьшения ошибки:

- 1) достаточное увеличение периода моделирования;
- 2) начало сбора статистики не с начального момента, а по истечении некоторого времени;
- 3) инициализация модели не с нулевого, а специально заданного состояния, близкого к установившемуся.

При моделировании нестационарных систем нет установившегося режима, поэтому используют **метод повторных экспериментов**.

#### Анализ результатов моделирования

Анализ результатов моделирования включает обычно

- обработку измерений имитационного эксперимента;
- определение зависимостей характеристик от параметров системы;
- использование результатов моделирования.

**Обработка измерений** заключается в вычислении оценок математических ожиданий, дисперсий и корреляционных моментов выходных характеристик.

Оценки должны быть **состоятельными**, **несмещёнными** и **эффективными**.

Оценка называется **состоятельной**, если она сходится к искомому значению параметра при неограниченном увеличении числа опытов.

Оценка является **несмещённой**, если её математическое ожидание при любом конечном  $N$  равно истинному значению параметра.

Оценка *эффективна*, если среди всех возможных она обладает наименьшей дисперсией.

Для стохастических характеристик используется гистограмма относительных частот – эмпирическая плотность распределения.

При *определении зависимостей характеристик системы от параметров* используется корреляционный, дисперсионный или регрессионный анализ.

**Корреляционный анализ** предназначен для установления наличия связи между двумя и более величинами. Оценка – коэффициент корреляции.

**Дисперсионный анализ** – для установления относительного влияния различных факторов на значения выходных характеристик. Общая дисперсия разлагается на компоненты, соответствующие рассматриваемым факторам.

**Регрессионный анализ** используется для нахождения аналитической зависимости между характеристиками и факторами. При этом

1) выбирается вид соотношения между **зависимыми** и **независимыми** переменными;

2) по экспериментальным данным вычисляются **параметры** выбранной зависимости;

3) оценивается **качество** аппроксимации экспериментальных данных моделью.

Результаты моделирования используются для принятия решения о работоспособности системы, для выбора лучшего проектного варианта или для оптимизации системы.

Изложенные технологические аспекты моделирования рассмотрим в следующих подразделах более подробно на примерах моделирования автоматизированных систем, в частности, вычислительных систем.

## 1.4. Разработка и формализация моделей сложных систем

### 1.4.1. Автоматизированные системы как объекты моделирования

Основным объектом моделирования в дисциплине выступают различные автоматизированные системы, в том числе:

- автоматизированные информационные системы;
- автоматизированные системы обработки данных;
- автоматизированные системы управления;
- автоматизированные обучающие системы и т. д.

Ядром любой автоматизированной системы выступает **вычислительная система**, предназначенная для хранения и автоматизации обработки данных на основе алгоритмов. **Алгоритм** – последовательность операций, приводящих к решению поставленной задачи. Данные – это информация, представленная в формализованном виде и предназначенная для обработки или обработанная вычислительной системой.

Рассмотрим основные особенности ВС с позиций моделирования на уровне системы. Базой для построения модели служит **структурная организация системы**, которая, в свою очередь, воплощает принципы определённой **функциональной организации**.

**Функциональная организация (архитектура) вычислительной системы** – это множество абстрактных элементов, в терминах которых отражаются данные и алгоритмы, формы представления информации, процедуры взаимодействия с внешней средой и порождаемые алгоритмами процессы функционирования системы. Сюда относятся наборы символов, используемых для представления данных, способы их кодирования, машинные операции, адреса и команды, процедуры ввода и вывода данных и т. д.

В соответствии с неймановским принципом данные представляются в ВС дискретными элементами – словами, каждое из которых выражается в двоично-кодированной форме. Вычислительная система оперирует с данными трёх основных типов: числами, строками символов и логическими значениями.

Другим важным элементом функциональной организации являются команды. Команда служит для предписания операции, выполняемой техническими средствами ВС, и месторасположения данных, участвующих в операции. При разной функциональной организации команды могут иметь разный формат, которым определяется длина команды в битах, количество и расположение полей, используемых для кодирования типа операции, адресов данных и другой управляющей информации.

ВС имеет следующие основные функциональные составляющие: ввод, обработка, вывод, хранение и управление (рис. 1.9).



Рис. 1.9. Функциональная структура вычислительной системы

**Ввод данных** обеспечивает восприятие данных с материальных носителей внешней среды, преобразование формы представления данных и формирование их представления на внутренних носителях. **Обработка** приводит к преобразованию (изменению содержания) данных в новую информацию – результаты.

Функция **вывода данных** обеспечивает обратное преобразование по некоторому алгоритму вывода. Функция **хранения** необходима для исключения строгой синхронности и последовательности во вводе, обработке и выводе данных. Многообразие операций по вводу, хранению, обработке и выводу создаёт потребность в функции **управления**, обеспечивающей автоматическую работу системы. Исходным объектом преобразования для системы управления ВС служат программы.

По **режиму использования** вычислительные системы подразделяют на системы индивидуального и коллективного пользования. К первым можно отнести микрокалькуляторы, специализированные ВС, персональные компьютеры, планшеты и т. д. Ко вторым – универсальные или проблемно-ориентированные системы с достаточно мощными ресурсами.

Для ВС коллективного пользования принято выделять пакетный, интерактивный режим и режим реального времени. В системах управления реальными объектами и информационно-измерительных системах ВС должны обрабатывать данные в темпе протекания процессов, происходящих вне ВС, что называется **режимом реального времени**. В **интерактивном режиме** работы каждому пользователю предоставляется на определённое время индивидуальные средства ввода-вывода и возможность инициирования требований на обработку данных в произвольные моменты времени. В **пакетном режиме** работы пользователи предъявляют требования на обработку данных в виде задания, которое содержит сведения о программе обработки и данных.

Исходя из режима обработки данных, все ВС можно разбить согласно классификации, предложенной в 1966 году М. Флином, на четыре класса:

- одиночный поток команд, одиночный поток данных (ОКОД);
- одиночный поток команд, множественный поток данных ОКМД);
- множественный поток команд, одиночный поток данных (МКОД)
- множественный поток команд, множественный поток данных (МКМД).

Системы класса ОКОД относятся к фон-неймановским ВС, в которых команды исполняются последовательно, а отдельная команда имеет дело не более чем с одной операцией над данными. Программа обработки и данные предварительно заносятся в память. К этому типу относятся **однопроцессорные ВС**.

Системы класса ОКМД имеют команды, в том числе векторные, которые могут непосредственно запускать большое число одинаковых операций над различными данными – элементами вектора. Способ выполнения векторных операций не оговаривается, поэтому обработка элементов вектора может производиться либо процессорной матрицей, либо с помощью конвейера. В эту категорию попадают **векторные процессоры**, в которых наиболее успешно реализуется конвейерный принцип обработки, – это конвейерные ВС.

В системах класса МКОД единый поток данных проходит одновременно обработку многими командами на разных процессорах. Представителями этого класса ВС служат *машины потоков данных*.

Системы класса МКМД характеризуются одновременным исполнением многих различных потоков команд (программ), в которых команды каждого потока имеют дело с различными наборами данных. К этому классу относятся *мультипроцессорные мультипрограммные ВС*.

**Структурная организация вычислительной системы** – это состав элементов, конфигурация связей между ними и способы управления, обеспечивающие реализацию требуемой функциональной организации.

С позиций системотехнического моделирования целесообразно подразделить основные технические средства на следующие классы: *процессоры, терминалы, накопители, контроллеры и коммутаторы*.

**Процессор** – устройство, которое может выполнять множество операций над операндами, взятыми из множества данных, с целью получения новых данных. Процессор – операционный элемент. В каждый момент времени процессор реализует ограниченное число операций.

**Терминал** – устройство, обеспечивающее связь ВС с внешней средой. Назначение – преобразование формы и физической среды представления информации. Разделяют входные и выходные терминалы. Дополнительная функция – поиск на носителе места для чтения-записи данных.

**Накопитель** – устройство, предназначенное для хранения множества элементов данных и снабжения средствами селекции, обеспечивающими запись или чтение заданного элемента. Различают накопители непосредственного доступа (СОЗУ, ОЗУ, ПЗУ), накопители прямого доступа (на МД и МБ), накопители последовательного доступа (динамические ЗУ и НМЛ).

**Контроллер** – средство организации взаимосвязанного функционирования элементов системы. Принимает управляющие воздействия из внешней среды или других контроллеров, собирает сведения о состоянии элементов и процессов, анализирует эту информацию. Его функции реализуются программными, микропрограммными или аппаратными средствами.

**Коммутатор** – устройство для передачи данных или управляющей информации от одного устройства к другому. Для расширения возможностей интерфейсов по числу подсоединяемых устройств используются формирователи сигналов, ретрансляторы, расширители и адаптеры. К коммутаторам также относят коммутаторы, селекторы, концентраторы, мультиплексоры.

Ресурсы приведённых классов могут быть объединены конструктивно в одном устройстве.

**Рабочей нагрузкой** для ВС является поток заданий на обработку и хранение данных, распределённый во времени. Задание определяет программу обработки данных и необходимые ресурсы.

Программа полностью определяет последовательность использования ресурсов и функции, которые должны быть выполнены каждым ресурсом.

Вне зависимости от того, каким образом инициируется выполнение программы – представлением управляющей информации в виде задания или самим фактом появления запроса, сигнала или новых данных, требование на выполнение работы при моделировании принято называть *заявкой*. Заявка характеризуется моментом появления на входе системы, статусом по отношению к другим заявкам и параметрами, определяющими потребности в ресурсах на выполнение вычислительной работы. Совокупность заявок, распределённая во времени, называется *поток заявок*. Таким образом, рабочей нагрузкой ВС является поток заявок.

Поток заявок может быть *однородным* и *неоднородным*. В универсальных ВС коллективного пользования поток заявок зачастую отличается крайней неоднородностью. При моделировании неоднородный поток может быть представлен совокупностью однородных потоков. В каждый из однородных потоков объединяются заявки с идентичным статусом и параметрами.

Одним из основных параметров потока заявок является промежуток времени между моментами поступления двух соседних заявок. Этот промежуток может быть постоянной или переменной величиной. Совокупность промежутков может иметь периодически повторяющееся количество значений, задаваемых определённым правилом, или большое количество значений, дающее основание говорить, что это величина случайная. В общем случае поток заявок рассматривается как случайный процесс, характеризуемый функцией распределения периода поступления заявок. Среднее число заявок, поступающих в единицу времени, называется интенсивностью потока ( $\lambda$ ).

Поток заявок может быть *стационарным* или *нестационарным*. Это зависит от того, не изменяются или изменяются статистические характеристики потока во времени. Режим работы реальных ВС далеко не всегда является стационарным. Рабочая нагрузка на ВС зачастую изменяется в зависимости от времени суток, дней недели, месяцев квартала или года и от состояния управляемого объекта или технологического процесса для управляющих систем. Интенсивность может представляться в виде функции времени.

#### 1.4.2. Выбор уровня детализации и декомпозиция моделей

Процесс создания моделей, вероятно, не подлежит полной формализации из-за многообразия систем. На начальных стадиях разработки модели конкретизируются цели моделирования и выполняются ориентация, стратификация, детализация и локализация.

При разработке модели ключевой проблемой является правильный **выбор уровня детализации**. Для ВС используется представление в виде одного эле-

мента или производится дифференциация на отдельные функциональные устройства, блоки, узлы, логические элементы.

1. **Логический** уровень. В качестве систем выступают логические и другие элементы (с транзисторами, диодами, сопротивлениями, ёмкостями).

2. **Цифровой** уровень. Системы – цифровые устройства (регистры, сумматоры, дешифраторы), образующиеся путём соединения логических элементов.

3. **Функциональный** уровень. Системы – отдельные функциональные устройства – процессоры, контроллеры, накопители, терминалы и коммутаторы.

4. **Системный** уровень. В качестве системы рассматривается ВС, а её элементами являются функциональные устройства.

5. **Сетевой** уровень. В качестве системы – сеть ЭВМ. В качестве элементов – отдельные машины.

6. **Межсетевой** уровень. В качестве системы – совокупность сетей (глобальная сеть). В качестве элементов – отдельные (локальные) сети.

При моделировании на системном уровне под **заявкой** понимается требование на выполнение одной программы, одного цикла выполнения программы по обработке запроса пользователя или определённого количества исходных данных.

В соответствии с иерархическим подходом к построению моделей сложных систем исследуемую ВС можно рассматривать на различных уровнях иерархии. Так, можно предложить трёхуровневый подход.

**На первом уровне** – исследуемая ВС представляется узлом обслуживания, входящим в состав сетевой модели, другие узлы которой представляют собой ВС или средства других подразделений и организаций.

**На втором уровне** в качестве сетевой модели выступает сама ВС, узлами которого являются отдельные ЭВМ, входящие в состав ВС.

**На третьем, низшем уровне**, рассматриваются **сетевые модели** отдельных ЭВМ, узлами которых выступают процессор, каналы, внешняя память, устройства ввода-вывода и т. д. Такой подход позволяет проводить исследования на различных этапах разработки ВС и с различной степенью детализации, соответствующей полноте и точности имеющихся исходных данных.

При решении вопроса о конкретизации понятия заявки и выборе уровня детализации рассматривается возможность разработки **последовательности моделей**.

Вначале проектируется и используется для решения задачи *модель первого порядка* сложности, адекватная системе только в первом приближении.

На её основе строится *модель второго порядка* сложности, обладающая более глубоким уровнем детализации, большим числом составляющих и параметров. При необходимости строится *модель третьего порядка*. Так до тех пор, пока не будет получена модель, наиболее подходящая для достижения поставленных целей.

Так, например, при исследовании ВС целесообразно использовать последовательность моделей ВС:

- 1) детерминированные модели обслуживания;
- 2) стохастические модели обслуживания;
- 3) стохастические модели обслуживания с неопределённостью [19, 22];
- 4) динамические модели (нестационарные, с изменениями режимов и др.);
- 5) имитационные модели.

Применение данной последовательности может повторяться при переходе на более детализированное описание элементов ВС и рабочей нагрузки, учитывающее их неоднородность, приоритетность, разнотипность.

Первоначальное формирование исходных данных можно производить на основе существующих **статистических оценок** и анализа полученной **экспертной информации**. Так, сведения о трудоёмкости решаемых задач получают на основе анализа алгоритмов обработки информации в рамках существующей нагрузки. Анализ форм обмена между различными абонентами, их видов и частоты прохождения позволяет сформировать сведения о временных характеристиках рабочей нагрузки. На начальных этапах исследования обычно не учитывают разнотипность и приоритетность различных задач, а также сетевую структуру ВС. Отбор альтернативных вариантов построения ВС осуществляется на основе предварительного оценивания для сравнения при дальнейшем исследовании.

При **уточнении исходных данных** определяется количество типов заданий, исходя из разумной детализации наиболее существенных видов обрабатываемой информации и возможности слияния нескольких потоков информации в один с интегральными параметрами.

Одним из основных принципов разбиения информационных потоков по группам должно быть **примерное равенство директивных времён** окончания обслуживания заданий. Необходимо также учесть **однотипность маршрутов** передвижения по узлам сети, **одинаковую потребность** в использовании программных ресурсов при обработке данного вида информации, **принадлежность** к одним и тем же источникам запросов на обработку информации. При этом необходимо выдерживать **принцип соразмерности длительности и периодичности**: в модели длительности обслуживания заявок в узлах сети и периодичность поступления заявок разных потоков должны отличаться не более чем на один-два порядка. Этот принцип особенно важен для подбора параметров моделирования на основе статистических данных, поскольку стремление включить в модель заявки, которые поступают несоизмеримо реже по сравнению с другими, или желание учесть задачи, длительность реализации которых несоизмерима со временем обслуживания заявок других типов, заведомо приводит к неудаче при моделировании.



**Декомпозицией** называется разделение исследуемой системы на отдельные подсистемы (модели), проводимое с учётом их взаимного влияния друг на друга, с полным отражением при этом целостных свойств системы.

После решения вопроса об уровне детализации надо изучить возможность расчленения ВС на подсистемы и создания самостоятельных моделей отдельных подсистем. Например, при анализе вычислительной сети целесообразнее сначала создать и исследовать модели отдельных машин, а затем – модель всей системы.

Задача декомпозиции системы в общем случае не является тривиальной. Разделение системы проводится по функциональной обособленности подсистем или по минимуму функциональных связей. Имеет значение не только количество, но и направленность и мощность связей. Нельзя расчленить системы, имеющие как прямые, так и обратные связи. При декомпозиции не должна быть потеряна целостность системы. Применение декомпозиции позволяет распараллелить процесс моделирования.

**Агрегирование** – объединение конструктивно независимых элементов, выполняющих различные функции, в единую систему. В системах обработки данных агрегирование осуществляется по функциональному признаку.

#### 1.4.3. Упрощение моделей и подбор параметров

Важными этапами разработки и формализации систем являются **упрощение моделей** и **подбор параметров** модели.

Для **упрощения моделей** используются следующие приёмы:

- 1) слияние потоков;
- 2) разделение ресурсов;
- 3) соблюдение соразмерности длительностей;
- 4) усечение модели;
- 5) интеграция ресурсов и др.

**Слияние нескольких потоков заявок** в один поток с интегральными параметрами возможен после окончательной подготовки алгоритмов обработки данных, выделения технологических цепочек обслуживания заявок, обработки исходных данных и определения параметров. При этом важным критерием слияния является схожесть маршрутов, приоритетов и объёмов, а не содержания заявок.

**Разделение ресурсов** предполагает введение потока в модель, если он разделяет ресурс с любым другим потоком. Если же он обслуживается обособленной группой устройств, то вводить поток и устройства не надо.

**Принцип соразмерности** длительности и периодичности процессов заключается в следующем. В модель включаются заявки, отличающиеся по временным параметрам поступления не более чем на один-два порядка. В противном случае включение редких заявок только усложнит модель, но не уточнит её.

**Усечение модели** предполагает замену цепочки устройств одним устройством с суммарной задержкой на всех устройствах. Часто модели имеют излишнюю размерность из-за включения лишних промежуточных звеньев.

В модель обязательно вносятся те устройства, которые вносят задержку в процессе обслуживания заявки независимо от работы других устройств. В модель в виде одного элемента вводится устройство или последовательность устройств, находящихся между двумя соседними местами хранения данных.

С целью **интеграции ресурсов** необходимо соблюдать следующее правило: в модель включать те устройства, которые вносят задержку в процесс обслуживания заявки независимо от работы других устройств. Для этого выделяем все возможные места хранения (задержки) данных. В модель вводим в виде одного элемента последовательность устройств, находящихся между двумя местами хранения данных.

Два или несколько соседних устройств могут быть заменены одним, если можно получить интегральные параметры обобщённого элемента. Например, задержки во внешних ЗУ учесть в соответствующих параметрах процессора, с которым они взаимодействуют.

При **подборе параметров модели** необходимо отобрать только те параметры, которые влияют на выходные характеристики. Все параметры делятся на параметры обслуживающих элементов и потоков заявок, количественные и функциональные, детерминированные и случайные. Рассмотрим основные из них.

К **количественным параметрам** модели относят те, которые позволяют вычислить время обслуживания  $i$ -й заявки любого потока  $k$ -м элементом. Это время, как правило, случайно и может задаваться функцией распределения.

Время обслуживания  $k$ -м процессором  $i$ -й заявки является функцией производительности

$$T_{ik} = N_{ik} / P_k.$$

Для широкого круга задач время обслуживания заявки процессором пропорционально количеству обрабатываемых данных

$$T_{ik} = V_{ik} \cdot Q_k + T_{ikd},$$

где  $V_{ik}$  – количество данных, обрабатываемых  $k$ -м процессором по  $i$ -й заявке;

$Q_k$  – время обслуживания  $k$ -м процессором единицы данных;

$T_{ikd}$  – независимая от количества обрабатываемых данных составляющая времени обслуживания.

Для всех терминалов, накопителей и коммутаторов время передачи данных определяется количеством передаваемых данных и скоростью обмена:

$$T_{cik} = V_{ik} / S_k,$$

где  $V_{ik}$  – количество данных, передаваемых (принимаемых)  $k$ -м устройством по  $i$ -й заявке;

$S_k$  – скорость передачи, ввода или вывода  $k$ -м устройством.

Для некоторых терминалов и внешних накопителей время обслуживания включает также и время установки в требуемый режим

$$T_{ik} = T_{cik} + T_{sik}$$

где  $T_{sik}$  – время установки  $k$ -го устройства по  $i$ -й заявке

К **функциональным параметрам** модели ВС относят правила (алгоритмы) занесения заявок в очередь – *дисциплины буферизации*, извлечения заявок из очереди на обслуживание – *дисциплины диспетчеризации*, определения длительности обслуживания заявки без отказов и с отказами устройств, обслуживания прерываний и др. Функциональными параметрами, которые управляют потоками заявок, являются *дисциплины маршрутизации* (правила определения последовательности использования заявками обслуживаемых элементов), алгоритмы принятия заявок в систему для обслуживания, распределения ресурсов системы, правила объединения и разделения потоков заявок. Также сюда относятся и приоритеты заявок.

#### 1.4.4. Формализованные схемы описания моделей сложных систем

При исследовании ВС может быть получено большое число разнообразных моделей в зависимости от ориентации, стратификации и детализации.

С целью перевода **обобщённой** модели в **конструктивную** необходимо конкретизировать свойства множеств переменных и операторов. Для определённых классов систем разработаны **формализованные схемы** и **математические методы**, которые позволяют описать функционирование системы или выполнять аналитические исследования.

**Формализованную (математическую) схему** можно определить как звено при переходе от содержательного к формальному описанию процесса функционирования системы с учётом воздействия внешней среды, т. е. имеет место цепочка «концептуальная (описательная) модель – формализованная схема – математическая (аналитическая или имитационная) модель».

Помимо формализованных схем общего вида, позволяющих описывать широкий класс систем, на начальных этапах исследования систем рациональнее использовать типовые формализованные схемы, которые имеют преимущества простоты и наглядности при сужении возможностей применения.

При построении математических моделей процессов функционирования систем можно выделить следующие основные подходы и формализованные схемы:

- непрерывно-детерминированный подход (дифференциальные, интегральные, интегро-дифференциальные уравнения) – **D-схемы**;
- дискретно-детерминированный подход (конечные автоматы) – **F-схемы**;
- дискретно-стохастический подход (вероятностные автоматы) – **P-схемы**;
- непрерывно-стохастический подход (системы массового обслуживания, стохастические сети) – **Q-схемы**;

- дискретно-сетевой подход (сети Петри, обобщающие автоматные модели на моделирование параллельных процессов) – **N-схемы**;
- обобщённый или универсальный подход (агрегативные системы, кусочно-линейные агрегаты) – **A-схемы**.

Использование **D-схем** (англ. *dynamic*) позволяет формализовать процесс функционирования непрерывно-детерминированных систем и оценить их основные характеристики, применяя аналитический или имитационный подход, реализованный в виде соответствующего языка моделирования непрерывных систем или использующий аналоговые и гибридные средства вычислительной техники.

Понятие **F-схемы** или *F-автомата* (англ. *finite automata*) в дискретно-детерминированном подходе к исследованию на моделях свойств объектов является математической абстракцией, удобной для описания широкого класса процессов функционирования реальных объектов в АСУ (элементы и узлы ЭВМ, устройства контроля, регулирования и управления и т. д.). Для этих объектов характерно наличие дискретных состояний и дискретный характер работы во времени, т. е. их описание с помощью **F-схем** является эффективным. Однако этот подход непригоден для описания процессов принятия решений, процессов в динамических системах с наличием переходных процессов и стохастических элементов.

Применение **P-схем** вероятностных автоматов (англ. *probabilistic automat*) имеет большое значение для разработки методов проектирования дискретных систем, проявляющих статистически закономерное случайное поведение, для выяснения алгоритмических возможностей таких систем и обоснования границ целесообразности их использования, а также для решения задач синтеза по выбранному критерию дискретных стохастических систем, удовлетворяющих заданным ограничениям. *P-автоматы* могут использоваться как генераторы марковских последовательностей, которые необходимы при построении и реализации процессов функционирования систем или воздействий внешней среды. **P-схемы** можно применять как в аналитических, так и в имитационных моделях.

**Q-схемы** (англ. *queueing system*) используют в качестве типовых формализованных схем системы массового обслуживания, которые представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания. Возможности оценки характеристик с использованием аналитических моделей ТМО весьма ограничены по сравнению с требованиями практики исследования и проектирования систем, формализуемых в виде **Q-схем**. Существенно расширяют возможности **Q-схем** имитационные модели, автоматизируемые с помощью специализированных языков моделирования.

***N*-схемы** (англ. *net*) основаны на сетях Петри, представляющих собой модели для анализа причинно-следственных связей в системах, где одновременно протекает несколько параллельных процессов. Сеть Петри состоит из некоторого множества позиций и связанных с ними переходов. Данные сетевые модели особенно удобны при построении иерархических конструкций систем. С одной стороны, каждая ***N*-схема** может рассматриваться как макропереход или макропозиция модели более высокого уровня, с другой – переход или позиция ***N*-схемы** допускает детализацию в форме отдельной подсети для более глубокого исследования. Данные виды сетевых схем эффективно используют для моделирования параллельных и конкурирующих процессов в системах различной сложности.

Универсальный подход, предложенный Н.П. Бусленко, позволяет описывать поведение непрерывных и дискретных, детерминированных и стохастических систем и базируется на понятии агрегативной системы (англ. *aggregate system*), представляющей собой формальную схему общего вида, называемую ***A*-схемой**.

Таким образом, конкретизация структуры пространств состояний, входных и выходных воздействий, операторов преобразования входных переменных в выходные и текущих состояний в последующие приводит к специализированным моделям, подчинённым конкретным целям моделирования:

- агрегаты и агрегативные системы;
- кусочно-линейные агрегаты;
- управляющие последовательности;
- стохастические сети;
- системы массового обслуживания;
- автоматные модели;
- модели теории расписаний;
- модели типа сетей Петри.

Рассмотрим более подробно различные типы моделей.

Наиболее общими формализованными моделями представления функционирования АС служат **агрегаты и агрегативные системы**, с помощью которых можно построить непрерывные и дискретные, детерминированные и стохастические модели. В основе такого представления лежит понятие агрегата, с помощью которого моделируются функциональные элементы системы.

**Агрегат** – это абстрактная схема функционирования сложной системы. В каждый момент времени  $t \in (0, T)$  агрегат находится в одном из возможных состояний из множества состояний  $Z$ .

Если состояния  $z = (z_1, \dots, z_l^*)$  являются действительными векторами, то  $z_l$  называют *фазовыми координатами*. С изменением  $t$  состояние  $z$  изменяется как функция времени  $z(t)$ , которая называется *фазовой траекторией*. В общем слу-

чае функции  $z(t)$  представляют собой реализации случайных функций  $Z(t)$ , характеризующих совокупностью многомерных законов распределения  $L[Z(t)]$ .

Функции  $z(t)$  зависят от ряда параметров  $\beta_m$  ( $m = 1, \dots, m^*$ ). В начальный момент времени  $t_0$  состояния  $z$  имеют значения  $z_0$ , задаваемые законом распределения  $L_0[Z(t_0)]$ . Состояния агрегата  $z(t)$  в произвольный момент времени  $t > t_0$  определяются по предыдущим состояниям случайным оператором переходов  $H$ :

$$z(t) = H[z(t_0), t].$$

Конкретное значение  $z(t)$  определяется как реализация в соответствии с законом распределения оператора  $H$ .

Агрегат имеет входные контакты, на которые поступают внешние воздействия, представленные набором своих параметров (входные сигналы)  $x$  в моменты времени  $t_j$  ( $j = 1, 2, \dots; t_{j+1} \geq t_j$ ). Входной сигнал является вектором, размерность которого равна числу входных контактов. На каждый контакт поступает своя координата входного сигнала. В общем случае последовательности вида  $(t_j, x_j)$  оказываются реализациями случайных последовательностей  $(Q_j, X_j)$  с законом распределения  $L[Q, X]$ .

Агрегат имеет особые входные контакты, на которые поступают управляющие сигналы в моменты времени  $t_i$ . Управляющий сигнал  $g$  является элементом множества  $G$ . В общем случае последовательности вида  $(t_j, g_j)$  – это реализации случайных последовательностей  $(Q_j, g_j)$  с законом распределения  $L[Q, g]$ .

На выходе агрегата образуются выходные сигналы. Выходной сигнал  $y$  является элементом некоторого множества  $Y$  и определяется по состояниям агрегата  $z(t)$  при помощи оператора выходов  $G$  (рис. 1.10). За конечный интервал времени в агрегат поступает конечное число входных и управляющих сигналов, и агрегат выдаёт конечное число выходных сигналов.

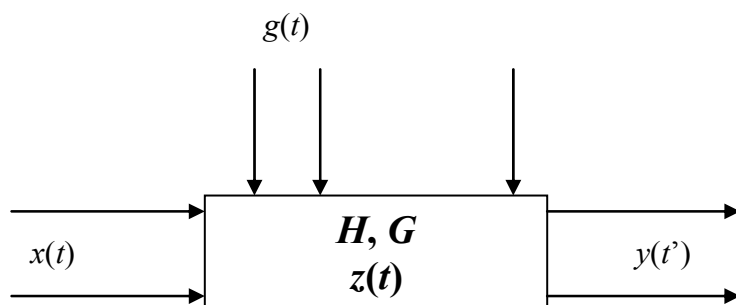


Рис. 1.10. Условное изображение агрегата

У агрегата выделяются особые состояния:

- в моменты получения входного или управляющего сигнала;
- в моменты выдачи выходного сигнала.

Остальные состояния относятся к неособым. Из особых состояний агрегат может переходить в новое состояние скачком.

Вид оператора перехода зависит от наличия особых состояний в рассматриваемый интервал времени.

Расположенная в порядке поступления в систему совокупность входных сигналов называется  $x$ -сообщением, управляющих сигналов –  $g$ -сообщением. Упорядоченная во времени совокупность выходных сообщений называется  $y$ -сообщением.

**Агрегативная система** образуется путём соединения нескольких агрегатов:  $A_1, \dots, A_N$ . Входной сигнал  $x(t)$ , имеющий компоненты  $x_1, \dots, x_m$  и поступающий к агрегату в момент  $t$ , рассматривается как совокупность элементарных сигналов  $x_1(t), \dots, x_m(t)$ , одновременно возникающих на его входе. Аналогично управляющий сигнал  $g(t)$  рассматривается в виде совокупности элементарных сигналов  $g_1(t), \dots, g_p(t)$ , и выходной сигнал  $y(t)$  представляется совокупностью элементарных сигналов  $y_1(t), \dots, y_m(t)$ .

Элементарные сигналы, составляющие сигнал, который поступает из внешней среды, принимаются одним или несколькими агрегатами системы. Каждый выдаваемый во внешнюю среду сигнал складывается из элементарных сигналов, выдаваемых одним или несколькими агрегатами системы. **Внешнюю среду** можно представить в виде фиктивного элемента  $A_0$ , вход которого содержит  $m_0$  входных контактов  $X_i(0)$ , а выход –  $r_0$  выходных контактов  $Y_i(0)$  (рис. 1.11).

Введение некоторых конкретизирующих предположений относительно функционирования агрегата приводит к понятию **кусочно-линейного агрегата**. Теория кусочно-линейных агрегативных систем получила распространение для формализованного описания функционирования ВС на системном уровне.

Дальнейшая конкретизация кусочно-линейного агрегата получила название **метода управляющих последовательностей**.

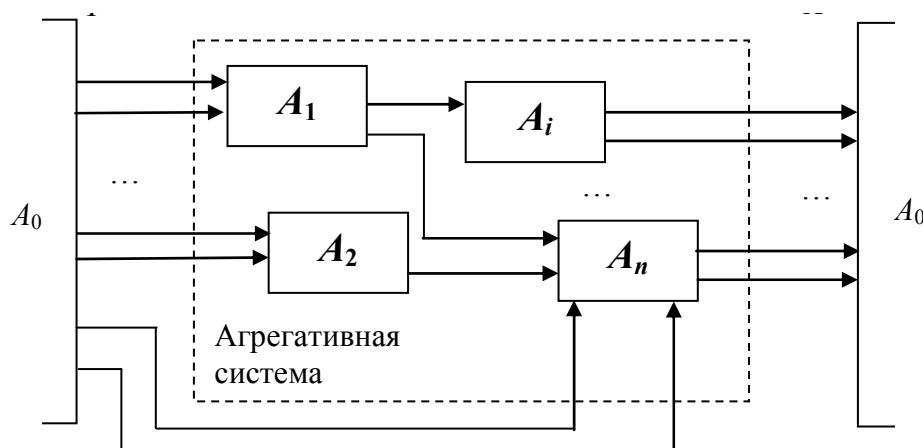


Рис. 1.11. Пример агрегативной системы

Для представления ВС наиболее широко используются **системы массового обслуживания** и **стохастические сети**. Это обусловлено тем, что теория массового обслуживания хорошо развита и допускает аналитическое моделирование.

**Система массового обслуживания** - это объект, в котором выполняется последовательность операций. Система может осуществлять конечное число операций различного типа. Элемент системы, в котором, происходят операции,

называется обслуживающим устройством или прибором. Физическая и алгоритмическая сущность операций игнорируется. Если прибор выполняет операцию, то считается, что он занят, в противном случае прибор свободен.

Операция может начаться лишь после того, как возникнет заявка (требование) на её выполнение, поэтому операции, происходящие в СМО, называют ещё операциями обслуживания заявок.

Рассмотренные выше формализованные математические схемы применимы к системам, функционирующим в непрерывном времени. **Системы, состояния которых определены в дискретные моменты времени  $t_0, t_1, t_2, \dots$** , получили название **автоматов**. Подробно этот вопрос рассматривается в теории автоматов. Если за единицу времени выбран такт  $Dt = t_i - t_{i-1}$ , то просто пишут:  $0, 1, 2, \dots$ . В каждый дискретный момент времени, за исключением  $t_0$ , в автомат поступает входной сигнал  $x(t)$ , под действием которого автомат переходит в новое состояние в соответствии с функцией переходов и выдаёт выходной сигнал, определяемый функцией выходов.

Если автомат характеризуется конечными множествами состояний  $z$ , входных сигналов  $x$  и выходных сигналов  $y$ , он называется **конечным автоматом**. Функции переходов и выходов конечного автомата задаются таблицами, матрицами или графами.

Функция переходов **вероятностного** автомата определяет не одно конкретное состояние, а распределение вероятностей состояний на множестве состояний, а функция выходов – распределение вероятностей на множестве выходных сигналов. Функционирование вероятностных автоматов изучается при помощи аппарата цепей Маркова.

**Сеть Петри** – это графическая модель системы с высокой степенью распараллеливания вычислений, используемая для анализа определённых её свойств.

Сеть Петри – абстрактный автомат для описания асинхронных алгоритмов в виде ориентированного графа, вершины которого соответствуют **действиям**, а дуги – **значениям** (рис. 1.12).

Состав сети Петри:

- множество узлов позиций (мест)  $P_i$ ;
- множество символов;
- множество переходов  $t_i$ ;
- множество ориентированных рёбер (дуг).

Каждый переход связывается с соответствующим множеством входных мест и соответствующим множеством выходных мест. Каждый переход соединяется с каждым из входных мест дугой, идущей из узла-места к переходу, и с каждым выходным местом – дугой, направленной от перехода к узлу-месту. Состояние параллельной системы представляется наличием определённых меток у узлов, а конкретное состояние отображается конкретной конфигурацией меток.



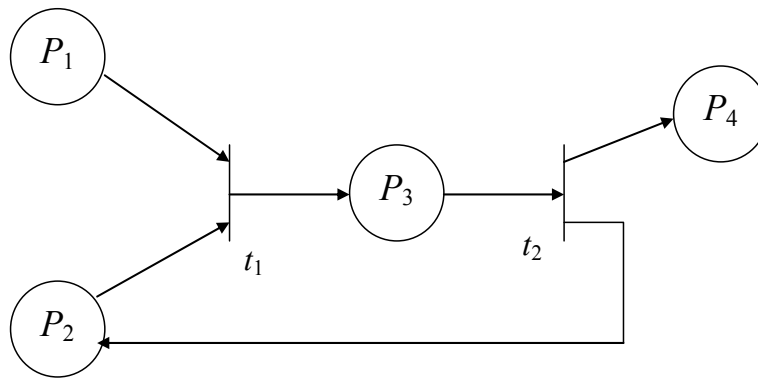


Рис. 1.12. Пример сети Петри

Применение **моделей теории расписаний** целесообразно при наличии детерминированных потоков задач и времён их решения для оптимизации планирования работ в ВС, маршрутизации задач на сетевой структуре ВС, коррекции рассчитанных расписаний.

В качестве математического аппарата используются методы математического программирования.

Модели теории расписания включают:

- множество задач (работ), выполняемых в ВС  $N = (1 \dots n)$ ;
- множество процессоров в составе ВС  $M = (1 \dots m)$ ;

*Одностадийная система* – каждой задаче ставится в соответствие множество процессоров, которые могут решить эту задачу.

*Многостадийная система* – каждой задаче  $i$  в каждой стадии  $j$  ставится в соответствие множество процессоров. При этом задача  $i$  на стадии  $j$  может выполняться на любом процессоре из данного множества, но не более чем на одном одновременно.

Процесс выполнения задач описывается совокупностью функций.

$$S = \{S_1(t), S_2(t), \dots, S_m(t)\}.$$

Каждой задаче могут быть установлены директивные сроки выполнения и последовательность выполнения.

## 2. Аналитическое моделирование

### 2.1. Элементы вероятностных моделей сложных систем

#### 2.1.1. Основы теории массового обслуживания

Основы *теории очередей* (*queueing theory*) были заложены в трудах датского математика, сотрудника Копенгагенской телефонной компании А. К. Эрланга и получили дальнейшее развитие в работах многих зарубежных (Д. Кендалл, Д. Линдли, Дж. Кифер, Дж. Вольфовиц, Р. Джексон, Дж. Литтл, Д. Кокс, Ф. Поллачек, Т. Саати, Ф. Кингман и др.) и отечественных (А.Я. Хинчин, А.Н. Колмогоров, Б.А. Севастьянов, Б.В. Гнеденко, Г.П. Башарин, Г.П. Климов, А.А. Боровков и др.) учёных. В отечественных научных изданиях теория очередей благодаря одному из её основателей выдающемуся математику А.Я. Хинчину получила название *теории массового обслуживания* [8, 18, 31, 32, 61].

Математическая модель *системы массового обслуживания* (СМО) включает пять основных элементов (рис. 2.1): поток поступающих заявок из источника заявок (ИЗ), накопитель (буфер), в котором образуется очередь заявок, один или несколько обслуживающих приборов (устройств, каналов, линий), дисциплину обслуживания и показатели качества обслуживания. Накопитель очереди характеризуется *ёмкостью* и *дисциплиной буферизации* (правилом занесения поступающих заявок в буфер). *Дисциплина обслуживания* представляет собой правило выбора заявок из очереди для обслуживания в приборе.

*Показатели качества* – это показатели, характеризующие качество процесса обслуживания в системе как с точки зрения потребителя, так и владельца системы (эксплуатирующего персонала).

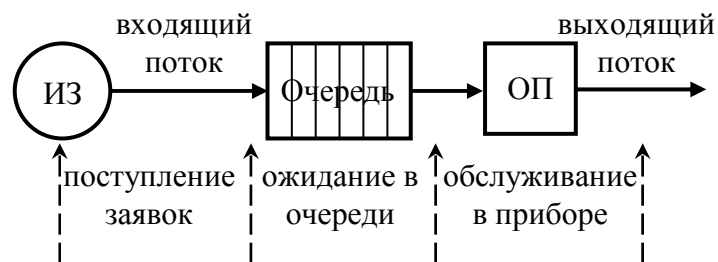


Рис. 2.1. Модель системы массового обслуживания

Д. Кендалл предложил четырёхбуквенное обозначение СМО:

$$A / B / n / m,$$

где  $A$  – тип входящего потока требований;

$B$  – тип распределения длительности обслуживания;

$n$  – число обслуживающих приборов (каналов) (ОП);

$m$  – ёмкость накопителя очереди (по умолчанию – бесконечная ( $\infty$ )).

Позже в классификацию также были добавлены обозначения:

$f_i$  – тип и число приоритетов поступающих заявок (по умолчанию беспriorитетная система);

$d$  – порядок обслуживания заявок (FIFO – первым пришёл, первым обслужен; LIFO – последним пришёл, первым обслужен; RND – случайная выборка из очереди (по умолчанию – FIFO)).

Устоявшимися обозначениями для законов распределений  $A$  и  $B$  являются следующие:

$M$  (Markovian) – экспоненциальное (показательное) распределение (от Маркова);

$D$  (Deterministic) – детерминированное (регулярное) распределение;

$E_k$  (Erlangian) – распределение Эрланга  $k$ -го порядка;

$U$  (Uniform) – равномерное распределение;

$H_k$  (Hiperexponential) – гиперэкспоненциальное распределение  $k$ -го порядка;

$h_k$  (Hipoexponential) – гипоекспоненциальное распределение  $k$ -го порядка;

$C_k$  (Coxian) – распределение Кокса  $k$ -го порядка;

$PH$  (Phase-type) – распределение фазового типа;

$g$  (Gamma) – гамма-распределение;

$P$  (Pareto) – распределение Парето;

$G$  (General) – произвольное распределение;

$GI$  (General Independent) – произвольное распределение с независимыми интервалами между событиями.

**Пример.** Обозначение СМО –  $H_2 / E_3 / 2 / 10$  – раскрывается следующим образом:

- $H_2$  – гиперэкспоненциальное распределение 2-го порядка времени между входящими заявками;
- $E_3$  – распределение Эрланга 3-го порядка времени обслуживания заявок;
- 2 прибора обслуживания;
- 10 мест для ожидания в накопителе очереди.

Определим случайные процессы, описывающие одноканальную систему (рис. 2.2). Поступающая в систему заявка ( $C_n$ ) застаёт обслуживающий прибор занятым и ожидает окончания обслуживания предыдущей заявки ( $C_{n-1}$ ) в очереди в течение времени  $w_n$ . Затем занимает освободившийся прибор и обслуживается время  $x_n$ , после чего покидает систему. Общее время пребывания в системе составит  $s_n = w_n + x_n$ . Поступающая заявка может застать систему свободной ( $C_{n+2}$ ) и тогда она без ожидания поступает сразу на обслуживание. Общее время пребывания в системе в этом случае составит  $s_{n+2} = x_{n+2}$ .

Существует большое число СМО, различающихся структурной и функциональной организацией. В то же время, разработка аналитических методов рас-

чѐта характеристик функционирования СМО во многих случаях предполагает наличие ряда ограничений множества исследуемых СМО.

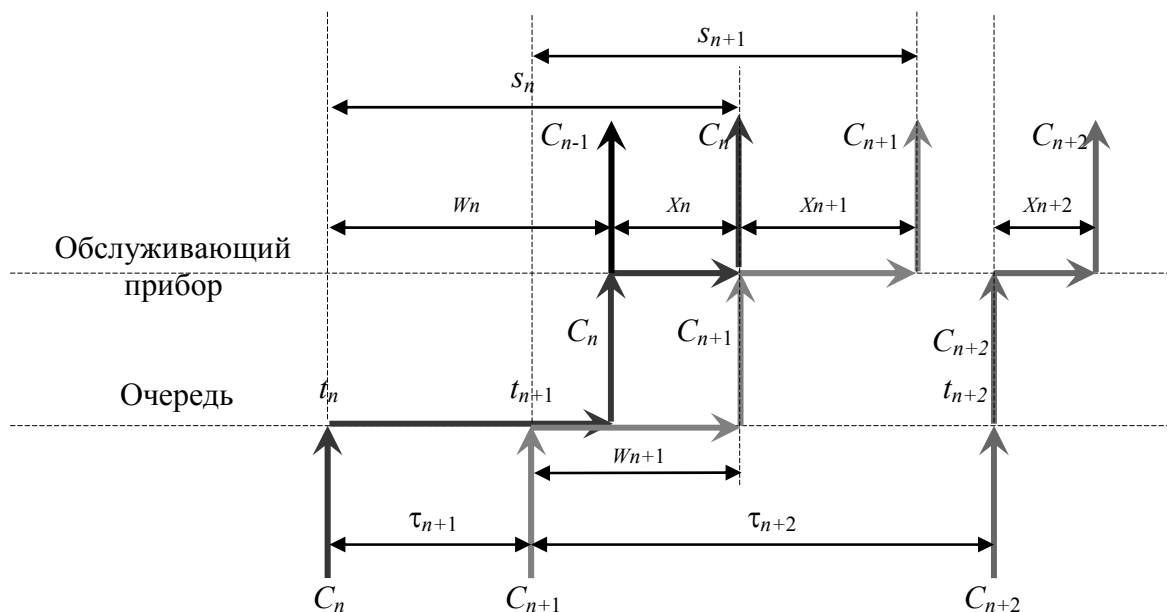


Рис. 2.2. Временная диаграмма вероятностных процессов в СМО

Системы массового обслуживания могут быть классифицированы:

- по числу мест в накопителе;
- по числу обслуживающих приборов;
- по количеству классов заявок, поступающих в СМО.

1. По **числу мест в накопителе** СМО делятся на системы:

1) **без накопителя**, в которых заявка, поступившая в систему и заставшая все обслуживающие приборы занятыми обслуживанием более высокоприоритетных заявок, получает отказ и теряется; такие системы называются **СМО с отказами**;

2) **с накопителем ограниченной ёмкости (СМО с потерями)**, в которых поступившая заявка теряется, если она застаёт накопитель заполненным до конца;

3) **системы с накопителем неограниченной ёмкости (СМО без потерь)**, в которых для любой поступившей заявки всегда найдѐтся место в накопителе для ожидания.

2. По **количеству обслуживающих приборов** СМО делятся на два вида:

- 1) **одноканальные**, содержащие один прибор П;
- 2) **многоканальные**, содержащие  $K$  обслуживающих приборов  $\Pi_1, \dots, \Pi_K$  ( $K > 1$ ).

В многоканальных СМО обычно предполагается, что все приборы идентичны и равнодоступны для любой заявки, то есть при наличии нескольких свободных приборов поступившая заявка с равной вероятностью может попасть в любой из них на обслуживание.

3. По **количеству классов (типов) заявок**, поступающих в СМО, различают:

- системы с **однородным потоком** заявок;
- системы с **неоднородным потоком** заявок.

Однородный поток заявок образуют заявки одного класса, а неоднородный поток представляет собой поток заявок нескольких классов.

В СМО, представляющей собой абстрактную математическую модель, *заявки относятся к разным классам* в том случае, если они в моделируемой реальной системе различаются хотя бы одним из следующих факторов:

- *длительностью обслуживания*;
- *приоритетами*.

**Приоритет** – преимущественное право на занесение (в накопитель) или выбор из очереди (для обслуживания в приборе) заявок одного класса по отношению к заявкам других классов. Если же заявки не различаются длительностью обслуживания и приоритетами, то в СМО они могут быть представлены как заявки одного класса, независимо от их физической сущности.

### 2.1.2. Обзор основных случайных процессов

В теории очередей, как и в целом в моделировании, используется ряд **случайных процессов** (рис. 2.3).

Случайный процесс можно представить как процесс движения частиц в некотором пространстве. В основе классификации случайных процессов лежат три характеристики:

- пространство состояний;
- индексирующий параметр (время);
- статистические зависимости между случайными значениями процесса.

*Пространством состояний* называют множество возможных значений (состояний) случайного процесса  $X(t)$ . Пространство состояний может быть **дискретным** (если множество возможных положений частиц в пространстве конечно или счётно) и **непрерывным** (если множество положений частиц представляет собой конечный или бесконечный непрерывный интервал).

*Индексирующий параметр* (время) фиксирует допустимые моменты изменения частицы в пространстве. Время также может быть **дискретным** (если допускаются изменения положения частиц в конечном или счётном числе моментов времени) и **непрерывным** (если допускаются изменения в любой момент конечного или бесконечного промежутка времени).

*Статистические зависимости* случайных величин задаются с помощью полной функции совместного распределения вероятностей этих случайных величин. Многие случайные процессы задаются существенно проще. Например, в марковских цепях с дискретным временем зависимость распространяется назад на одну единицу времени.

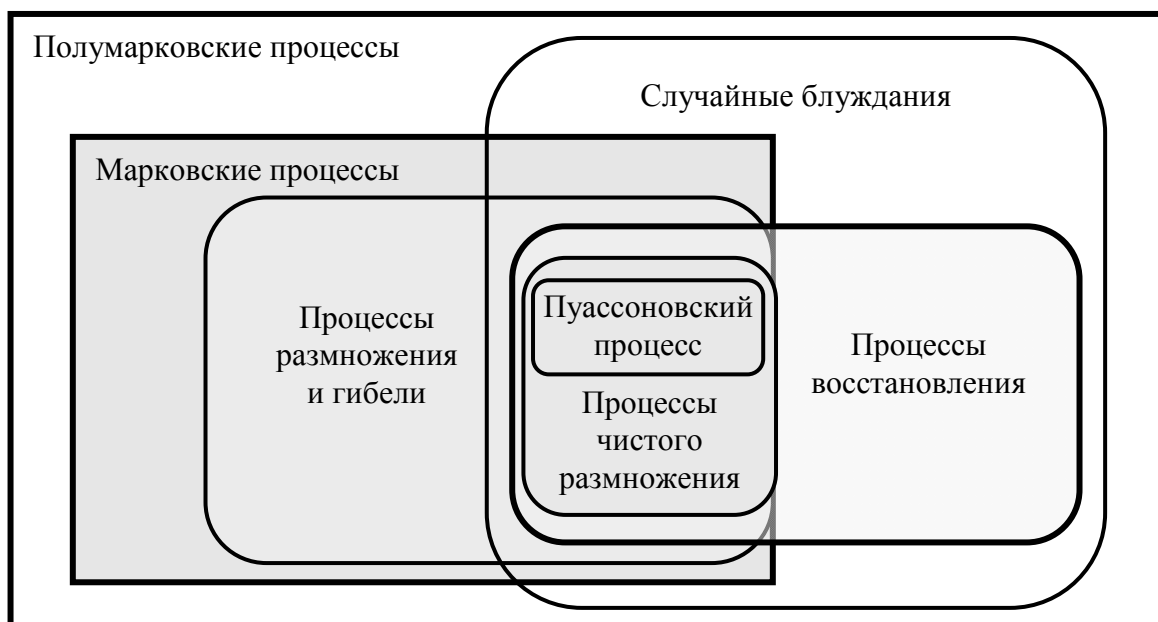


Рис. 2.3. Взаимосвязь между случайными процессами

*Марковский процесс* – это случайный процесс, для которого вероятность следующего состояния зависит только от текущего состояния и не зависит от предыдущих состояний.

*Процессы размножения и гибели* – частный случай марковских процессов, для которых допустимы переходы только в соседнее состояние. Термин «ПГР» ведёт начало от биологических задач, где такими процессами описывается изменение численности биологических популяций.

*Случайные блуждания* представляются как передвижения частицы из одного состояния в другое в некотором пространстве состояний. Задача состоит в нахождении местоположения частицы в этом пространстве.

В *полумарковских случайных процессах* вероятности перехода произвольны, как произвольны и распределения вероятностей времени между переходами из одного состояния в другое.

*Процессы восстановления* представляются как последовательность точек на временной оси, соответствующих моментам перехода из одного состояния в другое. Промежутки времени между переходами – независимые одинаково распределённые случайные величины по произвольному закону.

*Процессы размножения и гибели* образуют подмножество *марковских процессов*, которые, в свою очередь, являются подмножеством *полумарковских процессов*. Процессы восстановления образуют подмножество в множестве процессов случайных блужданий, которое также является подмножеством *полумарковских процессов*. Марковские процессы, для которых вероятности переходов зависят только от разности индексов состояний, перекрывают случайные блуждания без последствий. Последние при условии перехода только в соседние состояния пересекаются с классом процессов размножения и гибели,

при переходе только в соседнее старшее состояние получаем *процесс чистого размножения*. Если же при этом добавить постоянную интенсивность размножения, то получим *пуассоновский процесс*, который играет главную роль в теории массового обслуживания, поскольку обладает рядом замечательных свойств, позволяющих существенно облегчить труд исследователя.

### 2.1.3. Моделирование рабочей нагрузки системы

Для АС *рабочей нагрузкой* является поток заданий на обработку и хранение данных, распределённый во времени. Задание определяет программу обработки данных и необходимые ресурсы.

Программа полностью определяет последовательность использования ресурсов и функции, которые должны быть выполнены каждым ресурсом.

Вне зависимости от того, каким образом инициируется выполнение программы – представлением управляющей информации в виде задания или самим фактом появления запроса, сигнала или новых данных, требование на выполнение работы при моделировании принято называть *заявкой*. Заявка характеризуется моментом появления на входе системы, статусом по отношению к другим заявкам и параметрами, определяющими потребности в ресурсах на выполнение вычислительной работы. Совокупность заявок, распределённая во времени, – это *поток заявок*. Таким образом, рабочей нагрузкой ВС является поток заявок.

Потоки заявок по своей природе бывают *случайные* и *неслучайные*.

*Неслучайные потоки* описываются графиком, расписанием, таблицей или детерминированной функцией, и используются, как правило, в теории расписаний. В теории очередей при моделировании некоторых систем используются детерминированные потоки – с постоянным временем между событиями.

*Случайные потоки* – это случайные процессы, задаваемые функцией распределения времени между моментами поступления соседних заявок, либо распределение числа заявок, поступивших в очередной промежуток времени.

Поток заявок может быть *однородным* и *неоднородным*. В универсальных ВС коллективного пользования поток заявок зачастую отличается крайней неоднородностью. При моделировании неоднородный поток может быть представлен совокупностью однородных потоков. В каждый из однородных потоков объединяются заявки с идентичным статусом и параметрами.

Одним из основных параметров потока заявок является *промежуток времени между моментами поступления двух соседних заявок*. Этот промежуток может быть постоянной или переменной величиной. Совокупность промежутков может иметь периодически повторяющееся количество значений, задаваемых определённым правилом, или большое количество значений, дающее основание говорить, что это величина случайная. В общем случае поток заявок рассматривается как случайный процесс, характеризуемый функцией распределения пе-

риода поступления заявок. Среднее число заявок, поступающих в единицу времени, называется **интенсивностью потока** ( $\lambda$ ).

Поток заявок может быть *стационарным* или *нестационарным*. Это зависит от того, изменяются или не изменяются статистические характеристики потока во времени. Режим работы реальных ВС далеко не всегда является стационарным. Рабочая нагрузка на ВС зачастую изменяется в зависимости от времени суток, дней недели, месяцев квартала или года и от состояния управляемого объекта или технологического процесса для управляющих систем. Интенсивность может представляться в виде функции времени.

Модель рабочей нагрузки представляет собой совокупность потоков заявок:

$$X = \{X_1, \dots, X_M\},$$

где  $M$  – число потоков.

Каждый поток характеризуется приоритетом  $K$ , правилами и параметрами, определяющими время поступления заявок в систему  $T_z$ , последовательность использования ресурсов  $B$  и величину использования ресурсов  $C$ :

$$X_m = \{K, T_z, B, C\}.$$

Приоритет указывает статус заявок данного потока среди других потоков, например, определяет степень срочности заявки. Задаётся числовым или символьным кодом.

Для определения моментов поступления заявок в систему в модели потока указывают тип потока и соответствующие параметры:

- для детерминированного потока – период  $\tau_z$  или интенсивность  $\lambda$ ;
- для регулярного потока – расписание;
- для стохастического потока – функцию распределения интервалов между соседними заявками;
- для нестационарного потока – функцию распределения, задаваемую в виде случайного процесса.

По способу маршрутизации заявок (последовательности использования ресурсов) различают **трассировочные** и **вероятностные** модели потоков. В трассировочных моделях последовательность использования ресурсов указывается в явном виде, в вероятностных моделях задаются вероятности перехода заявки к одному из следующих ресурсов.

Все параметры модели потоков заявок определяются по результатам статистической обработки измерений или оценок экспертиз рабочей нагрузки.

Основными характеристиками случайного потока заявок являются:

$P_k(t, \tau)$  – вероятность поступления  $k$  заявок на интервале  $\tau$  с момента  $t$ ;

$\Pi_1(t, \tau)$  – вероятность появления хотя бы одной заявки за интервал  $\tau$  с момента  $t$ :

$$\Pi_1(t, \tau) = \sum_{k=1}^{\infty} P_k(t, \tau) = P_1(t, \tau) + P_2(t, \tau) + \dots$$



$M[k(t, \tau)]$  – математическое ожидание числа заявок, прибывших за интервал  $\tau$ :

$$M[k(t, \tau)] = \sum_{k=0}^{\infty} k p_k(t) = 0 \cdot p_0(t) + 1 \cdot p_1(t) + 2 \cdot p_2(t) + \dots$$

**Простейшим случайным потоком**, изучаемым в теории случайных процессов, является **стационарный пуассоновский поток**. Для него

$$P_k(\tau) = \frac{(\lambda \tau)^k}{k!} e^{-\lambda \tau}.$$

Пуассоновскому закону распределения вероятности числа событий на интервале времени соответствует экспоненциальная функция распределения времени между событиями, которое обладает свойствами:

- 1) стационарность;
- 2) ординарность;
- 3) отсутствие последдействия.

**Стационарность**: если вероятность  $P_k(t, \tau)$  определяется только длиной  $\tau$  и не зависит от положения интервала на оси времени:

$$P_k(t, \tau) = P_k(\tau).$$

**Ординарность**: если вероятность появления более одного требования заявок на интервале  $\tau$  имеет высший порядок малости в сравнении с  $\tau$ ,

$$\lim_{\tau \rightarrow 0} \frac{\Pi_2(t, \tau)}{\tau} = 0, \text{ где } \Pi_2(t, \tau) = \sum_{i=2}^{\infty} P_i(t, \tau).$$

**Отсутствие последдействия**: если события  $\xi_{k1}(t_1, \tau_1)$  и  $\xi_{k2}(t_2, \tau_2)$  независимы для неперекрывающихся интервалов времени,

$$P_k(t, \tau | x) = P_k(t, \tau),$$

где  $P_k(t, \tau | x)$  – условная вероятность поступления  $k$  событий в интервале  $(t, \tau)$  при условии, что от предыдущего события до момента  $t$  прошло  $x$  единиц времени.

Для случайных потоков определены **параметр потока**  $\Pi(t)$  и **интенсивность потока**  $\lambda(t)$ , как пределы отношений

$$\pi(t) = \lim_{\tau \rightarrow 0} \frac{\Pi_1(t, \tau)}{\tau}, \quad \lambda(t) = \lim_{\tau \rightarrow 0} \frac{M[k(t, \tau)]}{\tau}.$$

В идеальном случае  $\lambda(t) \geq \pi(t)$ . Для пуассоновского потока  $\lambda(t) = \pi(t)$ .

В связи с этим свойством  $p_k(t, t + \tau) = p_k(\tau)$ .

**Достоинства простейшего потока**:

- 1) удовлетворительно описывает множество реальных ситуаций, соответствующих появлению редких случайных событий;
- 2) сумма конечного числа простейших потоков даёт простейший поток с суммарным значением параметра потока;
- 3) суммирование стационарных потоков с ограниченным последствием

(рекуррентных) при неограниченном увеличении числа составляющей потока даёт поток, близкий к простейшему;

4) при прореживании потока, когда случайным образом выбрасывается каждая  $k$ -я заявка, с увеличением вероятности выбрасывания очередной заявки поток стремится к простейшему;

5) распределение числа заявок потока за фиксированный интервал времени является распределением Пуассона.

Перечислим некоторые обобщения стационарного пуассоновского потока.

1. Обобщением простейшего потока по **свойству стационарности** является *нестационарный пуассоновский поток*, где  $\lambda(t)$  является детерминированной функцией  $t$ . Сохраняется последствие, ординарность, но вероятность наступления  $k$  событий за  $\tau$  будет зависеть от начала отсчёта.

2. Обобщением простейшего потока по **свойству ординарности** является *неординарный пуассоновский поток*, в котором в один момент времени может появиться более одной заявки.

3. Поток с интенсивностью, являющейся его **случайной величиной**, распределённой и рассмотренной Хинчиным, называется *обобщённым пуассоновским потоком*. Это совокупность обычных пуассоновских потоков с любыми интенсивностями (при статическом моделировании).

4. Потоки, интенсивность которых представляет собой **случайную функцию** времени либо случайную величину:

- интенсивность является реализацией стационарного случайного процесса с непрерывным временем;

- *дважды стохастические пуассоновские потоки*. Интенсивность потока – кусочно-постоянная (скачкообразный) случайный процесс, т. е. интенсивность сохраняет постоянное значение в течение некоторого случайного  $t$ , распределённого по известному закону.

5. Обобщением простейшего потока по характеру последствия является **поток с ограниченным последствием**. Для него случайные величины – интервалы времени между событиями потока – взаимно независимы. Каждая из них может иметь произвольную функцию распределения:

$$F_k(t) = P(z_k < t)$$

Важный частный случай – рекуррентные потоки (потоки восстановления). Для них  $F_k(t) = F(t)$ , т. е. все интервалы времени между событиями распределены по одному и тому же закону. Стационарные потоки восстановления (потоки Пальма) являются хорошим приближением к реальным.

## 2.1.4. Показатели эффективности СМО

Приведём основные показатели эффективности СМО.

**Показатели непрерывных величин (временные показатели):**

- распределение времени пребывания заявки в системе, например,  $G(t)$ ;
- распределение времени ожидания заявки, например,  $W(t)$ ;
- распределение времени простоя в системе, например,  $V(t)$ ;
- набор средних значений:
  - среднее время ожидания в очереди  $W$ ;
  - среднее время пребывания в системе  $T$ .

**Показатели дискретных величин (счётные показатели):**

- вероятность нахождения  $k$  заявок в системе;
- вероятность нахождения  $k$  заявок в очереди;
- вероятность отказа в приёме (обслуживании) заявки;
- вероятность нулевого ожидания;
- набор средних значений:
  - средняя длина очереди  $L$ ;
  - среднее число заявок в системе  $N$ .

**Относительные показатели (комбинированные показатели):**

- коэффициент нагрузки  $\zeta$ ;
- коэффициент удельной нагрузки  $\chi$ ;
- коэффициент загрузки системы  $\rho$ ;
- коэффициент простоя системы  $\eta$ .

Остановимся чуть подробнее на некоторых показателях. Для систем с ограниченной ёмкостью накопителя очереди вероятность отказа в обслуживании заявки будет зависеть от **ёмкости накопителя** и от **коэффициента нагрузки системы**, характеризующейся отношением интенсивности поступления заявок  $\lambda = 1/\tau_{\text{ср}}$  к интенсивности обслуживания заявки одним прибором  $\mu = 1/x_{\text{ср}}$  ( $x_{\text{ср}}$  – среднее время обслуживания):

$$\zeta = \lambda/\mu = x_{\text{ср}}/\tau_{\text{ср}}.$$

Для интенсивности  $\lambda$  также верно  $\lambda = \lambda' + \lambda''$ , где  $\lambda'$  – интенсивность заявок, поступивших на обслуживание,  $\lambda''$  – интенсивность потерянных заявок.

Определим **коэффициент загрузки системы** (или коэффициент использования) как

$$\rho = \lambda'/n\mu = \min\left(\frac{(1-p_{\text{отк}})\zeta}{n}; 1\right) \leq 1$$

где  $p_{\text{отк}}$  – вероятность отказа в обслуживании (вероятность потери заявки).

Для  $\rho$  также верно выражение

$$\rho = \lim_{T \rightarrow \infty} \frac{1}{nT} \sum_{i=1}^n T_i;$$

где  $n$  – количество приборов в системе;

$T$  – общее время наблюдения системы;

$T_i$  – время работы обслуживающего прибора,  $i = 1 \dots n$ .

**Загрузка системы**, в отличие от **нагрузки**, определяется через интенсивность только *обслуженных* заявок, поскольку потерянные заявки не обслуживаются в приборах и, следовательно, не загружают систему.

Также полезным показателем для исследования систем, особенно в условиях перегрузки, является **коэффициент удельной нагрузки**, характеризующий приведённое к одному прибору количество работы, которое нужно выполнить:

$$\chi = \lambda / n\mu = x_{\text{ср}} / n\tau_{\text{ср}}.$$

Удельную нагрузку можно трактовать как потенциальную загрузку, приходящуюся на один прибор. Если  $\chi < 1$ , то система будет справляться с нагрузкой. Если  $\chi > 1$ , то система будет перегружена, и коэффициент как раз показывает, во сколько раз перегружена система.

Для систем с неограниченной ёмкостью накопителя очереди удельная нагрузка совпадает с коэффициентом загрузки ( $\chi = \rho$ ), так как интенсивность поступления заявок  $1/\tau_{\text{ср}}$  должна быть меньше интенсивности обслуживания  $1/x_{\text{ср}}$ :

$$1 / \tau_{\text{ср}} = \lambda < \mu = 1 / x_{\text{ср}}.$$

В данных системах отказ в обслуживании невозможен и очередь будет бесконечно расти при  $\chi \geq 1$ . Следовательно, условием наличия стационарного режима будет требование  $\chi < 1$ .

В системах с ограниченной очередью допускается  $\chi \geq 1$ , но резко будет возрастать вероятность отказов, поскольку  $\chi$  показывает, **во сколько раз входная нагрузка превышает суммарную мощность обслуживающих приборов**.

### 2.1.5. Законы сохранения в теории массового обслуживания

Изучение законов сохранения начнём с изучения марковских процессов, точнее, частного случая – процессов размножения и гибели.

Процессы размножения и гибели представляют собой частный случай марковских процессов, для которых допустимы переходы из состояния  $E_k$  только в соседние состояния  $E_{k-1}$ ,  $E_k$  и  $E_{k+1}$ .

Граф переходов между состояниями приведён на рис. 2.4. Состояния процесса пронумерованы от нуля до бесконечности, рёбра указывают возможные переходы между состояниями.

Процесс находится в состоянии  $E_k$ , если объём популяции равен  $k$ . Переход из  $E_k$  в  $E_{k+1}$  соответствует рождению новой особи и наоборот, переход из  $E_k$  в  $E_{k-1}$  соответствует гибели особи. Попасть в состояние  $E_k$  можно только из состояний  $E_{k-1}$  и  $E_{k+1}$ ; также и покинув его можно перейти только в эти состояния.

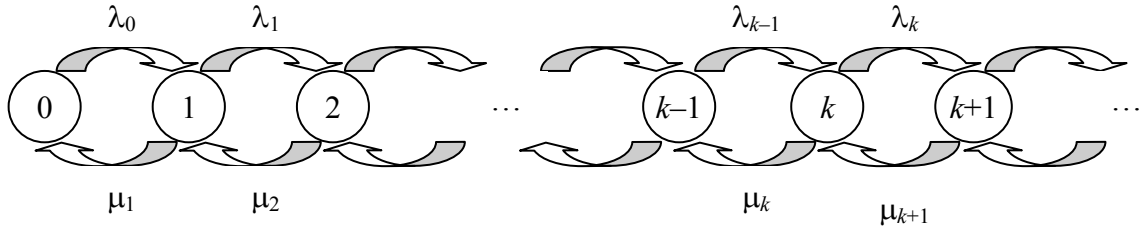


Рис. 2.4. Диаграмма переходов для процессов размножения и гибели

Скорость размножения характеризуется интенсивностью  $\lambda_k$ , скорость гибели – интенсивностью  $\mu_k$ . Интенсивности  $\lambda_k$  и  $\mu_k$  не зависят от времени, только от состояния. Тогда опишем **условия существования** процесса размножения-гибели: 1) вероятность точно одного рождения в течение времени  $(t, t + \Delta t)$  равна  $\lambda_k \Delta t + o(\Delta t)$ ; 2) вероятность точно одной гибели в течение времени  $(t, t + \Delta t)$  равна  $\mu_k \Delta t + o(\Delta t)$ ; 3) вероятность отсутствия рождений в течение времени  $(t, t + \Delta t)$  равна  $1 - \lambda_k \Delta t + o(\Delta t)$ ; 4) вероятность отсутствия гибелей в течение времени  $(t, t + \Delta t)$  равна  $1 - \mu_k \Delta t + o(\Delta t)$ .

Состояние случайного процесса характеризуется количеством заявок в системе и описывается вероятностью

$$P_k(t) = P(x(t) = k).$$

Рассмотрим возможные изменения состояния системы в течение времени  $(t, t + \tau)$ , приводящие её в состояние  $E_k$ :

- 1) в момент  $t$  число заявок в системе было  $k$  и в течение времени  $(t, t + \Delta t)$  не произошло изменения состояния;
- 2) в момент  $t$  число заявок в системе было  $k - 1$  и в течение времени  $(t, t + \Delta t)$  поступила одна заявка;
- 3) в момент  $t$  число заявок в системе было  $k + 1$  и в течение времени  $(t, t + \Delta t)$  обслужилась одна заявка.

Первый случай происходит с вероятностью  $P_k(t)$ , умноженной на вероятность перехода  $p_{k,k}(\Delta t)$  за время  $\Delta t$ , второй случай – с вероятностью  $P_{k-1}(t)$ , умноженной на вероятность перехода  $p_{k-1,k}(\Delta t)$  за время  $\Delta t$ , третий случай – с вероятностью  $P_{k+1}(t)$ , умноженной на вероятность перехода  $p_{k+1,k}(\Delta t)$  за время  $\Delta t$ . Переходы в состояние  $E_k$  из состояний, не являющихся соседними, не рассматриваются ввиду того, что вероятности таких событий имеют порядок  $o(\Delta t)$ . Запишем уравнение Колмогорова:

$$P_k(t + \Delta t) = P_k(t)p_{k,k}(\Delta t) + P_{k-1}(t)p_{k-1,k}(\Delta t) + P_{k+1}(t)p_{k+1,k}(\Delta t) + o(\Delta t), k \geq 1.$$

При  $k=0$  уравнение примет несколько другой вид:

$$P_0(t + \Delta t) = P_0(t)p_{00}(\Delta t) + P_1(t)p_{10}(\Delta t) + o(\Delta t), k = 0.$$

Кроме того, необходимо выполнить условие нормировки:  $\sum_{k=0}^{\infty} P_k(t) = 1$ .

Теперь перейдём к составлению системы уравнений Чепмена – Колмогорова с использованием интенсивностей переходов, учитывая описанные выше четыре условия:

$$P_k(t + \Delta t) = P_k(t)[1 - \lambda_k \Delta t + o(\Delta t)][1 - \mu_k \Delta t + o(\Delta t)] + P_{k-1}(t)[\lambda_{k-1} \Delta t + o(\Delta t)] + P_{k+1}(t)[\mu_{k+1} \Delta t + o(\Delta t)] + o(\Delta t), k \geq 1;$$

$$P_0(t + \Delta t) = P_0(t)[1 - \lambda_0 \Delta t + o(\Delta t)] + P_1(t)[\mu_1 \Delta t + o(\Delta t)] + o(\Delta t), k = 0.$$

Проведя преобразования и перейдя к пределу при  $\Delta t \rightarrow \infty$ , получим следующую дифференциальную систему уравнений:

$$\begin{aligned} dP_k(t)/dt &= -(\lambda_k + \mu_k)P_k(t) + \lambda_{k-1}P_{k-1}(t) + \mu_{k+1}P_{k+1}(t), k \geq 1; \\ dP_0(t)/dt &= -\lambda_0P_0(t) + \mu_1P_1(t), k = 0. \end{aligned} \quad (2.1)$$

В теории очередей чаще исследуют установившийся режим при достаточно большом промежутке времени с начала моделирования ( $t \rightarrow \infty$ ). Тогда получим стационарные вероятности состояний, не зависящие от времени:

$$P_k = \lim_{t \rightarrow \infty} P_k(t)$$

и, соответственно, система (2.1) примет вид:

$$\begin{aligned} 0 &= -P_k(\lambda_k + \mu_k) + \lambda_{k-1}P_{k-1} + \mu_{k+1}P_{k+1}; \\ 0 &= -\lambda_0P_0 + \mu_1P_1. \end{aligned}$$

Откуда получаем выражения для стационарных вероятностей:

$$P_k = \frac{\lambda_0 \lambda_1 \dots \lambda_{k-1}}{\mu_1 \mu_2 \dots \mu_k} P_0 = P_0 \prod_{i=0}^k \frac{\lambda_i}{\mu_{i+1}}; P_1 = \frac{\lambda_0}{\mu_1} P_0.$$

Теперь рассмотрим основные законы сохранения в теории массового обслуживания, сформулированные в своё время в трудах М. Краковски, Л. Клейнрока [5, 31], Ю. Рыжикова [20] и др. Большинство этих законов справедливо только для систем с неограниченной очередью, поскольку предполагается, что весь входной поток с интенсивностью  $\lambda$  попадает на обслуживание.

**1. Закон сохранения заявок.** Частота поступления заявок в канал обслуживания в среднем равна частоте выходов из этого канала.

Для одноканальной системы  $G/G/1$  имеем

$$\lambda = \lambda_{\text{вх}} = \lambda_{\text{вых}} = (1 - P_0)/x_{\text{ср}}.$$

При этом коэффициент нагрузки (загрузки) системы  $\rho = \lambda_{\text{вх}} x_{\text{ср}}$ , интенсивность обслуживания  $\mu = 1/x_{\text{ср}}$ .

Для многоканальной системы  $G/G/n$

$$\lambda = \lambda_{\text{вх}} = \lambda_{\text{вых}} = (1 - P_0)n/x_{\text{ср}}.$$

При этом коэффициент нагрузки (загрузки) системы  $\rho = \lambda_{\text{вх}} x_{\text{ср}}/n$

**2. Закон сохранения очереди.** Распределение числа заявок, прибывших за время ожидания начала обслуживания, совпадает с распределением длины очереди перед прибытием заявки.

Для произвольной системы массового обслуживания имеет место формула Литтла:

$$L = \lambda \cdot W,$$

где  $L$  – количество заявок, поступивших в СМО за время ожидания обслуживания в очереди первой заявки, или длина очереди;

$\lambda$  – интенсивность входного потока;

$W$  – среднее время ожидания заявки в очереди.

**3. Закон сохранения числа заявок в системе.** *Распределение числа заявок, прибывших за время пребывания заявки в системе, совпадает с распределением полного числа заявок в системе перед прибытием заявки.*

Аналогичная предыдущему закону формула Литтла может быть распространена в целом на всю систему:

$$N = \lambda T,$$

где  $N$  – количество заявок, поступивших в СМО за время пребывания в системе первой заявки;

$T$  – среднее время пребывания заявки в системе.

**4. Закон сохранения вероятностей состояний (потока переходов).** *В стационарном режиме средние частоты переходов через любой разрез диаграммы переходов в противоположных направлениях равны.*

В данном случае, исходя из уравнений Колмогорова для произвольной СМО проведя разрез диаграммы переходов (рис. 2.5), получим уравнение

$$\lambda_{k-1} P_{k-1} = \mu_k P_k.$$

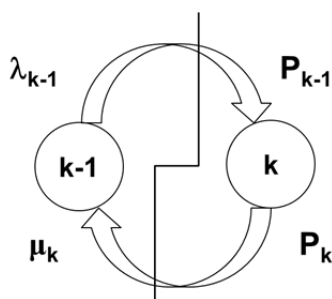


Рис. 2.5. Сохранение потока переходов

**5. Закон сохранения объёма работы.** *Распределение объёма невыполненной работы, находящегося в системе, постоянно и не зависит от выбора конкретной дисциплины обслуживания.*

**Для одноканальной СМО:**

– средний объём работы совпадает со средним временем ожидания начала обслуживания вновь прибывшей заявки;

– средний остаток начатого обслуживания учтём с вероятностью занятости системы  $\rho = \lambda/\mu$ ;  $f_1 = \rho \cdot b_2/b_1 = \lambda \cdot b_2$ .

## 2.2. Анализ систем с простейшими потоками событий

### 2.2.1. Одноканальная система с отказами M/M/1/0

Простейшая одноканальная система M/M/1/0 с пуассоновским входящим потоком и экспоненциальным временем обслуживания характеризуется отсутствием накопителя очереди (рис. 2.6, а). Заявка, поступившая в систему и заставшая прибор занятым, получает отказ в обслуживании и удаляется из системы, поэтому часто такую систему называют *системой с отказами*.

В системе всего два состояния (рис. 2.6, б): система свободна (0 заявок) и система занята (1 заявка).

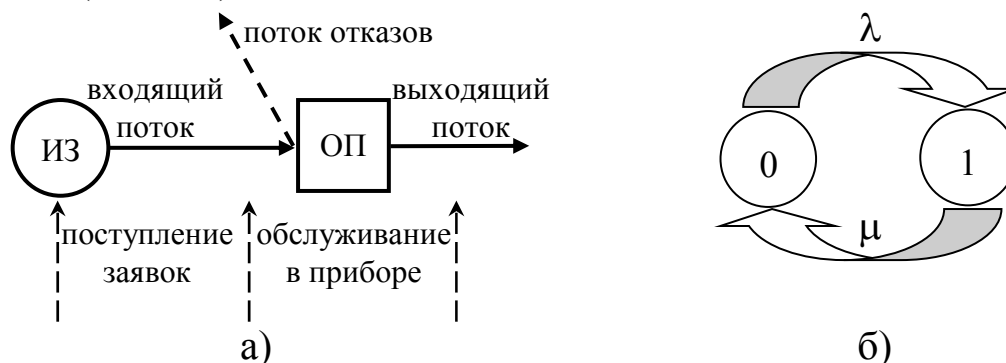


Рис. 2.6. Система M/M/1/0: а) Q-схема; б) диаграмма переходов

Поступающие заявки образуют пуассоновский поток с интенсивностью  $\lambda$  и обслуживаются с интенсивностью  $\mu = 1/x_{\text{ср}}$ , где  $x_{\text{ср}}$  – среднее время обслуживания.

Составим и решим систему уравнений для данной модели с отказами:

$$\begin{cases} \lambda P_0 = \mu P_1 \\ P_0 + P_1 = 1 \end{cases} \Rightarrow \begin{cases} P_0 = \mu / (\lambda + \mu) \\ P_1 = \lambda / (\lambda + \mu) \end{cases}$$

Таким образом, **вероятность отказа в обслуживании** будет совпадать с вероятностью нахождения системы в состоянии 1:  $p_{\text{отк}} = p_1$ .

Поскольку интенсивность входящих заявок  $\lambda = \lambda' + \lambda''$ , где  $\lambda'$  – интенсивность заявок, поступивших на обслуживание,  $\lambda''$  – интенсивность потерянных заявок, выразим  $\lambda'$  и  $\lambda''$ :

$$\lambda' = \lambda p_0 = \lambda \mu / (\lambda + \mu); \quad \lambda'' = \lambda p_1 = \lambda^2 / (\lambda + \mu).$$

Коэффициент нагрузки системы

$$\zeta = \frac{\lambda}{\mu} = \frac{x_{\text{ср}}}{t_{\text{ср}}}; \quad \rho = 0 \div \infty.$$

Коэффициент загрузки системы

$$\rho = \lambda' / \mu = \rho p_0 = p_1 = \frac{\lambda}{\lambda + \mu}; \quad 0 \leq \rho \leq 1.$$

Зависимость коэффициента загрузки системы от коэффициента нагрузки представлена на рис. 2.7. Очевидно, что с ростом коэффициента нагрузки коэффициент загрузки и вероятность отказа будет стремиться к 1.



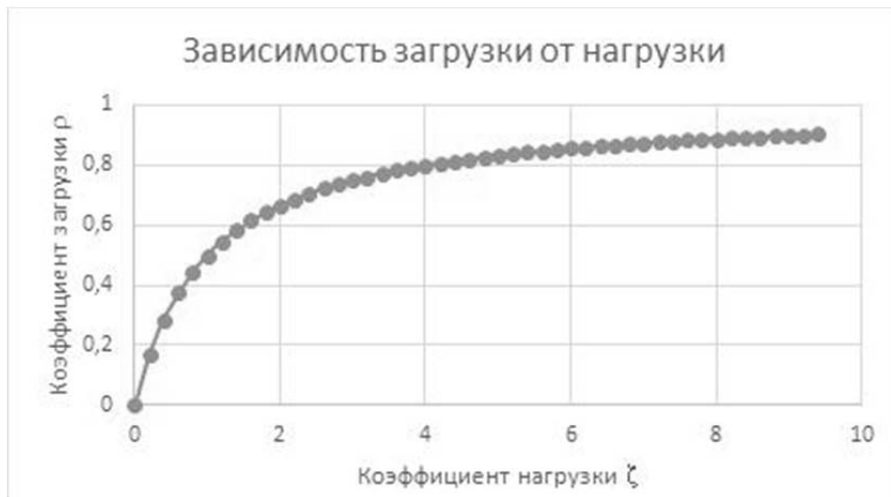


Рис. 2.7. Зависимость коэффициента загрузки от коэффициента нагрузки

### 2.2.2. Многоканальная система с отказами $M/M/n/0$

Система  $M/M/n/0$  (рис. 2.8) характеризуется  $n$  каналами обслуживания и отсутствием накопителя очереди. Это классическая модель телефонной сети, исследованная ещё Эрлангом, называемая **системой с отказами**. Заявка, поступившая в систему и заставшая все приборы занятыми обслуживанием других заявок, получает отказ и удаляется из системы.

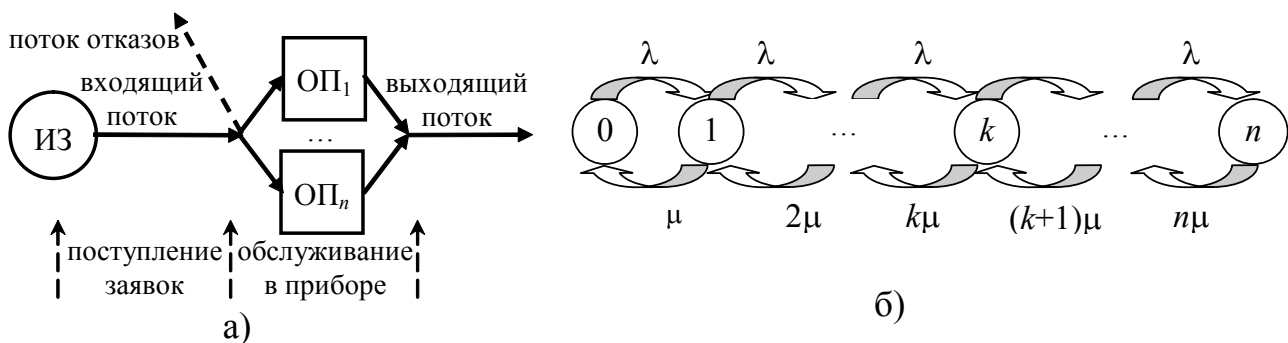


Рис. 2.8. Система  $M/M/n/0$ : а)  $Q$ -схема; б) диаграмма переходов

Определим основные характеристики системы.

1. Вероятность того, что обслуживанием заняты  $k$  приборов

$$p_k = \frac{\zeta^k / k!}{\sum_{i=0}^n \zeta^i / i!}.$$

2. Вероятность того, что все обслуживающие приборы свободны

$$p_0 = \frac{1}{\sum_{i=0}^n \zeta^i / i!}.$$

3. Вероятность отказа (вероятность того, что все обслуживающие каналы заняты)

$$p_n = \frac{\zeta^n / n!}{\sum_{i=0}^n \zeta^i / i!} = p_0 \frac{\zeta^n}{n!}.$$

4. Среднее число свободных от обслуживания приборов

$$N_0 = \sum_{i=0}^{n-1} (n-i) p_i = \sum_{i=0}^{n-1} \frac{\zeta^i (n-i)}{i!} p_0.$$

5. Среднее число занятых обслуживанием приборов

$$N_3 = \sum_{i=1}^n i p_i = \sum_{i=1}^n \frac{\zeta^i}{(i-1)!} p_0 = \zeta(1 - p_n).$$

6. Коэффициент занятости приборов

$$K_3 = N_3 / n.$$

7. Коэффициент простоя приборов

$$K_n = N_0 / n.$$

8. Коэффициент удельной нагрузки

$$\chi = \lambda / n\mu < 1.$$

9. Коэффициент нагрузки

$$\zeta = \lambda / \mu = \lambda t_{\text{обс}} < n.$$

10. Коэффициент загрузки

$$\rho = \frac{\lambda}{\lambda + \mu}; \quad 0 \leq \rho \leq 1.$$

### 2.2.3. Одноканальная система с неограниченной очередью M/M/1

Система M/M/1 (рис. 2.9) является простейшей нетривиальной системой. Интенсивности переходов для всех состояний совпадают:

$$\lambda_k = \lambda, \quad \mu_k = \mu, \quad k = 0, 1, 2 \dots$$

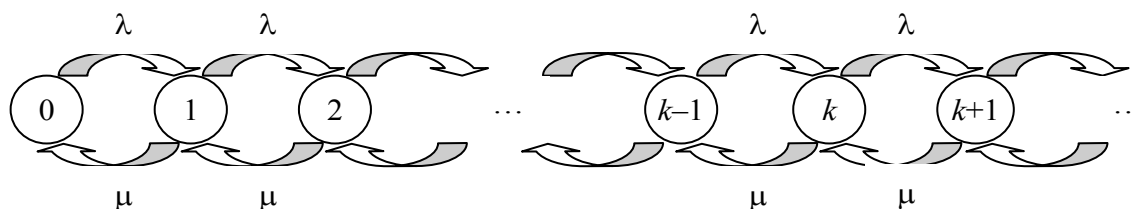


Рис. 2.9. Диаграмма переходов для СМО M/M/1

Пространство состояний в такой системе бесконечно, поскольку ёмкость накопителя очереди не ограничена. Интервалы между входными заявками и интервалы обслуживания обратно пропорциональны интенсивностям:

$$t_{\text{ср}} = \frac{1}{\lambda}; \quad x_{\text{ср}} = \frac{1}{\mu}.$$

Рассмотрим основные характеристики системы M/M/1.

1. Вероятность свободного состояния системы

$$p_0 = \frac{1}{[1 + \sum_{k=1}^{\infty} (\lambda/\mu)^k]} = \frac{1}{[1 + \frac{\lambda/\mu}{1 - \lambda/\mu}]} = 1 - \lambda/\mu, \quad \lambda < \mu.$$

2. Вероятность того, что в системе находится  $k$  требований

$$p_k = (1 - \rho)\rho^k, \quad k = 0, 1, 2, 3.$$

3. Средняя длина очереди

$$L = \frac{\rho^2}{1 - \rho}.$$

4. Среднее число требований в системе

$$N = \frac{\rho}{1 - \rho}.$$

5. Среднее время ожидания в очереди

$$W = \frac{\rho / \mu}{1 - \rho} = \frac{\rho}{\mu - \lambda}.$$

6. Среднее время пребывания в системе

$$T = \frac{1 / \mu}{1 - \rho} = \frac{1}{\mu - \lambda}.$$

7. Коэффициент загрузки (нагрузки, удельной нагрузки) системы

$$\rho = \zeta = \chi = \frac{\lambda}{\mu}.$$

При  $\lambda < \mu$  система устойчива, при  $\lambda \rightarrow \mu$ ,  $p_0 = 0$ ,  $x_{\text{ср}} \rightarrow \infty$

#### 2.2.4. Многоканальная система с неограниченной очередью $M/M/n$

В данной системе имеется  $n$  приборов обслуживания с одинаковой производительностью и заявка, приходящая на обслуживание, выбирает любой свободный из них. Интенсивность поступающих заявок постоянна при любом состоянии системы:  $\lambda_k = \lambda$ ,  $k = 0, 1, 2, \dots$ . При этом интенсивность обслуживания в различных состояниях (рис. 2.10) принимает следующие значения:

$$\mu_k = \min[k\mu, n\mu] = \begin{cases} k\mu, & 0 \leq k \leq n \\ n\mu, & n \leq k \end{cases}.$$

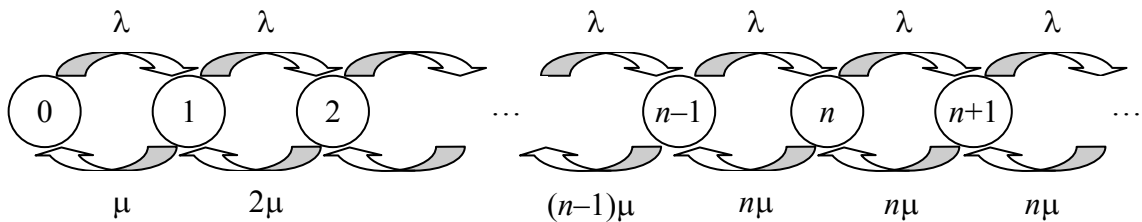


Рис. 2.10. Диаграмма переходов для СМО  $M/M/n$

Рассмотрим основные характеристики системы.

1. Вероятность того, что все обслуживающие приборы свободны

$$p_0 = \frac{1}{\sum_{k=0}^{n-1} \frac{\zeta^k}{k!} + \frac{\zeta^n}{n!(1-\rho)}} = \frac{1}{\sum_{k=0}^{n-1} \frac{\zeta^k}{k!} + \frac{\zeta^n}{(n-1)!(n-\zeta)}}.$$

2. Вероятность того, что в системе находится  $k$  требований

$$p_k = \begin{cases} \frac{\zeta^k}{k!} p_0, & 1 \leq k \leq n; \\ \frac{\zeta^k}{n! n^{k-n}} p_0 = \frac{\chi^k n^n}{n!} p_0, & k \geq n. \end{cases}$$

3. Вероятность того, что все обслуживающие приборы заняты (вероятность ожидания обслуживания приходящим требованием)

$$p_n = \frac{\zeta^n}{n!(1-\chi)} \cdot p_0 = \frac{\zeta^n}{(n-1)!(n-\zeta)} \cdot p_0.$$

4. Среднее число свободных от обслуживания приборов

$$N_0 = \sum_{k=0}^{n-1} (n-k) p_k = \sum_{k=0}^{n-1} \frac{\zeta^k (n-k)}{k!} p_0.$$

5. Среднее число занятых обслуживанием приборов

$$N_3 = n - N_0 = \sum_{k=1}^n k p_k = \sum_{k=1}^n \frac{\zeta^k}{(k-1)!} p_0 = \zeta(1 - p_n).$$

6. Средняя длина очереди

$$L = N - \zeta = \frac{\chi \cdot p_n}{1-\rho} = \frac{\zeta \cdot p_n}{n-\zeta}.$$

7. Среднее число требований в системе

$$N = \zeta + L = \zeta + \frac{\chi \cdot p_n}{1-\rho}.$$

8. Среднее время ожидания в очереди

$$W = \frac{\chi \cdot p_n}{\lambda(1-\chi)} = \frac{p_n}{\mu(n-\zeta)}.$$

9. Среднее время пребывания в системе

$$T = \frac{1}{\mu} + W = \frac{1}{\mu} + \frac{\chi \cdot p_n}{\lambda(1-\chi)}.$$

### 2.2.5. Многоканальная система с ограниченной очередью $M/M/n/m$

Отличие данной системы от предыдущей в том, что количество мест ожидания в очереди ограничено числом  $m$ , и если приходящая в систему заявка застанет все места ожидания занятыми, то она теряется (удаляется из системы). Такие системы часто называют *системами с потерями*.

1. Вероятность того, что все обслуживающие приборы свободны

$$p_0 = \frac{1}{\sum_{k=0}^{n-1} \zeta^k / k! + \frac{\zeta^n \cdot (1-\chi^{m+1})}{n!(1-\chi)}} = \frac{1}{\sum_{k=0}^{n-1} \zeta^k / k! + \frac{\zeta^n \cdot (1-(\zeta/n)^{m+1})}{(n-1)!(n-\zeta)}}.$$

2. Вероятность того, что в системе находится  $k$  требований

$$p_k = \begin{cases} \frac{\zeta^k}{k!} p_0, & 1 \leq k \leq n; \\ \frac{\zeta^k}{n! n^{k-n}} = \frac{\chi^k n^n}{n!}, & n \leq k \leq n+m. \end{cases}$$

3. Вероятность того, что все обслуживающие приборы заняты (вероятность ожидания обслуживания приходящим требованием)

$$p_n = \frac{(1 - \chi^{m+1}) \cdot \zeta^n}{n!(1 - \chi)} \cdot p_0 = \frac{(1 - (\zeta/n)^{m+1}) \cdot \zeta^n}{(n-1)!(n - \zeta)} \cdot p_0.$$

4. Вероятность отказа в постановке в очередь

$$p_n = \frac{\zeta^s}{n! n^{s-n}} \cdot p_0, \quad s = n + m.$$

5. Среднее число свободных от обслуживания приборов

$$N_0 = \sum_{k=0}^{n-1} (n-k) p_k = \sum_{k=0}^{n-1} \frac{\zeta^k (n-k)}{k!} p_0.$$

6. Среднее число занятых обслуживанием приборов

$$N_3 = n - N_0.$$

7. Средняя длина очереди

$$L = \frac{\zeta^n \cdot (\chi - (m+1) \cdot \chi^{m+1} - (m+2) \cdot \chi^{m+2})}{n!(1 - \chi)^2} \cdot p_0.$$

8. Среднее число требований в системе

$$N = \zeta + L.$$

9. Среднее время ожидания в очереди

$$W = \frac{L}{\lambda}.$$

10. Среднее время пребывания в системе

$$T = \frac{1}{\mu} + W.$$

## 2.3. Анализ полумарковских систем массового обслуживания

### 2.3.1. Классификация методов исследования СМО

Методы теории массового обслуживания делятся на *точные* и *приближённые*. Основными точными методами исследования СМО являются:

- 1) метод однородных цепей Маркова;
- 2) метод фаз Эрланга;
- 3) метод последовательно-параллельных фаз;

- 4) метод вложенных цепей Маркова;
- 5) метод дополнительных переменных;
- 6) метод интегральных уравнений Линдли;
- 7) метод введения дополнительных событий.

Рассмотрим подробнее некоторые из точных методов.

**Метод однородных цепей Маркова ( $\Delta t$ -метод)** применяется, если все функции распределений экспоненциальны.

Состояние системы описывается однородной непрерывной Марковской цепью. Вероятности состояний находятся из системы линейных дифференциальных уравнений.

С помощью искусственного приёма  $\Delta t$ -метод применим, когда наряду с экспоненциальными встречаются случайные величины с эрланговским распределением. Дело в том, что случайную величину, имеющую распределение Эрланга порядка  $k$  с параметром  $\lambda$ , можно представить в виде суммы  $k$  экспоненциально распределённых величин с параметром  $\lambda$ . При этом существенно возрастает количество состояний системы.

Этот метод называется **методом фаз Эрланга**. При этом существенно возрастает количество состояний системы.

В системах, где не все распределения экспоненциальны, ищут аналитические приёмы, приводящие интересующие процессы к Марковским. Различают внутреннее и внешнее марковизование.

**Внутреннее марковизование** состоит в том, что для случайного процесса ищут такие моменты времени, в которые процесс становится Марковским. Это **метод вложенных цепей Маркова**.

**Внешнее марковизование** характеризуется тем, что пространство состояний начального немарковского случайного процесса с помощью соответствующих дополнительных переменных расширяют так, чтобы случайный процесс с расширенным фазовым пространством стал Марковским. Это **метод дополнительных переменных**.

Для систем с произвольными распределениями используется **метод интегральных уравнений Линдли**, а также **метод введения дополнительных событий**.

В тех случаях, когда аналитические методы неприменимы или громоздки, используют два способа получения полезных результатов – **приближённые методы** и **метод моделирования** (Монте-Карло).

Использование  **$z$ -преобразований** (для функций дискретного времени) и **преобразований Лапласа** (для функций непрерывного времени) в ТМО значительно упрощает анализ СМО.

Данные преобразования дают возможность разложения функций времени в суммы комплексных показательных функций. Комплексные показательные функции образуют строительные блоки преобразований.

Рассмотрим **функцию дискретного времени**  $f_n$ , принимающую ненулевые значения только при неотрицательных значениях индекса ( $f_n = 0$  при  $n < 0$ ). Необходимо сжать эту бесконечную последовательность в одну функцию таким образом, чтобы потом можно было её снова перевести в начальную последовательность. Для этого каждый член последовательности  $f_n$  отметим, умножая на  $z^n$ , затем просуммируем. Получим производящую функцию или  $z$ -преобразование:

$$F(z) = \sum_{n=0}^{\infty} f_n z^n. \quad (2.2)$$

Производящая функция для последовательности существует, если члены последовательности возрастают не быстрее, чем по закону геометрической прогрессии, т. е. если при некотором  $a > 0$  будет выполняться равенство

$$\lim_{n \rightarrow \infty} \frac{|f_n|}{a^n} = 0.$$

Рассмотрим важные примеры  $z$ -преобразований. Равенство (2.2) можно записать в виде:  $f_n \Leftrightarrow F(z)$ . Двойная стрелка показывает единственность преобразования.

Геометрическая прогрессия

$$A\alpha^n \Leftrightarrow \frac{A}{1 - \alpha z}.$$

Рассмотрим **функцию непрерывного времени**  $f(t)$  (ФНВ), принимающую ненулевое значение только для неотрицательных значений непрерывной переменной  $t - f(t) = 0$  при  $t < 0$ . Желательно преобразовать ФНВ в функцию новой комплексной переменной ( $s$ ), а также иметь возможность получить обратное преобразование в область переменной  $t$ . В качестве такой метки целесообразно применить  $e^{-st}$ . Комплексная переменная в алгебраической форме записывается как  $s = \sigma + j\omega$ , где  $j = \sqrt{-1}$ . Умножив  $f(t)$  на эту метку и проинтегрировав по всем ненулевым значениям, получим преобразование Лапласа:

$$F^*(S) = \int_0^{\infty} f(t) e^{-st} dt. \quad (2.3)$$

Нижний предел воспринимается как 0. Такое преобразование Лапласа существует, если функция  $f(t)$  возрастает не быстрее показательной функции, т. е. если существует действительная постоянная  $c$ , удовлетворяющая условию:

$$\lim_{\tau \rightarrow \infty} \int_0^{\tau} |f(t)| e^{ct} dt < \infty.$$

Рассмотрим важные преобразования Лапласа. Равенство (2.3) можно записать в виде:  $f(t) \Leftrightarrow F^*(s)$ . Двойная стрелка показывает единственность преобразования.

Одно из наиболее важных преобразований – преобразование односторонней показательной функции

$$f(t) = \begin{cases} Ae^{-at}, & t \geq 0 \\ 0, & t < 0 \end{cases}; \quad f(t) \Leftrightarrow F^*(S) = \int_0^{\infty} Ae^{-at} e^{-st} dt = A \int_0^{\infty} e^{-(s+a)t} e^{-st} dt = \frac{A}{s+a}$$

$$\text{или } Ae^{-at} \delta(t) \Leftrightarrow \frac{A}{s+a}, \text{ где } \delta(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}.$$

Соответственно, для дельта-функции  $\delta(t) \Leftrightarrow 1/s$

$$tf(t) \Leftrightarrow -\frac{dF^*(s)}{ds}, \quad t^n f(t) \Leftrightarrow (-1)^n \frac{d^n F^*(s)}{ds^n}.$$

**Пример.** Для СМО  $M/M/1$  преобразования экспоненциальных распределений времени между входными заявками и времени обслуживания имеют вид:

$$\text{для ПР времени между входными заявками } a(t) = \lambda e^{-\lambda t} - A^*(S) = \frac{\lambda}{\lambda + s};$$

$$\text{для ПР времени обслуживания } b(t) = \mu e^{-\mu t} - B^*(S) = \frac{\mu}{\mu + s}.$$

### 2.3.2. Метод фаз Эрланга

Если одно из распределений является неэкспоненциальным, тогда существует принципиальная возможность представить неэкспоненциальный закон в виде марковского процесса, т. е. сохранить свойства потока, но на графе разрешить переход не только в соседние состояния. Свойства потока могут быть описаны с помощью набора экспоненциальных потоков.

Наиболее известным методом исследования систем массового обслуживания с неэкспоненциальными распределениями является **метод фаз Эрланга**. Этот метод разработан Эрлангом и позволяет исследовать СМО общего вида. Суть – распределение времени обслуживания может быть разложено в набор составляющих показательных распределений (рис. 2.11), каждое из этих распределений описывается обычным вероятностным законом, характерным для экспоненты.

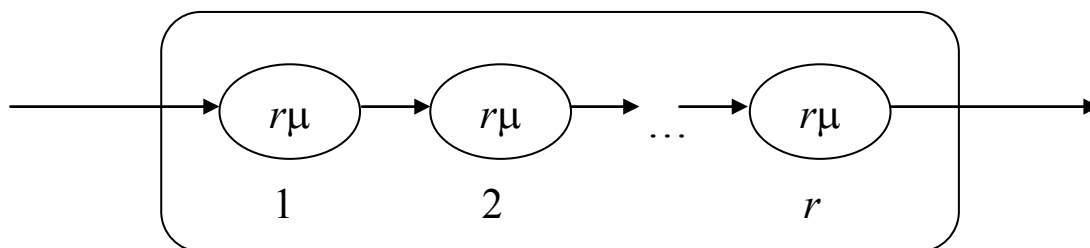


Рис. 2.11. Многофазный эрланговский обслуживающий прибор



Рассмотрим две фазы. Плотность распределения времени обслуживания на каждой из фаз

$$h(y) = 2\mu e^{-2\mu y}.$$

В любой момент времени в обслуживающем устройстве может находиться только одно требование находящегося на первой или второй фазе экспоненциального обслуживания. Преобразование Лапласа–Стилтьеса для одной фазы равно

$$H(s) = \frac{2\mu}{s + 2\mu}.$$

Для двух фаз имеем

$$B^*(s) = [H(s)]^2 = \left( \frac{2\mu}{s + 2\mu} \right)^2.$$

Плотность распределения случайной величины, разбитой на две фазы:

$$b(x) = 2\mu(2\mu x)e^{-2\mu x}, x \geq 0$$

Для  $r$  фаз имеем  $h(y) = r\mu e^{-r\mu y}$

$$b(x) = \frac{r\mu(r\mu x)^{r-1} e^{-r\mu x}}{(r-1)!}; x \geq 0.$$

**Рассмотрим систему  $M/E_r/1$  (рис. 2.12):**

$a(t) = \lambda e^{-\lambda t}, t \geq 0$  – распределение входного потока;

$b(x) = \frac{r\mu(r\mu x)^{r-1} e^{-r\mu x}}{(r-1)!}; x \geq 0$  – распределение времени обслуживания.

Кроме нахождения числа требований, надо найти число фаз, через которые остается пройти обслуживаемому требованию. Особенность метода показана на диаграмме интенсивностей переходов (рис. 2.12).

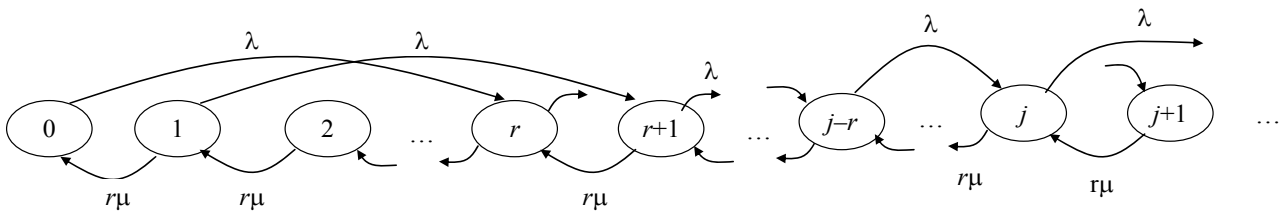


Рис. 2.12. Диаграмма переходов для системы  $M/E_r/1$

Остающееся число фаз определяется в соответствии с выражением

$$j = (k-1)r + (r-i+1) = rk - i + 1,$$

где  $P_j$  – вероятность  $j$  фаз в системе;

$p_k$  – стационарная вероятность наличия требований в системе,

$$p_k = \sum_{j=(k-1)r+1}^{kr} P_j, k = 1, 2, 3 \dots$$

В результате преобразований получим

$$p_k = (1 - \rho)\rho^k, \quad k = 0, 1, 2, \dots, \rho - \text{коэффициент загрузки.}$$

При  $r = 1$ ,  $p_k = P_k$ , следовательно

$$p_0 = 1 - \rho.$$

Для системы распределение числа фаз обслуживания представляет собой взвешенную сумму геометрических распределений.

Рассмотрим систему  $E_r/M/1$  (рис. 2.13):

$$a(t) = \frac{r\lambda(r\lambda t)^{r-1} e^{-r\lambda t}}{(r-1)!}; \quad b(x) = \mu e^{-\mu x}.$$

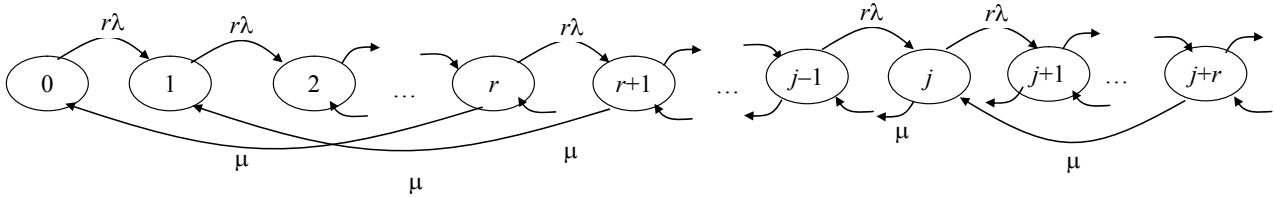


Рис. 2.13. Диаграмма переходов для системы  $E_r/M/1$

Описание состояний такой системы должно содержать не только число требований в системе, но и информацию о том, на какой фазе приёмного устройства находится очередное требование.

Если  $P_j$  – число фаз поступления в систему, а  $p_k$  – стационарная вероятность наличия  $k$ -требований в системе, тогда

$$p_k = \sum_{j=kr}^{r(k+1)-1} P_j;$$

$$P_j = \begin{cases} \frac{1}{r}(1 - z_0^{-j-1}), & 0 \leq j < r; \\ \rho(z_0 - 1)z_0^{r-j-1}, & j \geq r \end{cases};$$

$$p_k = \begin{cases} 1 - \rho, & k = 0 \\ \rho(z_0 - 1)z_0^{r-j-1}, & k > 0 \end{cases},$$

где  $z_0$  – корень уравнения для производящей функции.

Распределение числа требований в системе геометрическое.

На основе метода фаз Эрланга разработан более общий подход – метод последовательно-параллельных фаз.

### 2.3.3. Метод вложенных цепей Маркова

Метод вложенных цепей Маркова относится к стандартным методам теории массового обслуживания. Он был использован впервые в 1951 году Д. Кендаллом при исследовании модели  $M/G/1$ .

Основной смысл метода вложенных цепей Маркова состоит в том, что для данного случайного процесса, описывающего поведение рассматриваемой модели, конструируется удобная для интересующих нас вопросов марковская цепь (рис. 2.14), исследуемая аналитическими методами, обычными для цепей Маркова, и полученные результаты (например, стационарные вероятности состояний) пересчитываются для величин исходной системы [31].

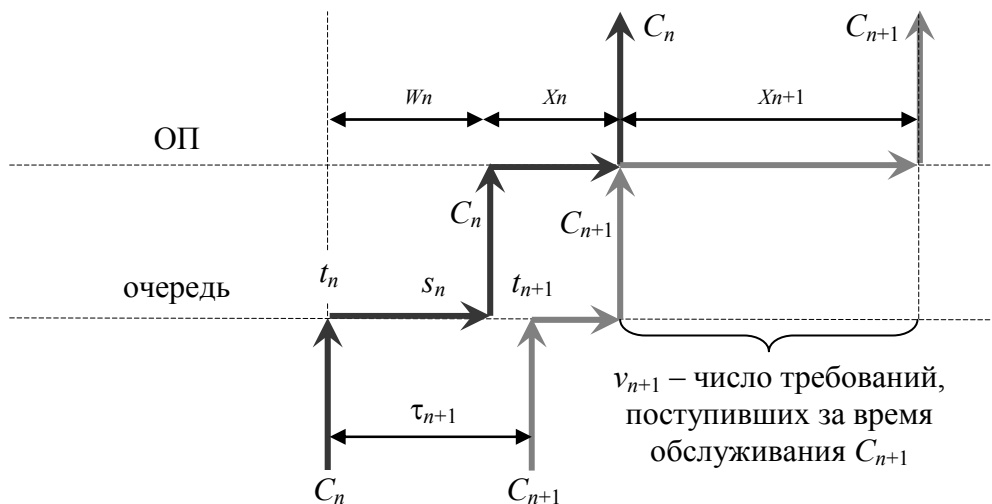


Рис. 2.14. Случайные процессы в системе  $M/G/1$

На данной диаграмме уходящее требование  $C_n$  не оставляет систему пустой.

Число требований, остающихся после ухода требования  $C_{n+1}$

$$q_{n+1} = q_n - 1 + v_{n+1}, \quad q_n > 0.$$

Если  $q_n = 0$ , то имеем следующую диаграмму (рис. 2.15). Объединив два выражения в систему, получим

$$q_{n+1} = \begin{cases} q_n - 1 + v_{n+1}, & q_n > 0; \\ v_{n+1}, & q_n = 0. \end{cases}$$

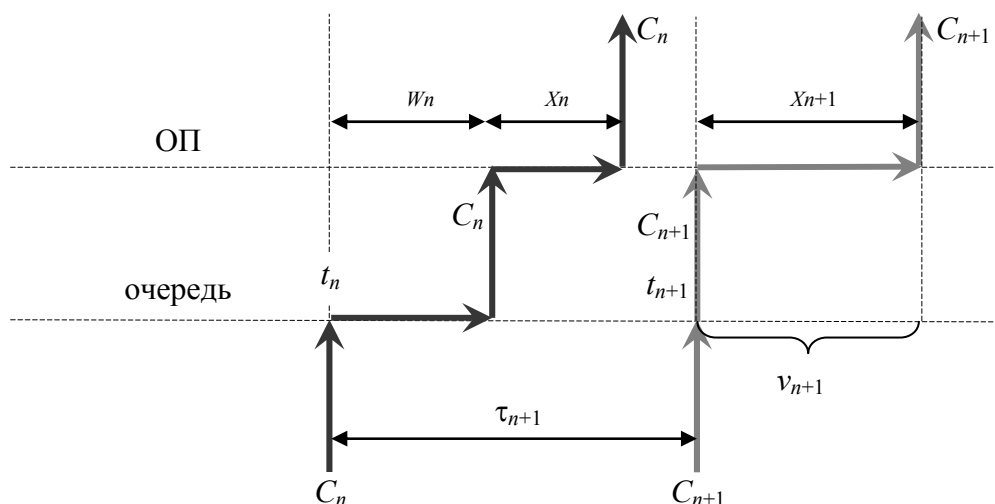


Рис. 2.15. Случайные процессы в системе  $M/G/1$  при  $q_n = 0$

Вероятность того, что в системе находится  $k$  заявок,  $p_k = P[N(t) = k]$ , равна вероятности того, что поступившая заявка застанет в системе  $k$  требований  $r_k$ , и равна вероятности  $d_k$  оставления заявкой  $k$  требований при выходе из системы:

$$p_k = r_k = d_k.$$

**Формула Полячека – Хинчина:**

– средняя длина очереди в момент ухода обслуженного требования

$$\tilde{q} = \rho^2 \frac{1 + C_\beta^2}{2(1 - \rho)};$$

– среднее число требований в системе

$$N = \rho + \rho^2 \frac{1 + C_\beta^2}{2(1 - \rho)};$$

– среднее время ожидания

$$W = \rho \cdot x_{\text{ср}} \frac{1 + C_\beta^2}{2(1 - \rho)};$$

$C_\beta^2 = \frac{D_\beta}{(x_{\text{ср}})^2}$  – квадрат коэффициента вариации времени обслуживания.

**Примеры:**

$$M/M/1: \quad N = \rho + \rho^2 \frac{2}{2(1 - \rho)} = \frac{\rho}{1 - \rho}, \quad W = x_{\text{ср}} \frac{\rho}{1 - \rho};$$

$$M/D/1: \quad N = \rho + \rho^2 \frac{1}{2(1 - \rho)} = \frac{\rho}{1 - \rho} - \frac{\rho^2}{2(1 - \rho)}, \quad W = x_{\text{ср}} \frac{\rho}{2(1 - \rho)}.$$

Отсюда следует интересный вывод: среднее время ожидания в системе  $M/M/1$  в два раза больше, чем в системе  $M/D/1$ .

**Формула Полячека – Хинчина для ПЛС ПР времени ожидания в системе:**

$$W^*(s) = \frac{1 - \rho}{1 - \rho \left[ \frac{1 - B^*(s)}{s x_{\text{ср}}} \right]},$$

где  $B^*(s)$  – ПЛС ПР времени обслуживания.

Таким образом, метод состоит из следующих четырёх шагов:

Шаг 1. Определение некоторой подходящей последовательности моментов времени  $(t_n)$ ,  $0 \leq t_1 < t_2 < \dots < \infty$  позволяющей конструирование вложенной цепи Маркова.

Шаг 2. Вычисление переходных вероятностей для вложенной цепи Маркова.

Шаг 3. Вычисление стационарного распределения или других величин для вложенной цепи Маркова.

Шаг 4. Пересчёт полученных в п. 3 результатов в искомые величины системы, причём при необходимости и для точек, отличных от  $t_n$ .

Шаги 2 и 4 часто довольно сложны и требуют больших вычислений. Для шагов 1 и 4 указывается общее правило, удобное для моделей теории массового обслуживания и теории надёжности.

**Пример:**

$$M/M/1: W^*(S) = (1 - \rho) + \frac{\lambda(1 - \rho)}{s + \mu(1 - \rho)}.$$

**Плотность распределения времени ожидания:**

$$w(y) = (1 - \rho)u_0(y) + \lambda(1 - \rho)e^{-\mu(1 - \rho)y}, y \geq 0.$$

**Функция распределения времени ожидания:**

$$W(y) = 1 - \rho e^{-\mu(1 - \rho)y}, y \geq 0.$$

## 2.4. Анализ систем с произвольными потоками событий

### 2.4.1. Метод интегральных уравнений Линдли

До сих пор рассматривались методы расчёта, использующие марковские свойства систем массового обслуживания. Для более сложных систем типа  $G/G/1$  требуется искать новые методы.

Рассмотрим основные уравнения для системы  $G/G/1$ . В данной системе с одним прибором обслуживания промежутки времени между поступающими требованиями  $t_n$ , а также времени обслуживания  $x_n$ , являются независимыми и имеют произвольные функции распределения  $A(t)$  и  $B(x)$  соответственно. Если  $\tau_n$  – момент времени поступления  $n$ -го требования  $C_n$ , то  $t_n = \tau_n - \tau_{n-1}$ . Время ожидания требования  $C_n$  в очереди обозначим  $w_n$ .

Для упрощения анализа необходимо найти марковский процесс.

Например, марковским процессом для системы  $M/G/1$  будет *незавершённая работа системы*  $U(t)$  в момент времени  $t$  для любых  $t$ .  $U(t)$  – это остаточное время, необходимое для освобождения системы от всех требований, находящихся в ней в момент  $t$ .

Для системы  $G/G/1$   $U(t)$  будет немарковским, но в него вложен марковский процесс, определённый в *моменты поступления требований*. Для систем с дисциплиной FIFO значение незавершённой работы непосредственно перед поступлением требования  $C_n$  в точности равно времени ожидания  $w_n$ .

Рассмотрим временные диаграммы для двух случаев:

1. Очередное требование  $C_{n+1}$  поступает в занятую систему (рис. 2.16).

Тогда  $w_{n+1} = w_n + x_n - t_{n+1}$ , если  $w_n + x_n - t_{n+1} \geq 0$ .

2. Очередное требование  $C_{n+1}$  поступает в свободную систему (рис. 2.17).

Тогда  $w_{n+1} = 0$ , если  $w_n + x_n - t_{n+1} \leq 0$ .

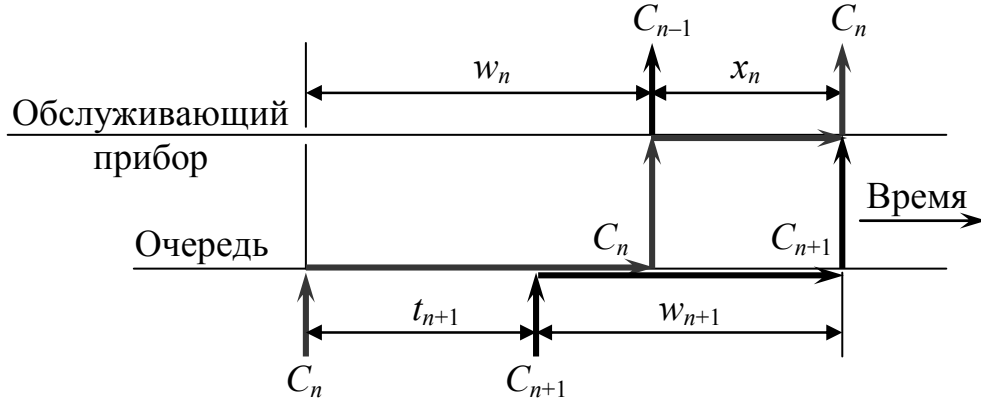


Рис. 2.16. Случай, когда требование  $C_{n+1}$  поступает в занятую систему

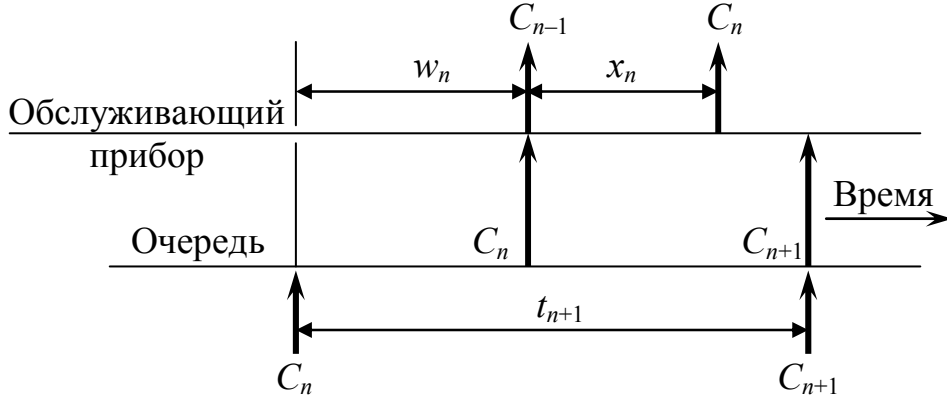


Рис. 2.17. Случай, когда требование  $C_{n+1}$  поступает в свободную систему

Определим новую случайную величину  $u_n = x_n - t_{n+1}$ , или разность между временем обслуживания требования и промежутком времени между поступлениями требований.

Тогда

$$w_{n+1} = \begin{cases} w_n + u_n, & w_n + u_n \geq 0; \\ 0, & w_n + u_n \leq 0. \end{cases}$$

Уравнение можно записать в более компактном виде:

$$w_{n+1} = \max[0, w_n + u_n] \text{ или } w_{n+1} = (w_n + u_n)^+.$$

Поскольку случайные величины  $\{t_n\}$  и  $\{x_n\}$  независимы, то последовательность случайных величин  $\{w_0, w_1, w_2, \dots\}$  образует марковский процесс со стационарными вероятностями перехода. Величина  $w_{n+1}$  зависит от предыдущей последовательности величин  $w_i$  только через последнюю величину  $w_n$  плюс случайную величину  $u_n$ , не зависящую от  $w_n$ .

Определим стационарную функцию распределения для  $w_n$ :

$$\lim_{n \rightarrow \infty} P[w_n \leq y] = W(y).$$

Для вычисления  $W(y)$  рассмотрим функцию распределения  $C_n(u)$  случайной величины  $u_n$ :

$$C_n(u) = P[u_n = x_n - t_{n+1} \leq u].$$

Выразим  $C_n(u)$  через  $A(t)$  и  $B(x)$ :

$$C_n(u) = P[x_n - t_{n+1} \leq u] = \int_0^{\infty} P[x_n \leq u + t_{n+1} | t_{n+1} = t] dA(t).$$

Однако время обслуживания требования  $C_n$  не зависит от  $t_{n+1}$ , значит

$$C(u) = \int_0^{\infty} B(u + t) dA(t).$$

Применим обозначение свёртки  $\odot$  и определим  $c_n(u) = dC_n(u)/du$ , тогда

$$c_n(u) = c(u) = a(-u) \odot b(u).$$

Определим ФР времени ожидания требования  $C_n$  в виде

$$W_n(y) = P[w_n \leq y].$$

Для  $y \geq 0$  получаем

$$W_{n+1}(y) = P[w_n + u_n \leq y] = \int_{0-}^{\infty} P[u_n \leq y - w | w_n = w] dW_n(w).$$

Поскольку  $u_n$  не зависит от  $w_n$ , получим

$$W_{n+1}(y) = \int_{0-}^{\infty} C_n(y - w) dW_n(w) \text{ при } y \geq 0.$$

Это распределение имеет предел

$$W(y) = \int_{0-}^{\infty} C_n(y - w) dW(w) \text{ при } y \geq 0.$$

Однако для  $y < 0$   $W(y) = 0$ .

Объединив эти неравенства, получим **интегральное уравнение Линдли**, которое является уравнением типа Винера – Хопфа:

$$W(y) = \begin{cases} \int_{0-}^{\infty} C(y - w) dW(w), & y \geq 0; \\ 0, & y < 0. \end{cases} \quad (2.4)$$

Данное выражение можно записать в другом виде

$$W(y) = \begin{cases} \int_{0-}^{\infty} W(w) dC(y - w), & y \geq 0; \\ 0, & y < 0. \end{cases} \quad (2.5)$$

или

$$W(y) = \begin{cases} \int_{-\infty}^y W(y - u) dC(u), & y \geq 0; \\ 0, & y < 0. \end{cases} \quad (2.6)$$

Для решения данного интегрального уравнения Линдли предложил *спектральное решение*. Спектральное разложение интегрального уравнения Линдли состоит в том, чтобы для полученного выражения найти подходящее представление в виде произведения двух множителей.

Возьмём за основу уравнение (2.6). Его можно было бы решить непосредственно, если бы справа стояла настоящая *свёртка*. У нас же имеется свёртка только для неотрицательных  $y$ . Поэтому используем следующий приём – определим «дополнительное» время ожидания, дополняющее (2.6) до настоящей свёртки:

$$W_-(y) = \begin{cases} 0, & y \geq 0; \\ \int_{-\infty}^y W(y-u) dC(u), & y < 0. \end{cases} \quad (2.7)$$

Левую часть уравнения (2.6) по аналогии обозначим  $W_+(y)$ . Сложив оба уравнения (2.6) и (2.7), получим справа интегральное выражение для всех  $y$ :

$$W_+(y) + W_-(y) = \int_{-\infty}^y W(y-u) dC(u) = \int_{-\infty}^y W(y-u) c(u) du, \text{ для } \forall y \quad (2.8)$$

где  $c(u) = \frac{dC(u)}{d(u)}$  – плотность распределения величины  $u = y - w$ .

Введём для  $W_-(y)$  и  $W_+(y)$  преобразования Лапласа:

$$\Phi_-(s) = \int_{-\infty}^{0^-} W_-(y) e^{-sy} dy, \quad \Phi_+(s) = \int_{0^-}^{\infty} W_+(y) e^{-sy} dy.$$

Преобразования Лапласа для функций и плотностей распределения связаны следующим образом:

$$s\Phi_+(s) = W^*(s),$$

где  $W^*(s)$  – преобразование Лапласа плотности распределения.

Выражение (2.8) в преобразованиях Лапласа запишем как

$$\Phi_+(s) + \Phi_-(s) = \Phi_+(s) C^*(s), \quad (2.9)$$

где  $C^*(s)$  – преобразование Лапласа ФР  $C(u)$ :

$$C^*(s) = A^*(-s) B^*(s),$$

откуда получаем

$$\frac{\Phi_-(s)}{\Phi_+(s)} = A^*(-s) B^*(s) - 1.$$

Задача спектрального разложения – найти для выражения

$$A^*(-s) B^*(s) - 1 \quad (2.10)$$

подходящее представление в виде произведения двух множителей. Из метода фаз Эрланга следует, что преобразования Лапласа фазовых распределений представляют собой рациональные функции от  $s$ . Аналогично можно попытаться аппроксимировать каждое из преобразований Лапласа  $A^*(-s)$  и  $B(s)$  рациональными функциями, тогда (2.10) также будет рациональной функцией  $s$ . В этом случае можно предложить следующее спектральное разложение:

$$A^*(-s) B^*(s) - 1 = \frac{\Psi_+(s)}{\Psi_-(s)}. \quad (2.11)$$



Отношение  $\Psi_+(s)/\Psi_-(s)$  будет некоторой рациональной функцией от  $s$ , для которой нужно найти разложение на множители.

Для  $Re(s) > 0$  функция  $\Psi_+(s)$  должна быть аналитической без нулей в этой полуплоскости, причём

$$\lim_{|s| \rightarrow \infty} \frac{\Psi_+(s)}{s} = 1. \quad (2.12)$$

Для  $Re(s) < D$  функция  $\Psi_-(s)$  должна быть аналитической без нулей в этой полуплоскости, причём

$$\lim_{|s| \rightarrow \infty} \frac{\Psi_-(s)}{s} = -1. \quad (2.13)$$

Условия (2.12) и (2.13) в пределе должны иметь противоположную полярность, так как при  $s \rightarrow \infty$  по мнимой оси обе функции  $A^*(-s)$  и  $B(s)$  должны стремиться к нулю и левая часть стремится к  $-1$ .

Если такое разложение будет найдено, то

$$\Phi_-(s)\Psi_-(s) = \Phi_+(s)\Psi_+(s) = K, \quad (2.14)$$

где  $K$  – константа, равная вероятности того, что поступающее требование застанет систему свободной,

$$K = \lim_{s \rightarrow \infty} \int_0^\infty e^{-sx} W\left(\frac{x}{s}\right) dx = W(0^+).$$

Однако существуют решения для уравнения (2.11) в общем случае, где левая часть не является рациональной функцией, хотя и удовлетворяет условиям аналитичности [19].

#### 2.4.2. Приближённые методы расчёта систем

Для получения оценки выходных характеристик при неопределённых исходных данных возникает проблема, так как нет однозначной связи между входными и выходными характеристиками. Кроме того, модели массового обслуживания могут быть сложны для аналитического решения.

Существует ряд способов преодоления проблем сложности моделей и неопределённости исходных данных:

- 1) **аппроксимация** исходных распределений по начальным моментам;
- 2) исследование системы при большой нагрузке методами **диффузной аппроксимации**;
- 3) нахождение приближённых оценок для начальных моментов периода простоя;
- 4) нахождение строгих **верхних и нижних границ** для времени ожидания или периода простоя;
- 5) использование **эвристических оценок** на основе анализа численных результатов;

б) замена исходных *непрерывных* случайных величин *дискретными* с целью получения прямого аналитического решения.

Наиболее известными в приближённых методах являются **эвристические формулы** Крамера и Лангенбаха – Бельца для системы  $G/G/1$ .

Среднее время ожидания заявкой начала обслуживания

$$w_1 = \frac{\rho \beta_1}{2(1-\rho)} (v_\alpha^2 + v_\beta^2) q(\rho, v_\alpha^2, v_\beta^2),$$

$v_\alpha$  – коэффициент вариации входного потока;

$v_\beta$  – коэффициент вариации потока обслуживания;

$\beta_1$  – среднее время обслуживания;

$q$  – поправочный коэффициент.

Поправочный коэффициент

$$q(\rho, v_\alpha, v_\beta) = \begin{cases} \exp\left(-\frac{2(1-\rho)}{3\rho} \frac{(1-v_\alpha^2)^2}{v_\alpha + v_\beta}\right), & v \leq 1. \end{cases}$$

Также известна формула Кюна для коэффициента вариации выходящего потока системы  $G/G/1$ :

$$v_d^2 = v_\alpha^2 + 2\rho v_\beta^2 - \rho^2 (v_\alpha^2 + v_\beta^2) q(\rho, v_\alpha, v_\beta).$$

В качестве верхней оценки для среднего времени ожидания можно использовать формулу Марчелла:

$$w_1^+ = \frac{1 + v_\beta^2}{(1/\rho)^2 + v_\beta^2} - \frac{D_\alpha + D_\beta}{2\alpha_1(1-\rho)},$$

где  $D_\alpha$  – дисперсия входного потока;

$D_\beta$  – дисперсия потока обслуживания;

$\alpha_1$  – среднее время между входящими заявками.

Достаточно простой и точной оценки *нижней границы* времени ожидания не найдено. Если распределение времени между заявками является распределением с возрастающей интенсивностью наступления событий, то нижнюю границу можно оценить по формуле

$$w_1^- = \frac{D_\alpha + D_\beta}{2\alpha_1(1-\rho)} - \frac{\alpha_1(1+\rho)}{2}.$$

Эта же формула при убывающей интенсивности наступления события входящего потока превращается в верхнюю границу, тогда в качестве нижней границы используется среднее время ожидания системы  $M/G/1$  или  $GI/M/1$ .

В общем случае имеющаяся информация об исходных распределениях **не является достаточной** для определения единственного значения выходных

характеристик. За исключением, когда система близка к системе  $M/G/1$ , которая **безразлична (инвариантна) к третьему моменту**.

Кроме того, широкое распространение получили различные аппроксимационные методы. Суть аппроксимационных методов – в приближённом описании исходных распределений с помощью других хорошо изученных распределений.

Существует несколько способов приближённого описания распределений с использованием различных видов **аппроксимации**.

1. Гиперэкспоненциальная аппроксимация:

$$f(t) = \sum_{i=1}^n C_i \lambda_i e^{-\lambda_i t}.$$

Частный случай – при  $n = 2$ :  $f(t) = C_1 \lambda_1 e^{-\lambda_1 t} + C_2 \lambda_2 e^{-\lambda_2 t}$ .

2. Гипоэкспоненциальная аппроксимация:

$$f(t) = \sum_{i=1}^n (-1)^{i-1} C_i \lambda_i e^{-\lambda_i t}.$$

Частный случай – при  $n = 2$ :  $f(t) = C_1 \lambda_1 e^{-\lambda_1 t} - C_2 \lambda_2 e^{-\lambda_2 t}$ .

3. Обобщённое распределение Эрланга:

$$f(t) = \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2} (e^{-\lambda_1 t} - e^{-\lambda_2 t}).$$

4. Простое распределение Эрланга:

$$f(t) = \lambda^2 e^{-\lambda t}.$$

5. Распределение Кокса:

$$C_1 \lambda_1 e^{-\lambda_1 t} + \frac{(1 - e_1) \lambda_1 \lambda_2}{\lambda_2 - \lambda_1} (e^{-\lambda_1 t} - e^{-\lambda_2 t}).$$

## 2.5. Основы теории сетей массового обслуживания

### 2.5.1. Основные понятия и классификация сетей массового обслуживания

В отличие от СМО СеМО представляет собой более сложное образование, состоящее из отдельных систем массового обслуживания, называемых узлами сети, которые взаимодействуют между собой с помощью некоторого вероятностного механизма.

**СеМО** представляет собой совокупность систем массового обслуживания (**узлов сети**, связанных между собой) и **потоков заявок**, циркулирующих от одной системы или внешнего источника к другой системе или за пределы сети.

Внешний источник заявок называют **истоком**. Узел, через который заявки покидают сеть – **стоком**.

Для задания модели СеМО необходимо определить:

- число узлов, входящих в состав сети  $(S_1, S_2, \dots, S_n)$ ,  $S_0$  – исток,  $S_{n+1}$  – сток;
- число приборов каждого узла  $(m_1, m_2, \dots, m_n)$  ( $m_i = 1$  – простой случай);
- матрицу вероятностей передач между узлами  $R = [r_{ij}]$  (вероятности того, что заявка, покидающая систему  $S_i$ , поступит в систему  $S_j$ );
- характеристики источников входного потока заявок;
- характеристики обслуживания узлов сети.

Вероятность покидания системы заявкой  $q_i = 1 - \sum_{j=1}^N r_{ij}$ ,  $r_{i,n} = 1$  – вероятность

покидания системы заявкой равна единице.

Графически сеть изображается в виде графа, где нулевой узел – исток (заявок), а  $n+1$ -й – сток (выходящий узел). Исток имеет только выход, сток – только вход.

При ненулевой вероятности перехода между  $i$  и  $j$  узлами эти узлы соединяются стрелками, над стрелками пишется вероятность перехода (рис. 2.18). Сумма вероятностей переходов из узла должна быть равна 1:

$$r_{01} + r_{02} = 1; r_{13} + r_{15} = 1; \dots$$

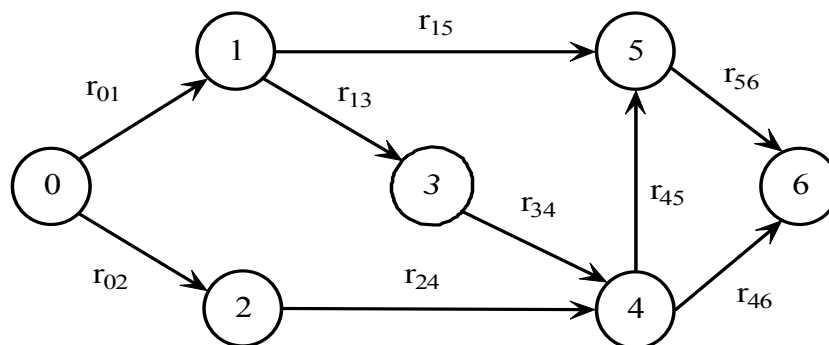


Рис. 2.18. Сеть массового обслуживания

**СеМО** ставится в соответствие реальной сетевой системе (вычислительной сети, сети связи), где вершинам соответствуют узлы обработки информации либо коммуникационные узлы, а рёбрам – каналы связи.

Рассмотрим более подробно классификацию СеМО (рис. 2.19).

По числу циркулирующих в сети заявок различают разомкнутые, замкнутые и смешанные СеМО.

Сеть называется **разомкнутой**, если заявки поступают в сеть извне и после окончания обслуживания покидают её.

Сеть называется **замкнутой**, если в ней нет поступления заявок извне и уходов из сети. Таким образом в сети циркулирует постоянное число заявок.

Сеть называется **однородной**, если все заявки принадлежат только одному классу.

Сеть называется **неоднородной**, если имеется несколько классов заявок.

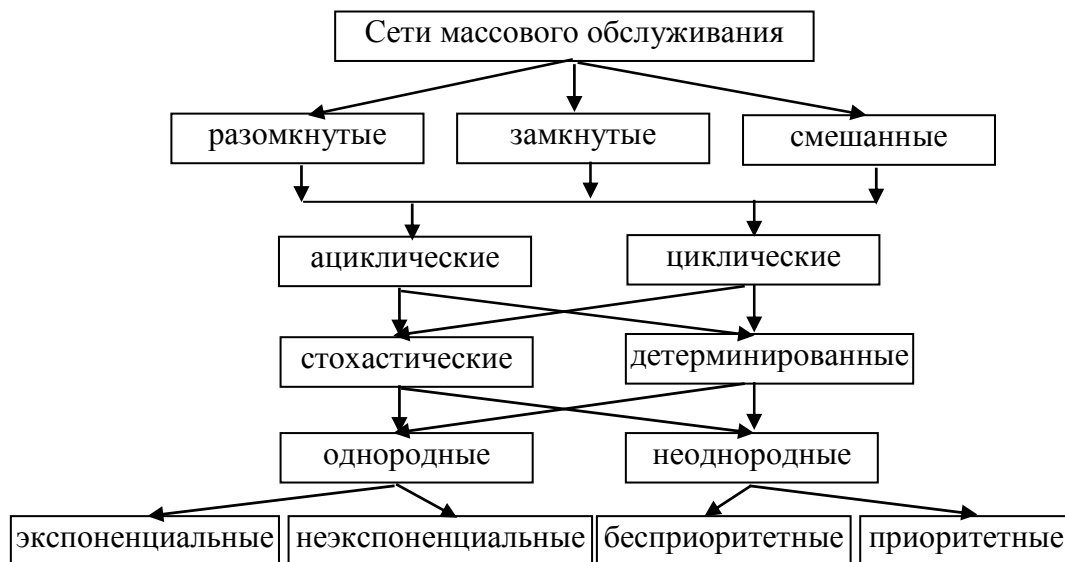


Рис. 2.19. Классификация сетей массового обслуживания

Сеть называется **смешанной**, если для различных классов заявок она выступает либо замкнутой, либо разомкнутой.

Сеть называется **ациклической**, если она не содержит петель обратной связи.

Сеть называется **циклической**, если она содержит петли обратной связи.

Понятие **открытой (разомкнутой) сети** массового обслуживания впервые ввёл в рассмотрение в 1957 году Дж.Р. Джексон, а в 1967 году В.Дж. Гордон и Дж.Ф. Ньюэлл ввели аналогичное понятие **замкнутой сети**.

Результаты Джексона и Гордона–Ньюэлла не использовались до тех пор, пока в 1971 году Ф.Р. Мур не обнаружил, что замкнутые сети адекватно описывают вычислительные системы со многими ресурсами.

С этого момента теория сетей обслуживания стала быстро развиваться благодаря задачам, связанным с математическим моделированием мультипрограммных вычислительных систем и анализом их производительности, с проектированием и анализом СПД и сетей ЭВМ [16, 26, 29, 60].

### 2.5.2. Экспоненциальные сети обслуживания

Простейшими сетями массового обслуживания являются экспоненциальные сети, имеющие пуассоновские стационарные входящие потоки и распространённые по экспоненте длительности обслуживания в различных узлах сети.

Использование гипотезы о простых входящих потоках и экспоненциальном обслуживании приводит к приближённому описанию реальной сети. Практика показывает, что среднее значение характеристик отличается обычно не более чем на 10–15% от реальных. Чем сложнее структура сети, чем больше в ней связей между узлами, тем точнее получаемые результаты. Это следствие из свойства суммирования потоков, поскольку существует теорема, которая показывает, что суммирование потоков при увеличении числа потоков приводит к потоку, близкому к простому.

Полная интенсивность потока  $\lambda_i = \gamma_i + \sum_{j=1}^N \lambda_j r_{ji}$ .

Условием существования стационарного режима в сети является  $\lambda_i < m_i \mu_i$ .

Внутренний поток поступлений в узел  $i$  остаётся пуассоновским, если сеть является ациклической, т. е. не содержит петель обратной связи:

$$q_i = 1 - \sum_{j=1}^N r_{ij}.$$

Выходные потоки из экспоненциальных узлов обслуживания имеют такое же экспоненциальное распределение промежутков времени между уходящими требованиями, что и во входном потоке:

$$\lambda_{\text{вх}i} = \lambda_{\text{вых}i}; D_i(t) = A_i(t) = 1 - e^{-\lambda_i t}; D^*(S) = \frac{\lambda}{s + \lambda}.$$

Дж.Р. Джексон доказал, что каждый узел сети ведёт себя так, как если бы он был независимой системой типа  $M/M/m$ . Если  $k_i$  – число требований в  $i$ -м узле, то состояние системы с  $N$  узлами описывается вектором  $(k_1, k_2, \dots, k_N)$ . Тогда совместное распределение по всем узлам разлагается в произведение маргинальных распределений (**теорема Джексона**):

$$p(k_1, k_2, \dots, k_N) = p_1(k_1) \dots p_N(k_N),$$

где  $p_i(k_i)$  – вероятность того что состояние равновесия в  $i$ -м узле будет находиться  $k_i$  число требований (заявок).

Для Марковских сетей составление равновесия решения получается в виде **произведения**. Замкнутая сеть отличается от разомкнутой внешним поступлением и уходом из сети, хотя перемещение по сети осуществляется в соответствии с матрицей переходов:

$$q_i = 1 - \sum_{j=1}^N r_{ij} = 0.$$

В общем случае решение в виде произведения не получится, так как существует зависимость

$$\sum_{i=1}^N k_i = K; p(k_1, k_2, \dots, k_n) = \frac{1}{G(K)} \prod_{i=1}^N \frac{x_i^{k_i}}{\beta_i(k_i)},$$

где  $G(K)$  – нормировочная константа,

$$G(K) = \sum_{k \in A} \prod_{i=1}^N \frac{x_i^{k_i}}{\beta_i(k_i)}; \beta_i(k_i) = \begin{cases} k_i!, k_i \leq m_i \\ m_i! \cdot m_i^{k_i - m_i}, k_i \geq m_i \end{cases}$$

Суммирование производится по всем векторам состояний, для которых выполняется условие:

$$\sum_{i=1}^N k_i = K; \mu_i x_i = \sum_{j=1}^N \mu_j x_j r_{ji} - \text{отсюда находится } X.$$

Были определены четыре типа систем обслуживания, сохранивших мультипликативные свойства:

1.  $M/M/1$ , FIFO;
2.  $M/G/\infty$ ;
3.  $M/G/1$ , EPS;
4.  $M/G/1$ , LIFO.

Для всех этих систем справедливы следующие формулы:

$$p_i(n_i) = (1 - \rho_i) \cdot \rho_i^{n_i}; \quad p_i(n_i) = \frac{\rho_i^{n_i}}{n!} \cdot e^{-\rho_i}.$$

### Распределение времени пребывания в сети

Пуассоновское свойство выходящего потока и предположение об отсутствии циклов обеспечивают полностью пуассоновский характер процессов по любым дугам сети. При этом предполагается, что любые два узла сети могут быть соединены не более чем одним направленным маршрутом.

Исключается возможность обгона требований требованиями, поступающими позднее, но проходящими по более короткому маршруту.

Преобразование Лапласа – Стильтеса (ПЛС) времени пребывания в  $i$ -м узле:

$$G_i^*(s) = \frac{\mu_i(1 - \rho)}{s + \mu(1 - \rho)} = \frac{\xi_i}{s + \xi_i};$$

$$T_i^*(s) = \left( q_i + \sum_{j=1}^N r_{ij} \cdot T_j^*(s) \right) = G_i^*(s); \quad \frac{s + \xi_i}{\xi_i} \cdot T_i^*(s) = q_i + \sum_{j=1}^N r_{ij} \cdot T_j^*(s);$$

$G(s) = D(s) - R$ , где  $R$  – матрица вероятностей переходов;

$T$  – длительность пребывания в сети произвольного требования;

$T_i$  – длительность пребывания в сети требования, поступающего на узел  $i$ ;

$D(s)$  – диагональная матрица,  $i$ -й элемент которой равен  $\frac{s + \xi_i}{\xi_i}$ ;

$G(s) = (I - R[D(s)]^{-1}) \cdot D(s)$ , где  $I$  – диагональная единичная матрица;

$$G(s) \cdot T^*(s) = q; \quad T^*(s) = \frac{\gamma_i}{\sum_{i=1}^N \gamma_i} \cdot [G(s)]^{-1} = q.$$

Начальные моменты и среднее время пребывания определяется исходя из следующих формул:

$$\tau^{(j)} = \lim_{s \rightarrow 0} (-1)^j \cdot \frac{dT_i^*(s)}{dS^j}; \quad T_{cp} = \sum_{i=1}^N (\gamma_i)^{-1} \cdot \sum_{i=1}^N \frac{\rho_i}{1 - \rho_i};$$

$$\tau^{(2)} = 2 \cdot \left( \sum_{i=1}^N \gamma_i \right)^{-1} \cdot \sum_{i=1}^N \frac{\rho_i}{1 - \rho_i} \cdot \tau_{i1}.$$

### 2.5.3. Преобразование потоков в сетях массового обслуживания

На узел сети заявки могут поступать как из внешнего источника, так и от других узлов сети. В свою очередь, выходящие из узла заявки поступают на другие узлы либо покидают сеть. При этом происходит преобразование исходных потоков заявок – их разделение (прореживание) и наложение (суммирование). В общем случае преобразованные потоки между узлами сети массового обслуживания не являются рекуррентными, однако при некоторых допущениях замена их рекуррентными потоками не приводит к существенным ошибкам в расчёте временных характеристик сети. Аппроксимация реальных потоков рекуррентными широко используется в теории восстановления и в теории массового обслуживания.

Разделение потока на несколько составляющих выполняется с помощью операции случайного просеивания потока, когда заявки с некоторой вероятностью  $p_j$  поступают на  $j$ -й узел или покидают сеть. Характеристики просеянного потока определяются довольно просто. Для  $j$ -го потока в разделяемом потоке имеет место формула для ПЛС плотности распределения времени между заявками:

$$F_j^*(S) = \frac{p_j F^*(S)}{1 - (1 - p_j) F^*(S)},$$

где  $p_j$  – вероятность передачи заявки из разделяемого потока в  $j$ -й узел;

$F^*(s)$  – преобразование Лапласа – Стильтеса плотности распределения времени между заявками в разделяемом потоке.

Начальные моменты  $k$ -го порядка для времени между заявками, поступающими на  $j$ -й узел

$$v_{k,j} = \frac{v_k + \sum_{i=1}^{k-1} v_{k-i} v_{i,j} (1 - p_j)}{p_j},$$

где  $v_k$  – начальные моменты разделяемого потока.

Для квадрата коэффициента вариации  $j$ -го потока имеет место формула

$$v_j^2 = 1 - p_j (1 - v^2).$$

*Коэффициент вариации* выражает степень случайности к промежуткам времени между событиями:

$$v = \sqrt{\frac{v_2 - v_1^2}{v_1^2}} = \sqrt{\frac{v_2}{v_1^2} - 1}.$$

Для экспоненциального закона распределения  $v = 1$ . Если  $v < 1$  – поток более детерминирован, если  $v > 1$  – поток более случаен.

Более трудным является получение характеристик суммарного потока заявок при наложении нескольких рекуррентных потоков. Существующие подходы рассматривают либо суммирование пуассоновских потоков, либо суммирова-



ние только двух потоков с одинаковыми законами распределений (например, гиперэкспоненциальными или эрланговскими). В последнем случае, если число потоков больше двух, суммирование потоков проводится попарно по схеме двоичного дерева с пересчётом на каждом шаге параметров аппроксимации (рис. 2.20, а). Это увеличивает трудоёмкость расчётов и снижает их точность.

Предлагаемая ниже методика позволяет суммировать одновременно сразу несколько рекуррентных потоков (рис. 2.20, б). При этом аппроксимирующими распределениями могут быть выбраны как гиперэкспоненциальные, так и обобщённые эрланговские распределения с произвольным числом фаз и в произвольном сочетании в одном суммируемом потоке [21].

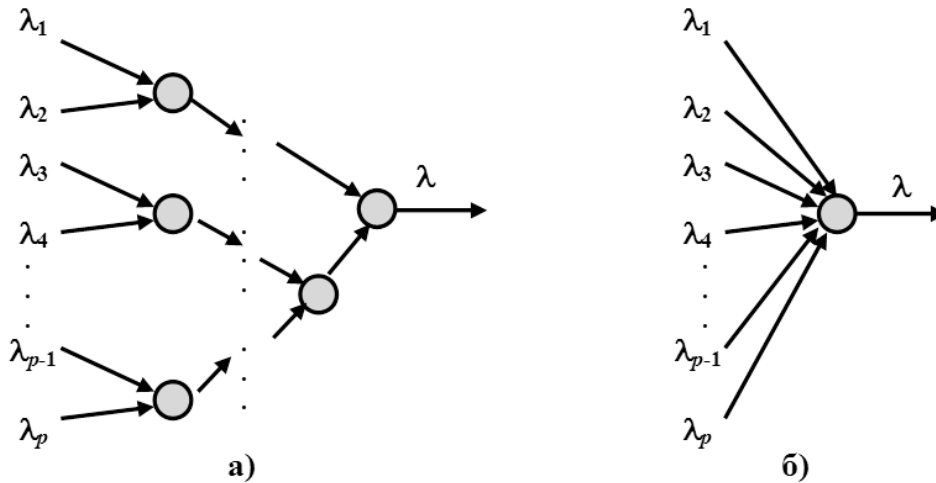


Рис. 2.20. Схемы суммирования рекуррентных потоков:  
а – схема двоичного дерева; б – схема одновременного суммирования

Известно следующее соотношение [8]:

$$\bar{H}(t) = \prod_{i=1}^p \bar{H}_i(t), \quad (2.15)$$

где  $\bar{H}(t)$  и  $\bar{H}_i(t)$  – дополнительные функции распределения (ФР) длительности промежутка времени от произвольного момента до момента поступления последующего требования из объединённого и  $i$ -го исходного потоков;

$p$  – число суммируемых потоков.

Для рекуррентного потока  $\bar{H}(t)$  и  $\bar{H}_i(t)$  выражаются через ФР времени между соседними требованиями  $A(t)$  и  $A_i(t)$  следующими формулами:

$$\bar{H}(t) = 1 - \lambda \int_0^t [1 - A(u)] du; \quad \bar{H}_i(t) = 1 - \lambda_i \int_0^t [1 - A_i(u)] du, \quad (2.16)$$

где  $\lambda = 1/v_1$  и  $\lambda_i = 1/v_{1i}$  – интенсивности поступления требований в объединённом и  $i$ -м суммируемом потоке, обратные первым начальным моментам соответствующего распределения, причём  $\lambda = \sum_{i=1}^p \lambda_i$ .

Подставляя формулы (2.16) в выражение (2.15) и дифференцируя его, получим

$$\bar{A}(t) = \sum_{k=1}^p \frac{\lambda_i}{\lambda} \bar{A}_k(t) \prod_{\substack{i=1 \\ i \neq k}}^p \bar{H}_i(t).$$

Выражение для плотности распределения времени между требованиями в суммарном потоке согласно [9] будет иметь вид

$$a(t) = \frac{d^2}{dt^2} \left[ \prod_{i=1}^p \frac{\lambda_i}{\lambda} \int_t^\infty \int_u^\infty a_i(x) dx du \right]. \quad (2.17)$$

После дифференцирования (2.17) имеем

$$a(t) = \prod_{i=1}^p \frac{\lambda_i}{\lambda} \sum_{j=1}^p \left[ S_j''(t) \prod_{\substack{k=1 \\ k \neq j}}^p S_k''(t) + S_j'(t) \sum_{\substack{k=1 \\ k \neq j}}^p \left( S_k'(t) \prod_{\substack{l=1 \\ l \neq j, l \neq k}}^p S_l(t) \right) \right], \quad (2.18)$$

где  $S_j(t) = \int_t^\infty \int_u^\infty a_i(x) dx du$ ;  $S_j'(t) = -\int_t^\infty a_i(u) du$ ;  $S_j''(t) = a_i(t)$ .

На основе (2.18) получим расчётные формулы для начальных моментов объединённого потока, подставив вместо  $a_i(t)$  соответствующие аппроксимирующие плотности распределений. В качестве последних можно использовать различные типы фазовых распределений. Параметры аппроксимации в общем случае могут быть комплексными. Для выполнения аппроксимаций и последующего суммирования потоков без задействования комплексных параметров достаточно использовать два типа фазовых распределений в зависимости от их коэффициента вариации.

Выделим основные этапы методики.

1. Аппроксимация  $a_i(t)$  суммируемых потоков одним из фазовых распределений, например, в зависимости от коэффициента вариации  $V$  соответствующего распределения.

Если  $v \geq 1$ , будем использовать гиперэкспоненциальную аппроксимацию:

$$a_i(t) = \sum_{j=1}^{Q_i} C_{ji} \lambda_{ji} e^{-\lambda_{ji} t}, \quad (2.19)$$

где  $C_{ji}$ ,  $\lambda_{ji}$  –  $j$ -е параметры разложения  $i$ -го потока.

Если же  $v < 1$ , то будем использовать для аппроксимации  $a_i(t)$  обобщённое распределение Эрланга с различными интенсивностями  $\lambda_{ji}$  последовательных фаз:

$$a_i(t) = \prod_{m=1}^{Q_i} \lambda_{mi} \sum_{j=1}^{Q_i} \frac{e^{-\lambda_{ji} t}}{p(\lambda_{ji})}, \quad \lambda_{ki} \neq \lambda_{ji}, \quad (2.20)$$

где  $\lambda_{ji}$  – интенсивность  $j$ -й фазы  $i$ -го потока, а

$$p(\lambda_{ji}) = \prod_{k=1, k \neq j}^{Q_i} (\lambda_{ki} - \lambda_{ji}).$$

2. Получение выражения для плотности распределения времени между заявками в суммарном потоке через параметры соответствующей аппроксимации.

В нашем случае, подставляя выражения (2.19) или (2.20) в формулу (2.18), после дифференцирования и преобразований получим

$$a(t) = \prod_{i=1}^p \frac{\lambda_i}{\lambda} \sum_{j_1=1}^{Q_1} \sum_{j_2=1}^{Q_2} \dots \sum_{j_p=1}^{Q_p} \left[ \left( \prod_{k=1}^p R_{j_k k} \cdot e^{-\lambda_{j_k k} t} \right) \cdot \left( \sum_{k=1}^p \lambda_{j_k k} \right)^2 \right], \quad (2.21)$$

где  $R_{j_k k} = \frac{C_{j_k k}}{\lambda_{j_k k}}$  при  $v \geq 1$ ;  $R_{j_k k} = \frac{\prod_{m_k=1}^{Q_k} \lambda_{m_k k}}{p(\lambda_{j_k k}) \cdot \lambda_{j_k k}^2}$  при  $v < 1$ .

3. Получение выражения для начальных моментов распределения  $a(t)$ . Для формулы (2.21) найдем следующее выражение:

$$v_n = \prod_{i=1}^p \frac{\lambda_i n!}{\lambda} \sum_{j_1=1}^{Q_1} \sum_{j_2=1}^{Q_2} \dots \sum_{j_p=1}^{Q_p} \left[ \prod_{k=1}^p R_{j_k k} / \left( \sum_{k=1}^p \lambda_{j_k k} \right)^{n-1} \right]. \quad (2.22)$$

В частности, при  $p=2$  и гиперэкспоненциальной аппроксимации распределений формулы (2.21) и (2.22) аналогичны формулам, полученным в [6]:

$$v_n = \frac{\lambda_1 \lambda_2 n!}{\lambda_1 + \lambda_2} \sum_{j_1=1}^{Q_1} \sum_{j_2=1}^{Q_2} \frac{\frac{C_{j_1 1}}{\lambda_{j_1 1}} \cdot \frac{C_{j_2 2}}{\lambda_{j_2 2}}}{\left( \lambda_{j_1 1} + \lambda_{j_2 2} \right)^{n-1}} = \frac{n!}{v_{11} + v_{21}} \sum_{j_1=1}^{Q_1} \sum_{j_2=1}^{Q_2} \left[ \frac{C_{j_1 1} C_{j_2 2}}{\lambda_{j_1 1} \lambda_{j_2 2} \left( \lambda_{j_1 1} + \lambda_{j_2 2} \right)^{n-1}} \right].$$

При обобщённой эрланговской аппроксимации распределений получим результаты, аналогичные результатам [20]:

$$v_n = \frac{n!}{v_{11} + v_{21}} \sum_{j_1=1}^{Q_1} \sum_{j_2=1}^{Q_2} \frac{\prod_{m_1=1}^{Q_1} \lambda_{m_1 1} \cdot \prod_{m_2=1}^{Q_2} \lambda_{m_2 2}}{\lambda_{j_1 1} \cdot \lambda_{j_2 2} \cdot p(\lambda_{j_1 1}) \cdot p(\lambda_{j_2 2}) \cdot \left( \lambda_{j_1 1} + \lambda_{j_2 2} \right)^{n-1}}.$$

4. Аппроксимация полученного суммарного потока на входе очередного узла фазовым распределением на основе метода моментов или метода производных [23].

#### 2.5.4. Приближённые методы расчёта сетей массового обслуживания

Условно все методы расчёта СеМО можно разбить на следующие:

1) *точные (аналитические)* позволяют получать решение в виде аналитического выражения, имеющего, как правило, мультипликативную форму;

2) *приближённые (аппроксимационные)* позволяют исследовать сети при наиболее общих предположениях о входящем потоке и распределении времени обслуживания.

Перечислим основные приближённые методы.

1. *Методы диффузной аппроксимации* основаны на аппроксимации дискретного случайного процесса, описывающего число заявок в узлах сети, непрерывным диффузным процессом с нормальным распределением в условиях большой нагрузки и установившегося режима работы сети. Погрешность метода значительно возрастает при небольшой загрузке и увеличении коэффициентов вариации времени обслуживания.

2. *Комбинаторные методы* основаны на алгоритмах типа свёртки Бузена для вычисления нормирующей константы, применимы только для замкнутых мультипликативных сетей разных видов.

3. *Метод средних значений* точен лишь для мультипликативных сетей.

4. Методы, основанные на *эвристических оценках* разомкнутых узлов.

5. *Итеративные методы* основаны на последовательном приближении решения с проверкой сходимости и коррекции.

6. *Методы декомпозиции* – используют концепцию декомпозиции и агрегирования. Сеть делится на подсети и решается отдельно.

Алгоритм расчёта узлов сетевой модели **декомпозиционным методом** предусматривает следующие шаги:

1) суммирование потоков на входе очередного узла аппроксимации плотности распределения входного потока;

2) расчёт начальных моментов времени ожидания в очередном узле согласно одному из методов (точных или приближённых);

3) нахождение начальных моментов времени пребывания в узле путём свёртки начальных моментов времени ожидания с начальными моментами времени обслуживания в узле;

4) расчёт начальных моментов выходного потока из узла и проведение декомпозиции выходного потока в соответствии с матрицей вероятностей передач;

5) переход к расчёту очередного узла сети;

6) нахождение характеристик времени пребывания в сети.

**Примечание.** Для расчёта циклических сетей используется один из приёмов:

– осуществляется эквивалентное преобразование сети с пересчётом параметров распределений времени обслуживания в узлах;

– выполняется итерационный расчёт входных и выходных потоков (рис. 2.21) до получения необходимой точности.

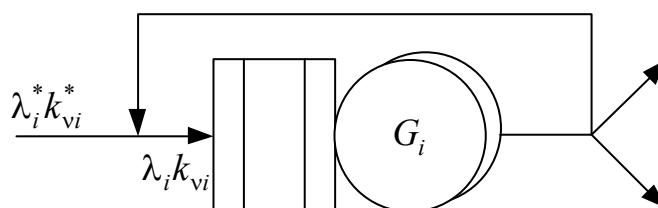


Рис. 2.21. Схема итерационного расчёта

Обозначим  $G_i^*(s)$  – ПЛС ПР времени пребывания задания в  $i$ -м узле сети. Введём в рассмотрение матрицу условных ПЛС ПР времени пребывания в  $i$ -м узле при условии перехода заявки после обслуживания в  $j$ -й узел:

$$Y(s) = \begin{vmatrix} p_{0,0} & p_{0,1} & \dots & p_{0,m} & p_{0,m+1} \\ p_{1,0}G_1(s) & p_{1,1}G_1(s) & \dots & p_{1,m}G_1(s) & p_{1,m+1}G_1(s) \\ \dots & \dots & \dots & \dots & \dots \\ p_{m,0}G_m(s) & p_{m,1}G_m(s) & \dots & p_{m,m}G_m(s) & p_{m,m+1}G_m(s) \\ p_{m+1,0} & p_{m+1,1} & \dots & p_{m+1,m} & p_{m+1,m+1} \end{vmatrix}$$

где  $p_{ij}$  – вероятность перехода заявки после обслуживания в  $i$ -м узле в  $j$ -й узел. При этом  $G_0^*(s) = G_{m+1}^*(s) = 1$ .

Исходя из этого получим матрицу ПЛС ПР пребывания заявки в сети между  $i$ -м и  $j$ -м узлом:

$$T(s) = I + Y(s) + Y^2(s) + \dots = I(I - Y(s))^{-1}.$$

ПЛС ПР времени пребывания заявки в сети – элемент матрицы  $T(s)$  номером  $(0, m+1)$ . В соответствии с правилами вычисления значений обратной матрицы, выражение для ПЛС ПР времени пребывания заявки в сети можно записать в виде:

$$G^*(s) = G_{0,m+1}^*(s) = A_{m+1,0}^*(s)/R(s); \quad (2.23)$$

где  $A_{m+1,0}^*(s)$  – алгебраическое дополнение элемента  $(0, m+1)$  матрицы  $T(s)$ ;

$R(s)$  – главный определитель матрицы  $T(s)$ .

Путём дифференцирования (2.23) получают начальные моменты времени пребывания заданий на обработке в сети, по которым можно построить ФР времени пребывания задания в сети [65, 66].

### **3. Моделирование случайных величин, процессов и полей**

#### **3.1. Моделирование случайных величин**

##### **3.1.1. Общая характеристика методов моделирования случайных величин**

Практически любая задача статистического моделирования содержит в качестве самостоятельного этапа получение реализаций случайной величины с заданным законом распределения.

Исходным материалом для формирования на ЭВМ случайных величин с различными законами распределения служат равномерно распределённые в интервале  $(0,1)$  случайные числа, которые вырабатываются на ЭВМ программным или же физическим датчиком случайных чисел.

Существуют различные приёмы преобразования случайных чисел с равномерным распределением в распределения с заданным законом [13]. Так, например, в качестве нормально распределённых случайных чисел можно использовать сумму нескольких независимых случайных чисел с равномерным распределением (приближение основано на центральной предельной теореме теории вероятностей, в силу которой сумма независимых случайных величин при весьма общих условиях имеет асимптотически нормальное распределение).

Рассмотрим сначала общие приёмы получения случайных чисел с заданным законом распределения из равномерно распределённых случайных чисел.

##### **3.1.2. Моделирование случайных величин с негауссовским распределением**

###### **Метод нелинейного преобразования, обратной функции распределения.**

Задачи моделирования случайных процессов, имеющих место в системах передачи и обработки сигналов, часто приводят к необходимости получения случайной величины с негауссовским законом распределения. Наиболее эффективным аналитическим методом получения негауссовских случайных величин является метод монотонного нелинейного преобразования (метод обратных функций).

Найдем закон распределения величины  $y$  полученной нелинейным преобразованием  $y = \psi(x)$  непрерывной случайной величины  $x$  (рис. 3.1). Будем считать, что существует взаимно однозначное преобразование  $y = \psi(x)$ . Обратное преобразование обозначим  $x = \psi^{-1}(y)$ .

Из рис. 3.1 видно, что всегда, когда случайная величина  $x$  попадает в интервал  $[x_0, x_0 + \Delta x]$ , случайная величина  $y$  попадает в интервал  $[y_0, y_0 + \Delta y]$ . Поэтому выполняется равенство

$$P\{x_0 < x < x_0 + \Delta x\} = P\{y_0 < y < y_0 + \Delta y\}, \quad (3.1)$$

откуда следует, что  $\omega(x)|\Delta x| = \omega(y)|\Delta y|$ . Получаем соотношение

$$\omega(y) = \frac{\omega(x)}{|dy/dx|}. \quad (3.2)$$

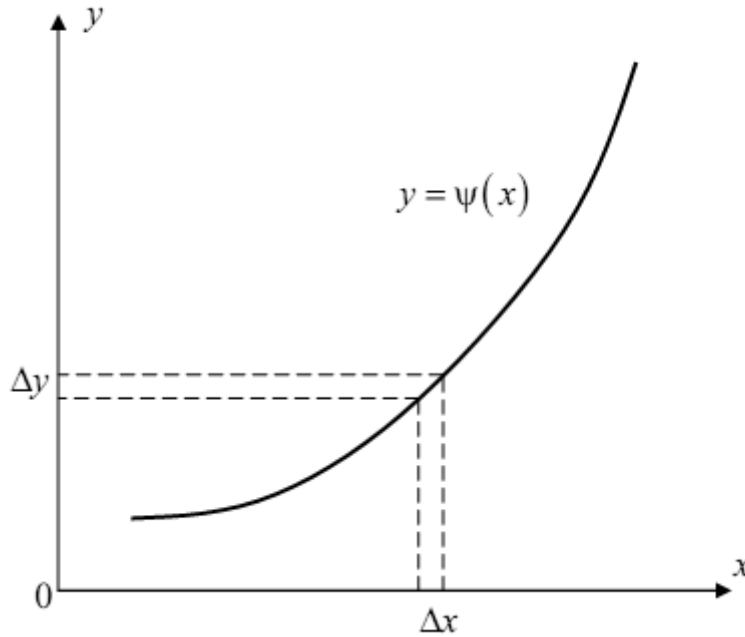


Рис. 3.1. Функциональное преобразование случайной величины

Рассмотрим типичный пример получения случайной величины с заданным законом распределения из случайной величины с равномерным распределением.

Пусть задана случайная величина  $x$  с равномерным законом распределения  $\omega(x) = 1$ ,  $x \in [0; 1]$ , необходимо получить случайную величину  $y$  с заданным законом распределения  $\omega(y)$ , которому соответствует некоторое нелинейное преобразование, например,  $y = x^3$  (рис. 3.2).

Далее по формуле (3.2) получаем плотность вероятности

$$\omega(y) = \frac{\omega(x)}{\left| \frac{dy}{dx} \right|} = 1 \cdot \frac{1}{3x^2} = \frac{1}{3 \cdot \sqrt[3]{y^2}}. \quad (3.3)$$

Теперь решим обратную задачу: найдем вид преобразования  $\psi(x)$  по заданной плотности распределения  $\omega(y) = 1 \cdot \left| \frac{dx}{dy} \right|$ ,  $y = \psi(x)$ . Для этого проинтегрируем левую и правую части (3.2)

$$x = \int_{-\infty}^y \omega(y) dy = F(y), \quad (3.4)$$

откуда находим функцию распределения  $F(y)$ , тогда случайную величину  $y$  можно найти с помощью преобразования  $y = \psi(x)$ .

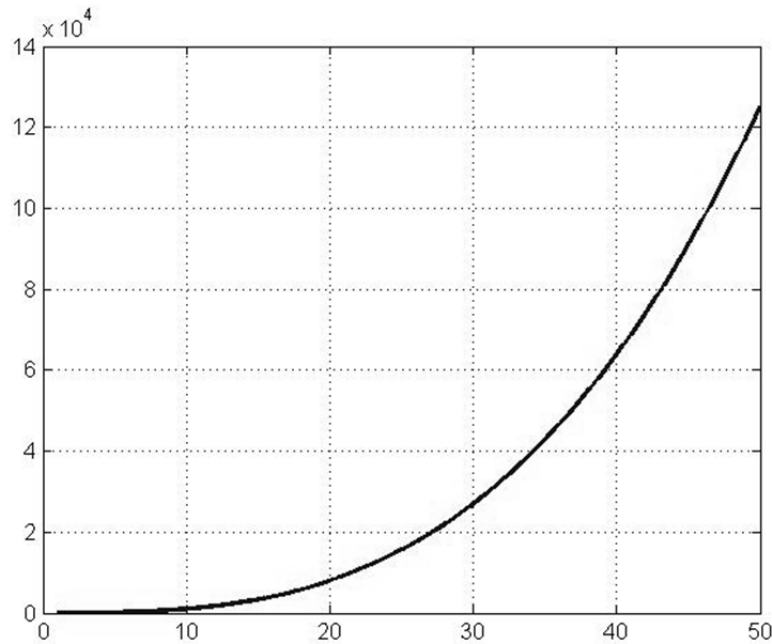


Рис. 3.2. График распределения случайной величины  $y = x^3$

Описанный выше метод моделирования называется **методом обратных функций**. Для моделирования случайной величины с заданной функцией распределения необходимо осуществить нелинейное преобразование вида

$$y = F^{-1}(x). \quad (3.5)$$

Формула (3.5) означает решение уравнения

$$F(y) = x; \quad x \sim U[0,1], \quad (3.6)$$

где  $x \sim U[0,1]$  означает, что случайная величина  $x$  имеет равномерное распределение на отрезке  $[0, 1]$ .

Комбинируя формулы (3.4) и (3.5), можно по реализации случайной величины  $x$  с произвольной функцией распределения моделировать величины с требуемой функцией распределения  $F(y)$ . Моделирующий алгоритм даёт суперпозиция нелинейных преобразований (3.4) и (3.5):

$$y = F^{-1}[F(x)].$$

Получим с помощью метода обратных функций моделирующие алгоритмы для ряда распределений, используемых при моделировании случайных процессов и полей.

Пусть  $\omega(y)$  – функция плотности,  $F(y) = \int_{-\infty}^y \omega(y)dy$  – функция распределения

вероятностей случайной величины  $y$ , а  $F^{-1}(x)$  – функция, обратная функции  $F(y)$ . Тогда случайная величина  $y = F^{-1}(x)$  имеет заданный закон распределения  $\omega(y)$ , если случайная величина  $x$  равномерно распределена в интервале  $(0,1)$  [13].



Например, случайную величину с релеевским законом распределения, у которой функция плотности, функция распределения, математическое ожидание и дисперсия имеют соответственно вид

$$\omega(y) = \frac{y}{\sigma^2} e^{-\frac{y^2}{2\sigma^2}}, y \geq 0; F(y) = 1 - e^{-\frac{y^2}{2\sigma^2}}, y \geq 0; m_y = \sqrt{\frac{\pi}{2}} \sigma; \sigma_y^2 = \left(2 - \frac{\pi}{2}\right) \sigma^2, \quad (3.7)$$

где  $\sigma$  – параметр распределения, можно получить путём преобразования равномерно распределённой в интервале  $(0,1)$  случайной величины  $x$  в соответствии с выражением:

$$y = \sigma \sqrt{-2 \ln(1-x)} = \sigma \sqrt{-2 \ln x}$$

(переход от  $\ln(1-x)$  к  $\ln x$  в последней формуле основан на том, что случайные величины  $1-x$  и  $x$  имеют здесь одинаковые законы распределения).

Аналогично случайную величину с показательным законом распределения, у которой

$$\omega(y) = \lambda e^{-\lambda y}, y \geq 0; F(y) = 1 - e^{-\lambda y}, y \geq 0; m_y = \sigma_y = \frac{1}{\lambda}, \quad (3.8)$$

можно сформировать путём преобразования  $y = -\frac{1}{\lambda} \ln x$ .

Путём преобразований

$$y = b \cdot \sin \pi \left( x - \frac{1}{2} \right) + a, \quad y = b \cdot \operatorname{tg} \pi \left( x - \frac{1}{2} \right) + a \quad (3.9)$$

можно сформировать случайные числа, распределённые по закону арксинуса и закону Коши соответственно:

$$\begin{aligned} \omega(y) &= \frac{1}{\pi b \sqrt{1 - (y-a)^2/b^2}}; F(y) = \frac{1}{\pi} \arcsin \frac{y-a}{b} + \frac{1}{2}; m_y = a; \sigma_y^2 = \frac{b^2}{2}; \\ \omega(y) &= \frac{1}{\pi b \left[ 1 + (y-a)^2/b^2 \right]}; F(y) = \frac{1}{\pi} \operatorname{arctg} \frac{y-a}{b} + \frac{1}{2}; m_y = a; \sigma_y^2 = \infty. \end{aligned} \quad (3.10)$$

Используя свойство симметрии тригонометрических функций, нетрудно убедиться, что закон распределения случайных величин  $y$ , формируемых согласно алгоритмам (3.9), не изменится, если аргумент  $\pi(x-1/2)$  у тригонометрических функций заменить аргументом  $2\pi x$ .

К сожалению, не всегда существуют элементарные преобразования для получения случайных величин с заданным законом распределения из равномерно распределённых случайных чисел. В частности, у случайных величин с нормальным распределением функция, обратная функции распределения, не выражается в замкнутом виде через элементарные функции. В этих случаях для формирования случайных величин с заданным распределением используются различные аппроксимации функции  $F^{-1}(y)$  [67].

## Метод Неймана

Для моделирования случайных величин, возможные значения которых не выходят за пределы некоторого ограниченного интервала  $(a, b)$  (случайные величины с усечёнными законами распределения), а также случайных величин, законы распределения которых можно аппроксимировать усечёнными, достаточно универсальным является метод Неймана [77, 11], состоящий в следующем.

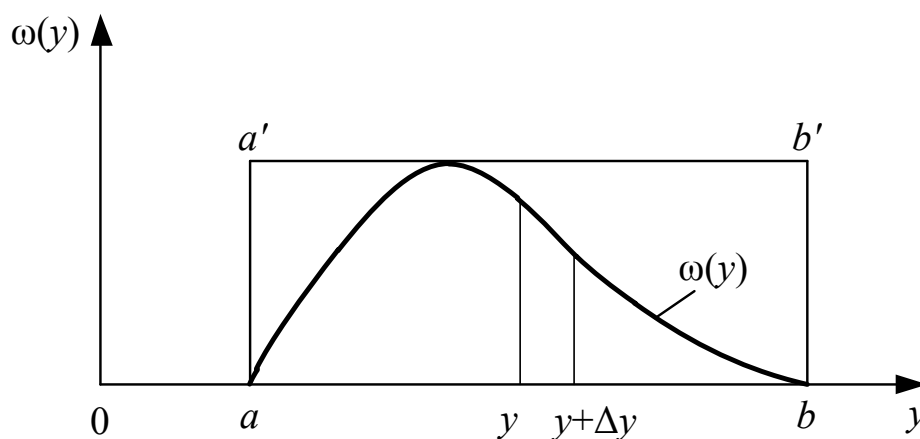


Рис. 3.3. Кривая плотности вероятности

Из датчика равномерно распределённых в интервале  $(0,1)$  случайных чисел независимо выбираются пары чисел  $x_{1k}, x_{2k}$ , из которых формируются преобразованные пары  $x_{1k}^* = a + (b - a) \cdot x_{1k}$ ,  $x_{2k}^* = \omega_M \cdot x_{2k}$ , где  $(a, b)$  — интервал возможных значений случайной величины  $y$  с заданной функцией плотности  $\omega(y)$ ;  $\omega_M$  — максимальное значение функции  $\omega(y)$ . В качестве реализации случайной величины берется число  $x_{2k}^*$  из тех пар  $x_{1k}^*, x_{2k}^*$ , в которых выполняется неравенство

$$x_{2k}^* \leq \omega(x_{1k}^*). \quad (3.11)$$

Пары, не удовлетворяющие неравенству (3.11), выбрасываются.

Нетрудно убедиться в справедливости такого метода моделирования случайных величин. Действительно, пары случайных чисел  $x_1^*, x_2^*$  можно рассматривать как координаты случайных точек плоскости, равномерно распределённых вдоль осей  $y$  и  $\omega(y)$  внутри прямоугольника  $aa'bb'$  (рис. 3.3). Пары  $x_1^*, x_2^*$ , удовлетворяющие условию (3.11), — это координаты случайных точек плоскости, равномерно распределённых вдоль осей  $y$  и  $\omega(y)$  внутри той части прямоугольника  $aa'bb'$ , которая расположена под кривой  $\omega(y)$ . Вероятность того, что случайная точка плоскости, находящаяся под кривой  $\omega(y)$ , окажется в элементарной полосе с основанием  $(y, y + \Delta y)$ , очевидно, пропорциональна  $\omega(y)$ , а вероятность попадания точки под кривую  $\omega(y)$  по условию равна единице, что и требуется.

## Метод кусочной аппроксимации

Существуют различные приближённые приёмы моделирования случайных величин: численное решение уравнения  $x = F(y)$  относительно  $y$  при использовании метода нелинейного преобразования, обратной функции распределения; замена непрерывных распределений соответствующими дискретными распределениями, для которых можно указать достаточно простые моделирующие алгоритмы, и другие приёмы [67].

Среди них универсальным и наиболее простым является метод кусочной аппроксимации, предложенный Н.П. Бусленко [9].

Сущность этого метода состоит в следующем. Пусть требуется получить случайную величину  $y$  с функцией плотности  $\omega(y)$ . Предположим, что область возможных значений величины  $y$  ограничена интервалом  $(a, b)$  (неограниченное распределение можно приближённо заменить ограниченным). Разобьём интервал  $(a, b)$  на  $n$  достаточно малых интервалов  $(a_m, a_{m+1})$ ,  $m = \overline{0, n-1}$ ,  $a_0 = a$ ,  $a_n = b$ , так, чтобы распределение заданной случайной величины в пределах этих интервалов можно было довольно точно аппроксимировать каким-нибудь простым распределением, например равномерным, трапецеидальным и т. д. В дальнейшем рассмотрим кусочную аппроксимацию равномерным распределением (рис. 3.4).

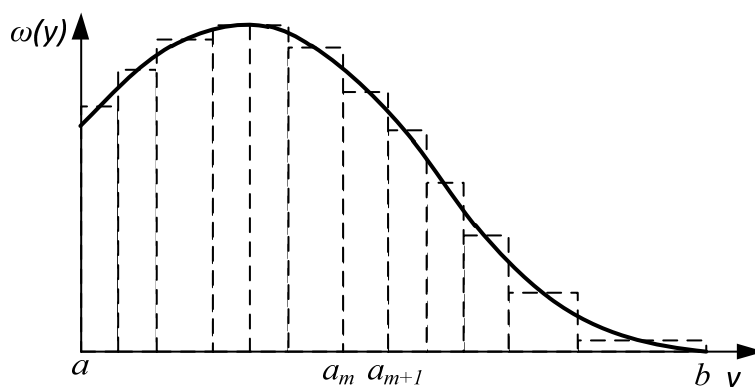


Рис. 3.4. Кусочная аппроксимация кривой плотности вероятности

Пусть  $P_m$  – вероятность попадания случайной величины  $y$  в каждый из интервалов  $(a_m, a_{m+1})$ . Получать реализации величины  $y$  с кусочно-равномерным распределением можно, очевидно, в соответствии со следующей схемой преобразования случайных чисел:

- 1) случайным образом с вероятностью  $P_m$  выбирается интервал  $(a_m, a_{m+1})$ ;
- 2) формируется реализация  $\Delta y_{m_k}$  случайной величины, равномерно распределённой в интервале  $(0, a_{m+1} - a_m)$ ;

3) искомая реализация  $y_k$  получается по формуле

$$y_k = a_m + \Delta y_{m_k}. \quad (3.12)$$

Случайный выбор интервала  $(a_m, a_{m+1})$  с вероятностью  $P_m$  означает, по существу, моделирование дискретной случайной величины, принимающей  $n$  значений  $a_m$ ,  $m = \overline{0, n-1}$ , с вероятностью  $P_m$  каждое, что можно сделать достаточно просто [9]. Интервал  $(0,1)$  разбивается на  $n$  интервалов  $(x_m, x_{m+1})$ ,  $m = \overline{0, n-1}$ ,  $x_0 = 0$ ,  $x_n = 1$  длиной  $x_{m+1} - x_m = P_m$  каждый. Из датчика случайных равномерно распределённых в интервале  $(0,1)$  чисел выбирается некоторая реализация  $x_k$ . Путём последовательного сравнения  $x_k$  с  $x_m$  определяется тот интервал  $(x_m, x_{m+1})$ , в котором оказывается  $x_k$ .

В основу этого процесса положен очевидный факт: вероятность попадания равномерно распределённой в интервале  $(0,1)$  случайной величины в некоторый подинтервал  $(x_m, x_{m+1})$  равна длине этого подинтервала.

Рассмотренный выше процесс представляет интерес не только как составной элемент метода кусочной аппроксимации, он широко используется в качестве алгоритма для моделирования дискретных случайных величин и случайных событий [9, 62].

Для моделирования случайных величин методом кусочной аппроксимации наиболее удобно при машинной реализации выбирать вероятности попадания во все интервалы  $(a_m, a_{m+1})$  одинаковыми  $P_m = \frac{1}{n}$ , а число  $n$  таким, что  $n = 2^N$ , где  $N$  – целое число, меньше или равное количеству двоичных разрядов чисел, вырабатываемых датчиком случайных чисел [9, 62]. В этом случае величины  $a_m$  должны быть выбраны такими, чтобы

$$\int_{a_m}^{a_{m+1}} \omega(y) dy = \frac{1}{n} = 2^{-N}. \quad (3.13)$$

При равенстве вероятностей  $P_m$  для случайного выбора индекса  $m$  можно использовать первые  $N$  разрядов числа, извлекаемого из датчика равномерно распределённых случайных чисел.

Используя рассмотренный приём, приходим к следующему способу преобразования равномерно распределённых случайных чисел в случайные числа с заданным законом распределения.

Из датчика равномерно распределённых в интервале  $(0,1)$  случайных чисел извлекается пара реализаций  $x_{1_k}, x_{2_k}$ . Первые  $N = \log_2 n$  разрядов числа  $x_{1_k}$  используются для нахождения адресов ячеек, в которых хранятся величины  $a_m$  и  $a_{m+1}$ ,

а затем по формуле

$$y_k = a_m + x_{2k} (a_{m+1} - a_m) \quad (3.14)$$

получается реализация  $y_k$  случайной величины  $y$  с заданным законом распределения. Такой алгоритм является довольно экономичным по количеству требуемых операций, которое не зависит от числа  $n$ , т. е. не зависит от точности кусочной аппроксимации. Однако с увеличением точности аппроксимации возрастает количество ячеек памяти, требуемое для хранения величин  $a_m$ ,  $m = \overline{0, n}$ , что является недостатком рассмотренного метода, в особенности при больших  $n$ .

### Некоторые специальные методы моделирования случайных величин

Для моделирования случайных величин с заданным законом распределения можно использовать и другие свойства преобразований случайных чисел. Известно, например, что распределение произведения двух независимых случайных величин, одна из которых имеет релеевское распределение (3.7), а другая распределена по закону арксинуса (3.10) с параметрами  $(0, 1/2)$ , т. е. с нулевым средним значением и дисперсией, равной  $1/2$ , является нормальным [4, 61]. Это позволяет формировать нормальную случайную величину путём следующего преобразования системы двух независимых равномерно распределённых в интервале  $(0, 1)$  случайных чисел  $x_1$  и  $x_2$ :

$$y = \sigma \sqrt{-2 \ln x_1} \cdot \sin 2\pi x_2. \quad (3.15)$$

Параметры получаемой этим способом нормальной случайной величины будут  $(0, \sigma^2)$ .

Для моделирования случайных величин с некоторыми законами распределения иногда удобно использовать преобразования нормально распределённых случайных чисел. Так, например, случайные величины с релеевским и показательным законами распределения (3.7) и (3.8) можно получить путём преобразования системы двух независимых нормальных случайных чисел  $x_1$  и  $x_2$  с параметрами  $(0, \sigma^2)$  в виде

$$y = \sqrt{x_1^2 + x_2^2}, \quad (3.16)$$

$$y = x_1^2 + x_2^2 \quad (3.17)$$

соответственно. При этом для релеевского распределения (3.7) параметр  $\sigma$  будет совпадать с параметром  $\sigma$  исходного нормального распределения, а для показательного распределения (3.8) параметр  $\lambda$  связан с параметром  $\sigma$  исходного нормального распределения соотношением  $\lambda = 1/2\sigma^2$ .

Алгоритмы (3.16) и (3.17) основаны на известных свойствах преобразований нормальных случайных величин [4]. Немного изменив эти алгоритмы, можно моделировать случайные величины с другими распространёнными законами распределения, а именно, обобщая формулы (3.16) и (3.17) в виде

$$y = \sqrt{(x_1 + a)^2 + x_2^2}, \quad y = \sum_{k=1}^m x_k^2, \quad (3.18)$$

где  $x_k$  – случайные числа, распределённые по нормальному закону с параметрами  $(0, \sigma^2)$ , получим алгоритмы для моделирования случайных величин с законом распределения Райса и законом распределения  $\chi^2$  с  $m$  степенями свободы соответственно:

$$\omega(y) = \frac{y}{\sigma^2} e^{-\frac{y^2 + a^2}{2\sigma^2}} I_0\left(\frac{ay}{\sigma^2}\right), \quad y \geq 0; \quad \omega(y) = \frac{1}{2^{m/2} \Gamma(m/2) \sigma^2} \left(\frac{y}{\sigma^2}\right)^{\frac{m}{2}-1} e^{-\frac{y}{2\sigma^2}}, \quad y \geq 0,$$

где  $I_0(x)$  – модифицированная функция Бесселя нулевого порядка;

$\Gamma(x)$  – гамма-функция.

### 3.1.3. Моделирование случайных величин с гауссовским распределением

Простейшие, но неточные методы моделирования основываются на центральной предельной теореме.

Существует много вариантов и соответственно формулировок центральной предельной теоремы, но во всех этих вариантах суть условий одна: если случайная величина может быть представлена в виде суммы большого числа независимых (или слабо зависимых) случайных величин, каждая из которых мала по сравнению с суммой, то эта сумма имеет закон распределения близкий к нормальному.

Например, если сложить 12 независимых базовых случайных величин, получится грубое приближение стандартного нормального распределения. С увеличением слагаемых распределение суммы стремится к нормальному.

Использование точных методов предпочтительно, поскольку у них практически нет недостатков. В частности, преобразование Бокса–Мюллера является точным, быстрым и простым для реализации методом генерации.

Преобразование Бокса–Мюллера – метод моделирования стандартных нормально распределённых случайных величин – имеет два варианта. Метод является точным, в отличие, например, от методов, основывающихся на центральной предельной теореме. Метод был опубликован в 1958 году Джорджем Боксом и Мервином Мюллером.

### *Первый вариант*

Пусть  $r$  и  $\varphi$  – независимые случайные величины, равномерно распределённые на интервале  $(0,1]$ . В этом случае, величины  $z_0$  и  $z_1$ , полученные по формулам

$$\begin{aligned} z_0 &= \cos(2\pi\varphi)\sqrt{-2\ln r}; \\ z_1 &= \sin(2\pi\varphi)\sqrt{-2\ln r}, \end{aligned} \quad (3.19)$$

будут независимы и распределены нормально с математическим ожиданием равным 0 и дисперсией 1.

При реализации на компьютере обычно быстрее не вычислять обе тригонометрические функции, а рассчитать одну из них через другую. Ещё лучше воспользоваться вместо этого вторым вариантом преобразования Бокса – Мюллера.

### *Второй вариант*

Пусть  $x$  и  $y$  – независимые случайные величины, равномерно распределённые на отрезке  $[-1;1]$ . Необходимо вычислить  $s = x^2 + y^2$ . Если окажется, что  $s > 1$  или  $s = 0$ , то значения  $x$  и  $y$  не следует использовать. Как только выполнится условие  $0 < s \leq 1$ , по формулам

$$\begin{aligned} z_0 &= x \cdot \sqrt{\frac{-2\ln s}{s}}; \\ z_1 &= y \cdot \sqrt{\frac{-2\ln s}{s}}, \end{aligned} \quad (3.20)$$

следует рассчитать  $z_0$  и  $z_1$ , которые, как и в первом случае, будут независимыми величинами, удовлетворяющими стандартному нормальному распределению.

Коэффициент использования базовых случайных величин для первого варианта равен единице. Для второго варианта это отношение площади окружности единичного радиуса к площади квадрата со стороной два, то есть  $\pi/4 \approx 0,785$ . Тем не менее, на практике второй вариант обычно оказывается быстрее за счёт того, что в нём используется только одна трансцендентная функция,  $\ln(\cdot)$ . Это преимущество для большинства реализаций перевешивает необходимость генерации большего числа равномерно распределённых случайных величин.

После получения стандартной нормальной случайной величины  $z$  можно легко перейти к величине  $\xi \sim N(m, \sigma^2)$ , распределённой нормально с математическим ожиданием  $m$  и среднеквадратическим отклонением  $\sigma$  по формуле

$$\xi = m + \sigma z. \quad (3.21)$$

Это уже не является частью преобразования Бокса – Мюллера, но позволяет завершить генерацию нормальной случайной величины.

### 3.1.4. Метод суперпозиции

Рассмотрим дискретную случайную величину  $y$ , принимающую  $n$  значений  $a_k$  с вероятностями  $P_1, \dots, P_n$ . Эта величина задаётся рядом распределения

$$\begin{pmatrix} a_1, & a_2, & \dots, & a_n \\ P_1, & P_2, & \dots, & P_n \end{pmatrix}, \sum_{k=1}^n P_k = 1.$$

Обычно используют следующий алгоритм моделирования. Отрезок  $[0,1]$  разбивают на  $n$  последовательных отрезков  $\Delta_1, \dots, \Delta_n$ , длины которых равны соответственно вероятностям  $P_1, \dots, P_n$ . Разыгрывается значение величины  $x \in [0,1]$  с равномерным распределением и далее принимается  $y = a_k$  если  $x \in \Delta_k$ .

Этот алгоритм применим и для дискретных случайных величин, принимающих бесконечное множество значений. Для моделирования случайных величин с плотностью распределения вида

$$\omega(y) = \sum_{i=1}^n p_i \omega_i(y), \text{ где } p_i > 0, \sum_{i=1}^n p_i = 1 \quad (3.22)$$

удобен метод суперпозиции. Моделирование осуществляется в два этапа. Сначала разыгрывается реализация дискретной случайной величины, принимающей значения  $1, 2, \dots, n$  с вероятностями  $p_k$ . После получения значения  $k$ , моделируется случайная величина с плотностью распределения вероятности  $\omega_k(y)$ . Её значение и принимается в качестве  $y$ .

Модели вида (3.22) называются смесями распределений  $\omega_1, \dots, \omega_n$ .

Описанный алгоритм по существу воспроизводит реальный физический механизм появления смесей распределений. Сумма в формуле (3.22) может содержать большое число слагаемых.

Рассмотрим пример применения метода суперпозиции. Пусть требуется промоделировать случайную величину с плотностью распределения вероятности вида

$$\omega(y) = \frac{\alpha}{2\pi} \left[ \frac{1}{\alpha^2 + (y + \beta)^2} + \frac{1}{\alpha^2 + (y - \beta)^2} \right]. \quad (3.23)$$

Функция (3.23) может рассматриваться как смесь двух распределений Коши, отличающихся параметрами сдвига  $b = \pm\beta$ . Вероятности  $p_i$  равны  $p_1 = p_2 = 0,5$ .

Моделирующий алгоритм для распределения Коши имеет вид

$$y = \alpha \cdot \operatorname{tg} \left[ \pi \left( x - \frac{1}{2} \right) + x_\beta \right], \quad (3.24)$$

где  $x, x_\beta$  - независимы,  $x \sim U[0;1]$ ,  $x_\beta$  принимает с равными вероятностями два значения  $\pm\beta$ .



Рассмотрим случайную величину, имеющую плотность распределения вероятности

$$\omega(y) = \frac{1}{2\alpha\sqrt{2\pi}} \left\{ \exp \left[ -\frac{(y+\beta)^2}{2\alpha^2} \right] + \exp \left[ -\frac{(y-\beta)^2}{2\alpha^2} \right] \right\}. \quad (3.25)$$

Распределение (3.25) есть смесь двух нормальных распределений с равными дисперсиями  $\alpha^2$  и математическими ожиданиями  $\pm\beta$ ,  $p_1 = p_2 = 0,5$ . Метод суперпозиции даёт следующий моделирующий алгоритм:

$$y = \alpha\xi + x_\beta.$$

Здесь  $\xi \sim N(0, 1)$ ,  $x_\beta$  определена в формуле (3.24) и обе величины независимы между собой.

## 3.2. Моделирование случайных процессов

### 3.2.1. Модели случайных процессов

Случайная величина позволяет представить поведение изменяющегося случайным образом сигнала в определённый момент времени. Однако при проектировании целого ряда систем связи важно учитывать изменение случайных сигналов не только по уровню, но и во времени. В качестве моделей случайных сигналов и помех, позволяющих отразить их динамические характеристики, используются случайные процессы, представляющие собой случайные функции времени. При этом конкретный вид, который принимает случайный процесс в отдельном эксперименте, называется реализацией случайного процесса.

Во многих радиотехнических приложениях случайные величины  $(x_1, x_2, \dots, x_n)$  связаны со значениями непрерывного процесса  $x(t)$  в моменты времени  $t_1, t_2, \dots, t_n$ , т. е.  $x_1 = x(t_1)$ ,  $x_2 = x(t_2)$ , ...,  $x_n = x(t_n)$ . В этом случае упорядоченная система непрерывных случайных величин  $x_1, x_2, \dots, x_n$  (рис. 3.5) называется случайной последовательностью, которую можно также интерпретировать как реализацию случайного процесса в данном опыте [14]

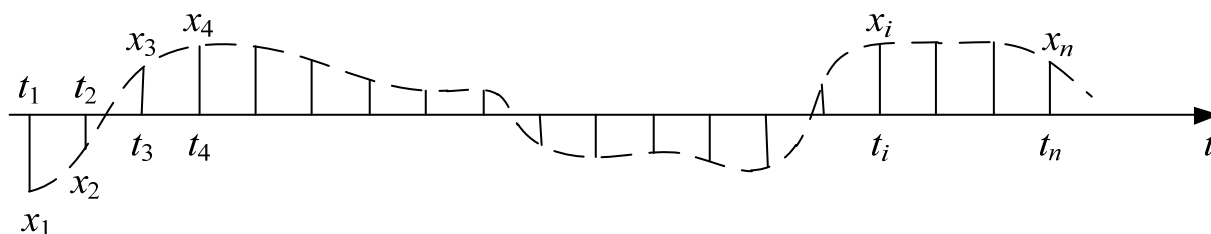


Рис. 3.5. Случайная последовательность

Простейшее вероятностное описание случайной последовательности соответствует независимым случайным величинам  $x_1, x_2, \dots, x_n$ , тогда совместная

плотность распределения вероятности  $f(x_1, x_2, \dots, x_n) = \prod_{i=1}^n f(x_i)$ . Однако последовательность независимых случайных величин представляет собой математическую модель довольно узкого класса реальных процессов.

Действительно, с помощью случайных процессов с независимыми значениями невозможно дать описание «гладких», коррелированных помех или медленно изменяющихся параметров полезных сигналов, например, координат радиолокационных целей. Поэтому во многих задачах необходимо использовать модели случайного процесса с зависимыми значениями. В общем случае совместная плотность распределения вероятности таких случайных процессов определяется по формуле

$$f(x_1, x_2, \dots, x_n) = f(x_1) \cdot f(x_2/x_1) \cdot f(x_3/x_1, x_2) \cdot \dots \cdot f(x_n/x_1, x_2, \dots, x_{n-1}).$$

Математические трудности применения этой формулы для вероятностных расчётов быстро нарастают с увеличением  $n$ . В связи с этим необходимо из всех возможных случайных процессов с зависимыми значениями выделить класс случайных процессов, имеющих относительно простое математическое описание. Очевидно, наиболее простые соотношения для плотности распределения вероятности получатся, если положить

$$f(x_i/x_1, x_2, \dots, x_{i-1}) = f(x_i/x_{i-1}). \quad (3.26)$$

Это равенство означает, что условная плотность распределения вероятности и, следовательно, любые другие вероятностные характеристики случайного процесса для момента времени  $t_i$  являются функциями только значения  $x_{i-1}$ , принятого случайного процесса в предшествующий момент времени.

Случайные последовательности, удовлетворяющие (3.26), называются *марковскими случайными последовательностями* по имени русского математика А.А. Маркова, разработавшего основы теории таких случайных процессов.

Марковская последовательность называется *однородной*, если условные плотности распределения вероятностей  $f(x_i/x_{i-1})$ , называемые плотностями распределения вероятностей перехода, не зависят от  $i$ .

Марковская последовательность называется *стационарной*, если она однородна и все состояния  $x_i$  имеют одну и ту же безусловную плотность распределения вероятности  $f(x)$ .

При моделировании марковских случайных процессов для формирования на ЭВМ случайных чисел с заданным законом распределения могут быть использованы методы, рассмотренные выше.

В более общем случае рассматриваются  $N$ -связные марковские процессы, т. е.  $N$  взаимосвязанных между собой процессов  $x_1(t), \dots, x_n(t)$ , в совокупности обла-

дающих марковскими свойствами [4, 11]. Эти процессы характеризуются условной плотностью распределения вероятностей перехода, которая имеет вид:

$$f(x_{1,n}, \dots, x_{N,n}, t_n | x_{1,n-1}, \dots, x_{N,n-1}, t_{n-1}) = f_0(x_{1,n}, \dots, x_{N,n}, t_n | x_{1,n-1}, \dots, x_{N,n-1}, t_n, t_{n-1}).$$

Моделирование  $N$ -связных марковских процессов по заданной условной плотности распределения вероятности перехода в принципе не отличается от моделирования рассмотренных выше одномерных (простейших) марковских процессов, однако получение  $N$ -связных дискретных реализаций с ростом  $N$  усложняется.

Другим обобщением одномерных марковских процессов являются одномерные марковские процессы  $N$ -го порядка, отличающиеся от простейших марковских процессов тем, что плотность распределения вероятностей перехода в очередное состояние зависит не от одного, а от  $N$  предшествующих состояний.

Стохастическое уравнение  $x_i = \varphi(x_{i-N}, x_{i-N+1}, \dots, x_{i-2}, x_{i-1}, \xi_i)$  при соответствующих начальных условиях порождает марковский процесс  $N$ -го порядка, который можно рассматривать как компоненту  $N$ -связного марковского процесса, поэтому моделирование марковских процессов  $N$ -го порядка может быть сведено к моделированию  $N$ -связных марковских процессов.

Выше шла речь о моделировании марковских процессов общего вида: на характеристики процессов не накладывалось других ограничений, кроме указанных выше. Распространёнными являются марковские процессы, которые удовлетворяют дополнительным условиям, чаще всего, условию нормальности распределения, стационарности (однородности), а также условию нормальности и стационарности одновременно. В этих случаях моделирование марковских процессов упрощается.

Как уже отмечалось, наиболее полное описание стационарных случайных процессов даёт многомерная плотность распределения вероятности. Однако этот подход требует большого количества информации. Для описания негауссовских случайных процессов используются различные преобразования гауссовских процессов и марковские процессы. Реальные случайные процессы можно с требуемой точностью аппроксимировать многомерными марковскими процессами.

Действительно, любой случайный процесс, спектральная плотность которого является дробно-рациональной функцией частоты, является компонентой многомерного марковского процесса.

### **Разностные и дифференциальные стохастические уравнения**

Одним из достоинств марковских процессов является возможность их описания с помощью *стохастических дифференциальных уравнений*. Для этого вводится понятие формирующей динамической системы. При этом случайный

процесс характеризуется параметрами линейных или нелинейных фильтров, на вход которых подаётся белый гауссовский шум или другое известное возмущение [57]. Так как любая динамическая система может быть описана дифференциальными уравнениями, то и случайный процесс на её выходе также может быть описан стохастическими дифференциальными уравнениями соответствующего порядка.

Модели случайных процессов в виде стохастических дифференциальных уравнений обладают рядом положительных свойств. Прежде всего, в отличие от других методов описания случайных процессов (например, с помощью плотности распределения вероятностей или моментных функций), они непосредственно определяют способ генерации его реализаций, для осуществления которого можно вполне успешно использовать аналоговые вычислительные устройства или соответствующие программы для ЭВМ. Использование стохастических дифференциальных уравнений в форме уравнений состояния позволяют синтезировать алгоритмы оптимальной обработки и генерации для широкого класса сигналов и помех [33, 57, 72].

Для большинства реальных случайных процессов используется модель в виде дифференциальных уравнений, линейных относительно входных воздействий типа белого шума

$$\frac{d\bar{x}(t)}{dt} = \psi(\bar{x}, t) + v(\bar{x}, t)\bar{\xi}(t), \quad (3.27)$$

где  $\psi(\bar{x}, t)$ ,  $v(\bar{x}, t)$  – векторная и матричная функции векторного аргумента соответственно [16, 44];

$\bar{\xi}(t)$  – белый гауссовский шум с нулевым математическим ожиданием и заданной корреляционной функцией.

Задавая различные функции  $\psi(\bar{x}, t)$  и  $v(\bar{x}, t)$  в уравнении (3.27), можно получить случайный процесс с различными статистическими характеристиками. Так, если  $\psi(\bar{x}, t)$  – линейная функция, а  $v(\bar{x}, t)$  не зависит от  $\bar{x}(t)$ , то уравнение (3.27) будет определять гауссовский случайный процесс.

Следует отметить, что задача синтеза марковской модели случайного процесса, т. е. определение функций  $\psi(\bar{x}, t)$  и  $v(\bar{x}, t)$  по заданным статистическим характеристикам, не всегда имеет одно решение. Здесь возникает вопрос о единственности решения. Если заданы статистические характеристики моделируемого процесса в форме функций распределения, то можно указать несколько уравнений различных типов, порождающих процессы с такими характеристиками. В случае нелинейных моделей это является неизбежным следствием

ограниченности исходных сведений о процессе. Если единственное решение задачи синтеза отсутствует, то при выборе уравнения, которое будет заложено в основу функционирования какого-либо устройства, можно учитывать соображения сложности, экономичности и так далее. На практике число эквивалентных, с точки зрения статистических характеристик, уравнений обычно не велико (не более трёх, четырёх), и из них всегда можно выбрать единственное [57].

В последнее время, в связи с широким использованием цифровых устройств в радиотехнических системах для статистического описания сигналов и помех используют случайные последовательности, которые могут быть заданы *разностными стохастическими уравнениями*

$$\bar{x}_i = \varphi(\bar{x}_{i-1}, i) + v(\bar{x}_{i-1}, i-1) \bar{\xi}_i, \quad i = 2, 3, \dots, k, \quad (3.28)$$

где  $v(\bar{x}_{i-1}, i-1)$  – матричная функция;

$\varphi(\bar{x}_{i-1}, i)$  – векторная функция;

$\bar{x}_i$  – состояние системы в момент времени  $t_i$ ;

$\bar{\xi}_i$  – последовательность случайных величин.

Если в качестве возбуждающей последовательности  $\bar{\xi}_i$  использовать дискретный белый шум, то уравнение (3.28) будет определять марковскую последовательность.

При использовании стохастических уравнений (3.26), (3.28) в качестве моделей случайных сигналов и помех возникают *задачи анализа и синтеза*. Первая из них заключается в определении статистических характеристик (в первую очередь плотность распределения вероятности и корреляционная функция) случайного процесса или случайной последовательности по заданным функциям  $\psi$ ,  $v$  и  $\varphi$ . Более сложной задачей синтеза является определение неизвестных функций  $\psi$ ,  $v$  и  $\varphi$  по заданным статистическим характеристикам.

Для непрерывных СП обе эти задачи решены для достаточно большого числа частных случаев. При этом анализ выполняется на основе решения уравнения Фоккера – Планка – Колмогорова [72].

Для случайных последовательностей эти задачи до конца решены лишь для случая линейной функции  $\varphi(\bar{x}_{i-1}, i)$  и независимой от  $\bar{x}_{i-1}$  функции  $v$  [13].

Кроме того, в работах [57, 73, 72] рассмотрено решение задачи синтеза и для нелинейной функции. Однако при этом накладываются ограничения на вид плотности распределения вероятности. Отсутствие конструктивных решений задачи синтеза, применительно к разностному уравнению (3.28), не позволяет получить математические модели случайных последовательностей с заданными статистическими свойствами.

## Нелинейные модели марковских случайных процессов

Рассмотрим задачу построения стохастических моделей негауссовских случайных процессов с заданной плотностью распределения вероятности. Возможность достаточно простого определения нелинейных функций  $\psi(\bar{x}, t)$  и  $v(\bar{x}, t)$  для стохастических уравнений (3.27) обусловлена существованием обыкновенного дифференциального уравнения, связывающего эти функции с плотностью распределения вероятности  $f(x)$  при  $t \rightarrow 0$ . При этом задача нахождения нелинейных функций  $\psi(\bar{x}, t)$  и  $v(\bar{x}, t)$  наиболее проста если  $f(x)$  входит в класс распределения Пирсона [44]. Вместе с тем, для стохастических разностных уравнений (3.28) лишь для линейных систем может быть точно определено безусловное распределение.

В работах [33, 73] подробно исследованы стохастические дифференциальные уравнения, определяющие различные негауссовские процессы. Векторное стохастическое дифференциальное уравнение, описывающее непрерывную динамическую систему можно представить:

$$d\bar{x}(t) = \psi(\bar{x}, t)dt + v(\bar{x}, t)\sqrt{N_\xi(t)}d\chi(t) \quad (3.29)$$

где  $N_\xi(t)$  – диагональная матрица с неотрицательными элементами;

$\chi(t)$  – стандартный винеровский процесс с  $M\{d\chi(t)d\chi^T(t)\} = E dt$ ;

$E$  – единичная диагональная матрица.

При этом стационарная плотность распределения вероятности будет:

$$1) \text{ рэлеевской } f(x) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right), \quad x > 0, \quad \sigma^2 = \frac{N_\xi}{2a},$$

$$\text{если } \frac{dx}{dt} = ax + \frac{N_\xi}{2x} + \xi(t); \quad (3.30)$$

$$2) \text{ Накагами } f(x) = \frac{2}{\Gamma(m)} \left(\frac{m}{a}\right)^m x^{2m-1} \exp\left(-\frac{mx^2}{a}\right), \quad x > 0, \quad a > 0, \quad m \geq 0,5;$$

$$\text{когда } \frac{dx}{dt} = -N_\xi \left(2m \left(\frac{x^2}{a} + 1\right) - 1\right) x^{1-4m} + \sqrt{2} x^{1-2m} \xi(t). \quad (3.31)$$

$$3) \text{ логарифмически-нормальной } f(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-\frac{\ln^2(x/m)}{2\sigma^2}\right), \quad x > 0, \quad \sigma^2 = \frac{aN_\xi}{2},$$

$$\frac{dx}{dt} = -ax(\ln(x/m) - \sigma^2) + ax\xi(t). \quad (3.32)$$

Стохастические дифференциальные уравнения (3.29–3.32) исследованы наиболее полно, дают адекватное описание случайных процессов во многих радиотехнических системах и позволяют получить замкнутые выражения для алгоритмов оценивания сигналов. Однако системы, базирующиеся непосред-

ственно на таких уравнениях, как правило, не могут быть реализованы и исследованы из-за отсутствия необходимых стабильных нелинейных элементов аналоговой техники. В связи с этим на практике осуществляется переход от непрерывных к дискретным во времени системам на основе различных разностных схем [57, 63]. Для обеспечения соответствия разностной схемы и уравнений вида (3.28), (3.29) от ЭВМ требуется высокое быстродействие и большая точность представления чисел.

Стохастические разностные уравнения (3.28) свободны от этого недостатка, так как представляют последовательность чисел, для формирования которой можно воспользоваться методами цифровой техники. Поэтому в задачах с нелинейными моделями наблюдений, но при гауссовских компонентах, системы с дискретным временем имеют очевидные преимущества при технической реализации. Вместе с тем, при отличии плотности распределения вероятности случайных последовательностей (3.28) от гауссовских возникает задача определения нелинейных функций  $\varphi(\bar{x}_{i-1}, i)$  и  $v(\bar{x}_{i-1}, i-1)$  по заданным распределениям. В ряде задач желательно построить (синтезировать) нелинейное уравнение (3.28) на основе априорных данных о виде безусловного одномерного распределения. К сожалению, такого же простого пути, какой существует для построения моделей (3.30–3.32), здесь нет. В работе [30] синтез марковской модели приводится не на основе априорных статистических характеристик случайных процессов, а на экспериментальных данных. При этом решение получено лишь для ограниченного класса линейных функций  $\varphi(\bar{x}_{i-1}, i)$ ,  $v(\bar{x}_{i-1}, i-1)$ . Полученные модели марковских случайных процессов используются для решения задач прогноза.

### **Авторегрессионные модели случайных последовательностей**

Пусть последовательность случайных величин удовлетворяет стохастическому уравнению

$$x_i = \varphi(x_{i-1}, \xi_i), \quad i = 1, 2, \dots, \quad (3.33)$$

с начальным условием  $x_0 = \varphi_0(\xi_0)$ , где  $\varphi$  и  $\varphi_0$  – некоторые функции;  $\xi = (\xi_0, \xi_1, \xi_2, \dots)$  – заданная последовательность независимых случайных величин, называемая порождающей или возмущающей последовательностью.

Уравнение (3.33) представляет собой простейшую авторегрессионную модель случайной последовательности  $X = \{x_0, x_1, x_2, \dots\}$  [43].

В качестве возмущающей последовательности чаще всего используется последовательность стандартных независимых гауссовских случайных величин. В случае линейности функции  $\varphi$  процесс  $X$  также будет гауссовским, а при соответствующем выборе нелинейных функций и негауссовских возмущений можно получить широкий класс негауссовских процессов [11, 56].

Примером марковской случайной последовательности может быть *процесс авторегрессии 1-го порядка*, полученный с помощью линейного преобразования последовательности  $\xi_1, \xi_2, \dots, \xi_i, \dots$  независимых гауссовских случайных величин  $\{\xi_i\}$  по следующему правилу:

$$x_i = \rho x_{i-1} + \xi_i, \quad i = 2, 3, \dots, \quad (3.34)$$

где  $|\rho| < 1$ ;  $M\{\xi_i\} = 0$ ;  $M\{\xi_i^2\} = \sigma_\xi^2$ . Каждое очередное значение  $x_i$  содержит часть предыдущего  $\rho x_{i-1}$  и добавку в виде независимой случайной величины  $\xi_i$  [7, 14].

На рис. 3.6 представлены типичные графики реализаций такого процесса при различных значениях параметра  $\rho$ , входящего в модель (3.34). Во всех случаях параметр  $\sigma$ , влияющий только на масштаб по оси ординат, выбран равным единице. Из этих рисунков видно, что при  $\rho$ , близких к единице, процесс становится более гладким; при малых  $\rho$ , напротив, значения процесса слабо зависят между собой; при отрицательных  $\rho$  корреляция между соседними значениями процесса отрицательна, поэтому он часто меняет знак.

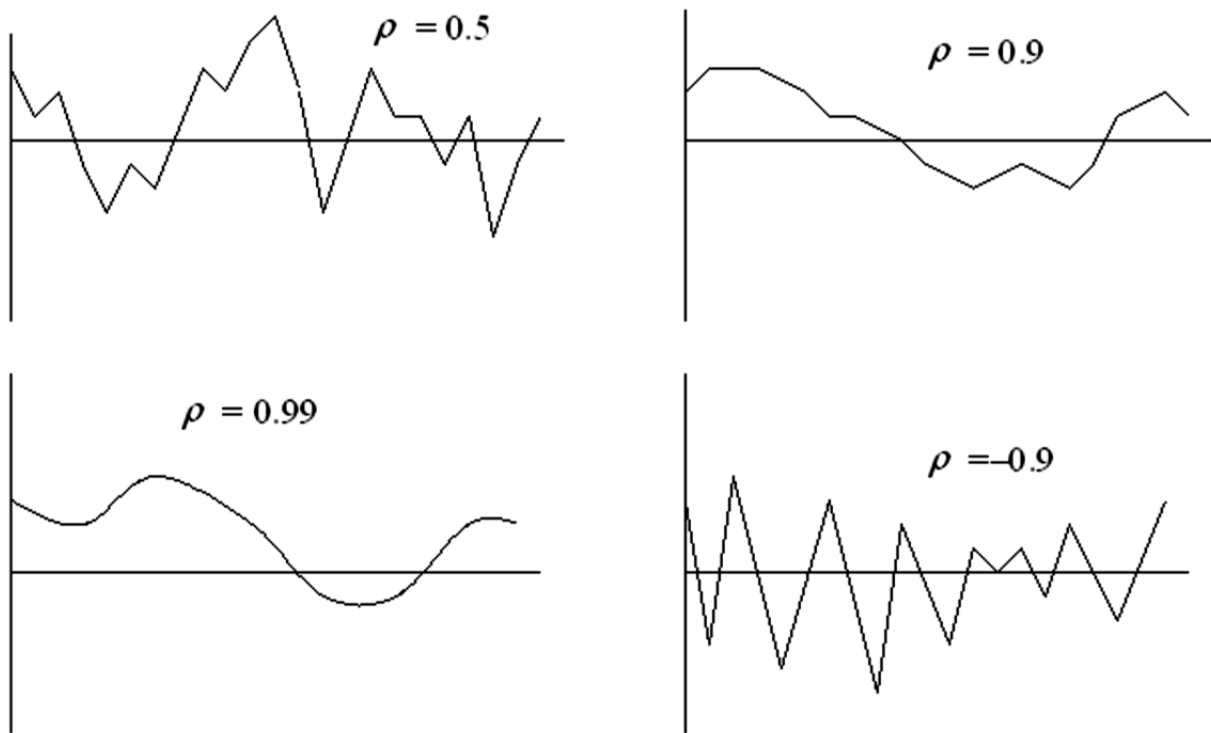


Рис. 3.6. Процессы авторегрессии первого порядка

При выборе начального значения  $x_1$ , обеспечивающего стационарность и постоянство дисперсии  $M\{x_i^2\} = \sigma_x^2$ , параметр  $\rho$  равен коэффициенту корреляции между любыми двумя соседними значениями случайной последовательности. Действительно, умножая левую и правую часть (3.34) на  $x_{i-1}$  и находя математическое ожидание, получим  $M\{x_i x_{i-1}\} = \rho \sigma_x^2$  или  $\rho = M\{x_i x_{i-1}\} / \sigma_x^2$ .



Повторяя аналогичные операции после подстановки в уравнение (3.34)  $x_{i-1} = \rho x_{i-2} + \xi_{i-1}$ ,  $x_{i-2} = \rho x_{i-3} + \xi_{i-2}$ , ..., можно записать следующую формулу для корреляционной функции:

$$R_x(k) = M\{x_i x_{i-k}\} / \sigma_x^2 = e^{-\alpha|k|},$$

где  $\alpha = -\ln \rho$ ;  $\sigma_x^2 = \sigma_\xi^2 / (1 - \rho^2)$ .

Таким образом, случайная последовательность (3.26) имеет экспоненциальную корреляционную функцию. В то же время случайная последовательность (3.34) является марковской, поскольку любые вероятностные характеристики значения  $x_i$  полностью определяются только предшествующим значением случайной последовательности  $x_{i-1}$ . При заданном  $x_{i-1}$  формула (3.34) позволяет найти все характеристики  $x_i$  без учёта предыстории, т.е. значений  $x_1, x_2, \dots, x_{i-2}$  случайной последовательности. Так, условная плотность распределения вероятности

$$f\left(\frac{x_i}{x_{i-1}}\right) = \frac{1}{\sqrt{2\pi}\sigma_\xi} \exp\left(-\frac{1}{2\sigma_\xi^2}(x_i - \rho x_{i-1})^2\right)$$

может быть получена из плотности распределения вероятности  $f(\xi_i) = (1/\sqrt{2\pi}\sigma_\xi) \exp(-\xi_i^2/2\sigma_\xi^2)$  с учётом связи  $\xi_i = x_i - \rho x_{i-1}$  и правил нахождения плотности распределения вероятности функций случайной величины. Легко записать выражение и для совместного распределения произвольного числа  $n$  членов рассмотренной марковской случайной последовательности:

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= f(x_1) \prod_{i=2}^n f\left(\frac{x_i}{x_{i-1}}\right) = \\ &= f(x_1) \left(\frac{1}{\sqrt{2\pi}\sigma_\xi}\right)^{n-1} \exp\left(-\frac{1}{2\sigma_\xi^2} \sum_{i=2}^n (x_i - \rho x_{i-1})^2\right) \end{aligned} \quad (3.35)$$

Поскольку вид всех плотностей распределения вероятности перехода (3.34) не зависит от номера члена случайной последовательности, то уравнение (3.34) представляет однородную марковскую случайную последовательность. Для стационарности необходимо выбрать случайную величину  $x_1$  таким образом, чтобы все безусловные плотности распределения вероятности  $f(x_i)$ ,  $i = 1, 2, \dots, n$  были одинаковыми. Анализ (3.34) показывает, что в стационарном случае все члены  $\{x_i\}$  последовательности имеют нулевое среднее и дисперсию  $\sigma_x^2 = \sigma_\xi^2 / (1 - \rho^2)$ . Кроме того, случайная последовательность  $\{x_i\}$  гауссовская, так как получена в результате линейного преобразования гауссовских случайных величин  $\{\xi_i\}$ . Таким образом, плотности распределения вероятности всех значений стационарной последовательности (3.34) будут иметь следующий вид:

$$f(x_i) = (1/\sqrt{2\pi}\sigma_x) \exp(-x_i^2/2\sigma_x^2), \quad i = 1, 2, \dots, n.$$

При этом начальное значение  $x_1$  формируется как нормальная случайная величина с нулевым средним и дисперсией  $\sigma_x^2$ , а последующие члены последовательности образуются в соответствии с рекуррентным соотношением (3.34).

Уравнения вида (3.34), которые часто называются *уравнениями авторегрессии* или *стохастическими разностными уравнениями*, представляют весьма узкий класс гауссовских марковских случайных последовательностей с экспоненциальной корреляционной функцией. Вместе с тем имеются различные возможности для существенного расширения этого класса [7, 14, 43]. Одной из них является описание случайной последовательности с помощью авторегрессионных уравнений более высокого порядка:

$$x_i = \rho_1 x_{i-1} + \rho_2 x_{i-2} + \dots + \rho_m x_{i-m} + \xi_i, \quad i = 2, 3, \dots, n, \quad (3.36)$$

где  $m$  – порядок авторегрессии.

С помощью подбора коэффициентов  $\rho_1, \rho_2, \dots, \rho_m$  можно получить гауссовские случайные последовательности  $x_i$ ,  $i = 1, 2, \dots, n$ , с разнообразными корреляционными свойствами [7]. Действительно, умножая (3.35) на  $x_{i-k}$  и находя математические ожидания, получим после деления на  $\sigma_x^2 = M\{x_i^2\}$ ,  $i = 1, 2, \dots, n$ , следующее соотношение для значений корреляционной функции:

$$R_x(k) = \rho_1 R_x(k-1) + \rho_2 R_x(k-2) + \dots + \rho_m R_x(k-m), \quad k > 0. \quad (3.37)$$

Общее решение этого разностного уравнения в стационарном случае представляется суммой экспонент [63]:

$$R_x(k) = A_1 e^{\alpha_1 |k|} + A_2 e^{\alpha_2 |k|} + \dots + A_m e^{\alpha_m |k|},$$

где  $\alpha_v = \ln z_v$ ;  $z_v$ ,  $v = 1, 2, \dots, m$  – корни характеристического уравнения

$$z^m - \rho_1 z^{m-1} - \rho_2 z^{m-2} - \dots - \rho_m = 0.$$

Требование стационарности случайной последовательности (3.36) выполняется, если  $\alpha_v < 0$ , т. е. когда все корни  $z_v$ ,  $v = 1, 2, \dots, m$  характеристического уравнения лежат внутри единичного круга на комплексной плоскости.

### Модели авторегрессии скользящего среднего

Стохастический линейный процесс можно представить как выходной сигнал линейного фильтра, на вход которого поступает белый шум  $\xi_i$  (рис. 3.7)

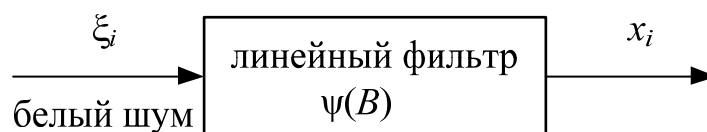


Рис. 3.7. Представление временного ряда с помощью линейного фильтра

Модель авторегрессии (3.36) выражает отсчёт  $x_i$  процесса в виде конечной взвешенной суммы  $n$  предыдущих отсчётов процесса  $x_{i-1}, x_{i-2}, \dots, x_{i-n}$  плюс случайный отсчёт  $\xi_i$ . Другой тип моделей, имеющий большое значение в описании случайных процессов, – это так называемый *процесс скользящего среднего*. Пусть  $x_i$  линейно зависит от конечного числа  $m$  предыдущих отсчётов  $\xi$ :

$$x_i = \xi_i - \theta_1 \xi_{i-1} - \theta_2 \xi_{i-2} - \dots - \theta_m \xi_{i-m}. \quad (3.38)$$

Такой процесс называется процессом скользящего среднего порядка  $m$ . Заметим, что веса  $1, -\theta_1, -\theta_2, \dots, -\theta_m$ , на которые умножаются  $\xi$ , не обязаны давать в сумме единицу или хотя бы быть положительными [7].

Если определить оператор скользящего среднего порядка  $m$  как  $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_m B^m$ , то модель скользящего среднего можно сжато записать как  $x_i = \theta(B) \xi_i$ . Она содержит  $m + 2$  неизвестных параметра:  $\mu, \theta_1, \dots, \theta_m, \sigma_\xi^2$ , которые на практике должны оцениваться по наблюдениям.

Для достижения большей гибкости в подгонке моделей к наблюдаемым временным рядам иногда целесообразно объединить в одной модели и авторегрессию, и скользящее среднее. Это приводит к комбинированной модели авторегрессии – скользящего среднего [7]

$$x_i = \rho_1 x_{i-1} + \dots + \rho_p x_{i-p} + \xi_i - \theta_1 \xi_{i-1} - \dots - \theta_q \xi_{i-q}$$

или  $\rho(B)x_i = \theta(B)\xi_i$ , в которой имеется  $p+q+2$  неизвестных параметра:  $\mu; \rho_1, \dots, \rho_p; \theta_1, \dots, \theta_q; \sigma_\xi^2$ , оцениваемых по наблюдениям.

На практике часто оказывается, что адекватное описание наблюдаемых временных рядов достигается при помощи моделей авторегрессии, скользящего среднего или комбинированной модели, в которых  $p$  и  $q$  не больше, а часто и меньше 2 [11].

### 3.2.2. Методы моделирования случайных процессов

Большой класс СП, имеющих место в информационно-измерительных системах, системах автоматического управления, подверженных воздействию случайных возмущений, описывается с помощью дифференциальных уравнений вида

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t, \mathbf{x}(t)), \quad \mathbf{y}(0) = \mathbf{y}_0, \quad t \in [0, T], \quad (3.39)$$

где  $\mathbf{x}(t) = (x_1(t), \dots, x_m(t))^T$  – векторный стационарный случайный процесс;

$\mathbf{y} = (y_1, \dots, y_n)^T$  – вектор состояния системы.

При исследовании на ЭВМ системы (3.39) необходимо получать реализации случайного процесса. Методы моделирования случайных величин рассматривались в подразд. 3.1. Ниже приводятся некоторые распространённые на прак-

тике методы моделирования гауссовских стационарных случайных процессов: метод формирующего фильтра, метод скользящего суммирования и рекуррентные моделирующие алгоритмы.

При имитации системы (3.39) на ЭВМ осуществляется переход от непрерывной системы к её дискретной модели. Как правило, используются численные методы, входящие в математическое обеспечение ЭВМ (например, метод Рунге–Кутты и его модификации) [71]. При этом возникают методические ошибки, в том числе и при получении реализаций случайного процесса. Величина ошибок определяется выбранным шагом интегрирования  $h$ .

Для линейных стационарных систем, находящихся под воздействием гауссовских стационарных случайных возмущений, могут быть получены алгоритмы моделирования, лишённые методических ошибок. Алгоритмы, основанные на методе дискретизации линейных стохастических уравнений, рассмотрены ниже. Метод дискретизации даёт сравнительно простые и легко реализуемые алгоритмы моделирования гауссовских векторных и скалярных случайных процессов с дробно-рациональным спектром высокого порядка. Помимо задач цифрового моделирования алгоритмы дискретизации оказываются полезными при расчётах корреляционных характеристик линейных систем и применении методов оптимальной фильтрации к обработке случайных процессов.

В настоящее время разработан ряд методов моделирования гауссовских стационарных случайных процессов  $x(t)$  с заданными характеристиками: математическим ожиданием  $m_x$ , корреляционной функцией  $R_x(\tau)$  или спектральной плотностью  $S_x(\omega)$ . При решении задач моделирования в целях удобства зачастую считают математическое ожидание нулевым, а дисперсию  $\sigma_x^2$  – единичной.

Использование преобразования

$$x(t) = m_x + \sigma_x \overset{\circ}{x}(t)$$

позволяет получить процессы с требуемыми значениями этих характеристик.

Здесь  $M\left[\overset{\circ}{x}(t)\right] = 0$ ,  $\sigma^2\left[\overset{\circ}{x}(t)\right] = 1$ . Как известно, любая из функций  $R_x(\tau)$  или  $S_x(\omega)$  описывает полностью рассматриваемый класс процессов. Обе характеристики связаны взаимно однозначно преобразованиями Фурье [73]:

$$R_x(\tau) = \int_{-\infty}^{\infty} S_x(\omega) e^{j\omega\tau} d\omega, \quad (3.40)$$

$$S_x(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} R_x(\tau) e^{-j\omega\tau} d\tau, \quad (3.41)$$

Обычно задача формулируется следующим образом. По известным характеристикам процесса (математическому ожиданию, дисперсии и корреляционной функции или спектральной плотности) требуется построить вычислительный алгоритм, позволяющий получать на ЭВМ реализации случайного процесса  $x(t)$  или последовательностей  $x_k, k = 0, 1, 2, \dots$ . В гауссовском случае модель процесса, заданная математическим ожиданием и корреляционной функцией, является полностью определённой. Шаг дискретизации  $\Delta t$  может быть не равен шагу интегрирования  $h$  системы (3.39).

Известные методы можно разбить на две большие группы: *точные* (метод рекуррентных алгоритмов дискретизации) и *приближённые* (методы формирующего фильтра, скользящего суммирования). В точных методах отсутствует методическая ошибка по корреляционной функции, т. е. корреляционная функция  $R_x[l] = M[x_{k+l}x_k]$  последовательности  $x_k$  равна дискретным значениям  $R_x(k\Delta t)$  корреляционной функции  $R_x(\tau)$  моделируемого процесса с непрерывным временем [76].

Для приближённых методов равенство заданных и воспроизводимых на ЭВМ характеристик выдерживается не точно, с некоторой погрешностью [67, 41]. В настоящее время практически отсутствуют работы по анализу погрешностей приближённых методов моделирования, поэтому основным и наиболее надёжным способом контроля приближённых алгоритмов является статистическая обработка моделируемых реализаций [15, 56, 64, 75].

### Метод формирующего фильтра

Формирующим фильтром называют динамическую систему, преобразующую случайный процесс  $\xi(t)$  вида белого шума в случайный процесс  $x(t)$  с заданными статистическими характеристиками [4, 76]. Белый шум представляет собой стационарный случайный процесс с постоянной спектральной плотностью  $S_0$ . Его корреляционная функция имеет вид

$$R_x(\tau) = 2\pi S_0 \delta(\tau),$$

где  $\delta(\tau)$  – дельта-функция Дирака, определяемая соотношениями

$$\delta(\tau) = \begin{cases} \infty, & \tau = 0, \\ 0, & \tau \neq 0, \end{cases} \quad \int_0^\infty \delta(\tau) d\tau = 1.$$

Полагаем процесс  $\xi(t)$  гауссовским, нормированным условием  $S_0 = 1/2\pi$ ,  $M[\xi(t)] = 0$ . Чтобы найти передаточную функцию формирующего фильтра  $\Phi(p)$ , спектральную плотность процесса  $x(t)$  представляют в виде произведения двух комплексно сопряжённых сомножителей:

$$S_x = S(j\omega)S(-j\omega). \quad (3.42)$$

Формирующий фильтр с передаточной функцией  $\Phi(p) = S(p)/\sqrt{S_0}$  должен быть устойчивым. Отметим, что моделируемый процесс является стационарным с заданной спектральной плотностью лишь при  $t \rightarrow \infty$ . Для дробно-рациональной спектральной плотности  $S_x(\omega)$  функция  $\Phi(p)$  имеет вид

$$\Phi(p) = \frac{1}{\sqrt{S_0}} \frac{F_m(p)}{H_n(p)}. \quad (3.43)$$

где  $F_m(p), H_n(p)$  – полиномы степени  $m, n, m < n$ . Ей соответствует дифференциальное уравнение, записанное в операторной форме:

$$H_n(D)x(t) = \sqrt{S_0} F_m(D)\xi(t), \quad D = d/dt. \quad (3.44)$$

От этого уравнения с помощью известных преобразований легко перейти к системе дифференциальных уравнений первого порядка [76]. Стационарный случайный процесс  $x(t)$  может быть представлен первой компонентой  $n$ -мерного марковского процесса  $\mathbf{x}(t) = [x_1(t), \dots, x_n(t)]^T$ ,  $x_1(t) = x(t)$ , удовлетворяющего уравнению

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\xi(t). \quad (3.45)$$

Если  $\sqrt{2\pi}F_m(p) = \beta_0 p^m + \beta_1 p^{m-1} + \dots + \beta_m$ ,  $H_n(p) = p^n + \alpha_1 p^{n-1} + \dots + \alpha_n$ , то матрица  $\mathbf{A}$  и вектор  $\mathbf{B}$  равны

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -\alpha_n & -\alpha_{n-1} & -\alpha_{n-2} & \dots & -\alpha_1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \dots \\ 0 \\ b_{n-m} \\ \dots \\ b_{n-1} \\ b_n \end{bmatrix}, \quad (3.46)$$

где  $b_{n-m} = \beta_0$ ,

$$b_k = \beta_{k-(n-m)} - \sum_{j=1}^{k-(n-m)} \alpha_j b_{k-j}, \quad k = n-m+1, \dots, n. \quad (3.47)$$

Процесс  $x(t)$  – гауссовский с нулевым средним. После окончания переходного процесса в уравнении (3.45) корреляционная матрица  $\Gamma = M[\mathbf{x}\mathbf{x}^T]$ , установившегося стационарного процесса находится из уравнения:

$$\mathbf{A}\Gamma + \Gamma\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = 0. \quad (3.48)$$

Белый шум с бесконечно большой дисперсией является абстрактным физически нереализуемым процессом. Для моделирования формирующего фильтра на ЭВМ разработаны различные способы, требующие предварительных вычислений [67].

Приведём приближённый и достаточно простой метод интегрирования на цифровой ЭВМ уравнений формирующего фильтра [15, 76]. На ЭВМ моделируется дискретный белый шум  $\xi_k \sim N(0,1)$ ,  $k=0,1,2,\dots$ ,  $\xi_k$  – некоррелированы. Рассмотрим ступенчатый процесс  $\xi_h(t)$  с шагом  $h$ , порождаемый дискретным белым шумом

$$\xi_k(t) = \xi_k, \quad t \in [kh, (k+1)h].$$

Спектральная плотность процесса  $\xi_h(t)$  равна

$$S_h(\omega) = \frac{h}{2\pi} \left[ \frac{\sin(\omega h/2)}{\omega h/2} \right]^2.$$

При  $h \rightarrow 0$  и фиксированном диапазоне частот  $\omega \in [0, \omega_0]$  функция  $S_h(\omega)$  стремится к постоянной спектральной плотности, причём максимальное по  $\omega$  отклонение достигается на конце промежутка при  $\omega = \omega_0$ . Относительная погрешность в имитации процессом  $\xi_h(t)$  свойств белого шума характеризуется величиной

$$\frac{|S_h(0) - S_h(\omega_0)|}{S_h(0)} \leq \varepsilon_*, \quad (3.49)$$

где  $\varepsilon_*$  – заданное значение погрешности.

Из неравенства (3.49) получаем неравенство  $h \leq h_* = 2\sqrt{3\varepsilon_*}/\omega_0$ , позволяющее выбрать величину  $h$ .

Величина  $\omega_0$  определяет тот частотный диапазон, в пределах которого необходимо воспроизводить спектральную плотность  $S_x(\omega)$  моделируемого процесса. Значение  $\omega_0$  находится из условия

$$\sup_{\omega \in [0, \omega_0]} \{S_x(\omega)/S_x(0)\} < \varepsilon,$$

где  $\varepsilon > 0$  – заданная малая величина. При этом диапазон частот  $[0, \omega_0]$  должен перекрывать полосу пропускания системы, на вход которой подаётся процесс  $x(t)$ .

Уравнение формирующего фильтра при моделировании на ЭВМ, получается из формулы (3.44) при  $S_0 = h/2\pi$  и имеет вид

$$H_n(D)x(t) = \sqrt{\frac{2\pi}{h}} F_m(D) \xi_h(t). \quad (3.50)$$

где  $h$  – шаг интегрирования дифференциального уравнения формирующего фильтра. При моделировании на ЭВМ от этого уравнения следует перейти к системе дифференциальных уравнений первого порядка.

Векторному уравнению (3.45) соответствует уравнение

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \frac{1}{\sqrt{h}} \mathbf{B} \xi_h(t). \quad (3.51)$$

Начальные условия задаются такими, чтобы можно было исключить переходный процесс  $\mathbf{x}(0) \sim N(0, \Gamma)$ ,  $\Gamma$  – корреляционная матрица, определяемая из уравнения (3.48). При нулевых начальных условиях следует отбросить начальный отрезок реализации длиной  $T_0$ . Для процессов с типовыми нормированными корреляционными функциями  $\rho_x(\tau)$  величина  $T_0$  приближённо равна  $T_0 \approx (1 \dots 3)\tau_k$ , где  $\tau_k$  – интервал корреляции процесса. Величина  $\tau_k$  определяется условием  $|\rho_x(\tau_k)| = 0,05$ , где в случае неоднозначности в качестве  $\tau_k$  берется наибольший из корней уравнения [76].

При малых  $h$  корреляционные функции процессов (3.45) и (3.51), а также (3.44) и (3.50) приближённо равны. При  $h \rightarrow 0$  методическая ошибка стремится к нулю. Отметим, что при интегрировании системы дифференциальных уравнений методом Рунге–Кутты на каждом шаге несколько раз вычисляются правые части. Составляя программы, следует предусмотреть, чтобы при вычислении правых частей использовалась одна и та же для данного шага интегрирования случайная величина  $\xi_k$ . Приведённый метод, основанный на моделировании посредством цифровой ЭВМ формирующего фильтра, обладает методической ошибкой, величина которой уменьшается при  $h \rightarrow 0$ .

### Метод скользящего суммирования

При использовании данного метода, дискретные значения моделируемого процесса формируются в виде скользящей суммы

$$x_k = \sum_{j=-M}^M \alpha_j \xi_{k-j}, \quad \xi_k \sim N(0,1) \quad (3.52)$$

с весовыми коэффициентами  $\alpha_j$ . Существует ряд способов определения  $\alpha_j$  [4, 76].

Один из них основан на применении интеграла свёртки

$$x(t) = \int_{-\infty}^{\infty} g(\tau) \xi(t-\tau) d\tau, \quad (3.53)$$

где  $\xi(t)$  – нормированный белый шум;

$g(\tau)$  – весовая функция формирующего фильтра.

Функция  $g(t)$  определяется формулой

$$g(t) = \sqrt{\frac{2}{\pi}} \int_0^{\infty} \sqrt{S_x(\omega)} e^{j\omega t} d\omega. \quad (3.54)$$

Формирующий фильтр с весовой функцией (3.54) имеет вещественную частотную характеристику  $G(\omega) = \sqrt{2\pi S_x(\omega)}$ . Соответствующая весовая функция (3.54) чётна, поэтому непрерывный линейный фильтр с такой весовой функцией



ей физически не реализуем. Однако это свойство не является препятствием для цифрового моделирования. Дискретизация интеграла (3.53) с шагом  $\Delta t$  даёт следующие значения весовых коэффициентов:

$$\alpha_j = \sqrt{\Delta t} g(j\Delta t), \quad j = 0, \pm 1, \dots, \pm M.$$

Значения  $g(j\Delta t)$  вычисляются, как правило, с помощью численных методов [71]. При этом бесконечный верхний предел интегрирования в формуле (3.54) заменяют на конечный. Генерируемая последовательность  $x_k$  имеет корреляционную функцию, равную

$$R_x[l] = \sum_{j=-M}^{M-l} a_j a_{l+j} = \sum_{j=-M}^{M-l} g(j\Delta t) g((j+l)\Delta t) \Delta t. \quad (3.55)$$

Истинная корреляционная функция имеет вид

$$R_x(\tau) = \int_{-\infty}^{\infty} g(t) g(t+\tau) dt. \quad (3.56)$$

Функция  $R_x[l]$  является интегральной суммой для интеграла (3.56). При условиях  $\Delta t \rightarrow 0$ ,  $l\Delta t = \tau = \text{const}$ , корреляционная функция последовательности  $x_k$  стремится к требуемой  $R_x(\tau)$ . Контроль правильности вычисления  $a_j$  и выбора числа членов  $M$  (теоретически должно быть  $M = \infty$ ) осуществляется путём расчёта по формуле (3.41) функции  $R_x[l]$  и сравнением её с требуемой корреляционной функцией. Поскольку последовательность (3.38) является гауссовской, то близость функций  $R_x[l]$  и  $R_x[\tau]$  означает близость заданного и моделируемого процессов на уровне конечномерных распределений. Метод скользящего суммирования пригоден для моделирования гауссовских процессов с произвольными спектральными плотностями.

### 3.3. Моделирование случайных полей

#### 3.3.1. Основные понятия и подходы моделирования случайных полей

Случайными полями называются случайные функции многих переменных. В дальнейшем будут рассматриваться четыре переменные: координаты  $x, y, z$ , определяющие положение точки в пространстве, и время  $t$ . Случайное поле будет обозначаться как  $\xi(\mathbf{r}, t) = \xi(x, y, z, t)$ . Случайные поля могут быть скалярными (одномерными) и векторными ( $N_i$ -мерными).

В общем случае скалярное поле  $\xi(\mathbf{r}, t)$  задаётся совокупностью своих  $N$ -мерных распределений

$$\omega(\xi_1, \dots, \xi_N) \partial \xi_1, \dots, \partial \xi_N = P\{\xi_v \leq \xi(\mathbf{r}_v, t_v) < \xi_v + \partial \xi_v\}, \quad v = \overline{1, N}, \quad (3.57)$$

а векторное поле  $\|\xi_k(\mathbf{r}, t)\|$ ,  $k = \overline{1, N_1}$  – совокупностью своих  $N \times N_1$ -мерных распределений

$$\begin{aligned} & \omega(\xi_{11}, \dots, \xi_{1N}; \xi_{21}, \dots, \xi_{2N}; \dots; \xi_{N_11}, \dots, \xi_{N_1N}) \partial \xi_{11}, \dots, \partial \xi_{N_1N} = \\ & = P\{\xi_{kv} \leq \xi_k(\mathbf{r}_v, t_v) < \xi_{kv} + \partial \xi_{kv}\}, \quad v = \overline{1, N}, \quad k = \overline{1, N_1}. \end{aligned} \quad (3.58)$$

Если статистические характеристики поля не изменяются при изменении начала отсчёта времени, т. е. они зависят только от разности  $\tau = t_2 - t_1$ , то такое поле называется стационарным. Если перенос начала координат не влияет на статистические характеристики поля, т. е. они зависят только от разности  $\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$ , то такое поле называется однородным по пространству. Однородное поле изотропно, если его статистические характеристики не изменяются при изменении направления вектора  $\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$ , т. е. зависят лишь от длины  $\rho = |\mathbf{r}|$  этого вектора [11].

Примерами случайных полей являются электромагнитное поле при распространении электромагнитной волны в статистически неоднородной среде, в частности электромагнитное поле сигнала, отражённого от флуктуирующей цели (это, вообще говоря, векторное случайное поле); объёмные диаграммы направленности антенн и диаграммы вторичного излучения целей, на формирование которых оказывают влияние случайные параметры; статистически неровные поверхности, в частности земная поверхность и поверхность моря при волнениях, и ряд других примеров.

В данном подразделе рассматриваются некоторые вопросы моделирования случайных полей на ЭВМ. Под задачей моделирования понимается разработка алгоритмов для формирования на ЭВМ дискретных реализаций поля, т. е. совокупностей выборочных значений поля.

$$\xi(\mathbf{r}_m, t_n) = \xi(x_i, y_j, z_k, t_n), \quad (3.59)$$

где  $\mathbf{r}_m = (x_i, y_j, z_k)$  – дискретная пространственная координата;

$t_n$  – дискретное время.

При этом полагается, что исходными при моделировании случайного поля являются независимые случайные числа. Совокупность таких чисел будет рассматриваться как случайное  $\delta$ -коррелированное поле, называемое в дальнейшем  $\delta$ -полем. Случайное  $\delta$ -поле – это элементарное обобщение дискретного белого шума на случай нескольких переменных. Моделирование  $\delta$ -поля на ЭВМ осуществляется весьма просто: пространственно-временной координате  $(\mathbf{r}_m, t_n)$  ставится в соответствие выборочное значение  $x_\delta(\mathbf{r}_m, t_n)$  числа из датчика нормальных случайных чисел с параметрами  $(0, 1)$ .

Задача цифрового моделирования случайных полей является новой в общей проблеме разработки системы эффективных алгоритмов для имитации различного рода случайных функций, ориентированной на решение статистических задач радиотехники, радиофизики, акустики и т. д. методом моделирования на ЭВМ.

В самом общем виде, если известен  $N$  или  $N \times N_1$ -мерный закон распределения, случайное поле можно моделировать на ЭВМ как случайный  $N$  или  $N \times N_1$ -мерный вектор. Однако ясно, что этот путь даже при сравнительно небольшом числе дискретных точек по каждой координате является очень сложным. Например, моделирование плоского (не зависящего от  $z$ ) скалярного случайного поля в 10 дискретных точках по координатам  $x$  и  $y$  и для 10 моментов времени сводится к формированию на ЭВМ реализаций 1000-мерного случайного вектора.

Упрощения алгоритма и сокращения объёма вычислений можно достичь, если разрабатывать алгоритмы для моделирования специальных классов случайных полей.

Рассмотрим возможные алгоритмы моделирования стационарных однородных скалярных нормальных случайных полей. Случайные поля этого класса так же, как и стационарные нормальные случайные процессы, играют важную роль в приложениях [11]. Такие поля полностью задаются своими пространственно-временными корреляционными функциями

$$R(\mathbf{p}, \tau) = M \{ \xi(\mathbf{r}, t) \xi(\mathbf{r} + \mathbf{p}, t + \tau) \}. \quad (3.60)$$

Здесь и в дальнейшем предполагается, что среднее значение поля равно нулю.

Столь же полной характеристикой рассматриваемого класса случайных полей является функция спектральной плотности поля  $G(\mathbf{s}, \omega)$ , представляющая собой четырёхмерное преобразование Фурье от корреляционной функции  $R(\mathbf{p}, \tau)$  (обобщение теоремы Винера – Хинчина [11]):

$$G(\mathbf{s}, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R(\mathbf{p}, \tau) e^{-j(\mathbf{s}\mathbf{p} + \omega\tau)} d\mathbf{p} d\tau, \quad (3.61)$$

где  $\mathbf{s}\mathbf{p}$  – скалярное произведение векторов  $\mathbf{s} = (s_x, s_y, s_z)$  и  $\mathbf{p} = (p_x, p_y, p_z)$ ;

$$d\mathbf{p} = dp_x dp_y dp_z.$$

При этом

$$R(\mathbf{p}, \tau) = \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(\mathbf{s}, \omega) e^{j(\mathbf{s}\mathbf{p} + \omega\tau)} d\mathbf{s} d\omega. \quad (3.62)$$

Функция спектральной плотности  $G(\mathbf{s}, \omega)$  случайного поля и энергетический спектр  $G(\omega)$  стационарного случайного процесса имеют аналогичный смысл, а именно: если случайное поле  $\xi(\mathbf{r}, t)$  представить в виде суперпозиции про-

пространственно-временных гармоник со сплошным спектром частот, то интенсивность их (суммарная дисперсия амплитуд) в полосе частот  $(\omega, \omega + \partial\omega)$  и полосе пространственных частот  $(\mathbf{s}, \mathbf{s} + \partial\mathbf{s})$  равна  $G(\mathbf{s}, \omega) \partial\mathbf{s} \partial\omega$ .

Случайное поле с интенсивностью  $G(\mathbf{s}, \omega) \partial\mathbf{s} \partial\omega$  можно получить из случайного поля  $\xi(\mathbf{r}, t)$ , имеющего спектральную плотность  $G(\mathbf{s}, \omega)$ , если пропустить поле  $\xi(\mathbf{r}, t)$  через пространственно-временной фильтр с коэффициентом передачи, равным единице в полосе  $(\omega, \omega + \partial\omega)$ ,  $(\mathbf{s}, \mathbf{s} + \partial\mathbf{s})$  и равным нулю вне этой полосы.

Пространственно-временные фильтры (ПВФ) являются обобщением обычных (временных) фильтров. Линейные ПВФ, как и обычные фильтры, описываются с помощью импульсной переходной характеристики [11]

$$h(\mathbf{r}, t) = \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} K(j\mathbf{s}, j\omega) e^{j(\mathbf{s}\mathbf{r} + \omega t)} \partial\mathbf{s} \partial\omega \quad (3.63)$$

и передаточной функции

$$K(j\mathbf{s}, j\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\mathbf{r}, t) e^{-j(\mathbf{s}\mathbf{r} + \omega t)} \partial\mathbf{s} \partial\omega. \quad (3.64)$$

Процесс линейной пространственно-временной фильтрации поля  $\xi_1(\mathbf{r}, t)$ , можно записать в виде четырёхмерной свёртки:

$$\begin{aligned} \xi_2(\mathbf{r}, t) &= \xi_1(\mathbf{r}, t) \cdot h(\mathbf{r}, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \xi_1(\mathbf{r} - \boldsymbol{\rho}, t - \tau) h(\boldsymbol{\rho}, \tau) \partial\boldsymbol{\rho} \partial\tau = \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \xi_1(\boldsymbol{\rho}, \tau) h(\mathbf{r} - \boldsymbol{\rho}, t - \tau) \partial\boldsymbol{\rho} \partial\tau, \end{aligned} \quad (3.65)$$

где  $\xi_2(\mathbf{r}, t)$  – поле на выходе ПВФ с импульсной переходной характеристикой  $h(\mathbf{r}, t)$ . При этом

$$G_2(\mathbf{s}, \omega) = G_1(\mathbf{s}, \omega) |K(j\mathbf{s}, j\omega)|^2, \quad (3.66)$$

$$R_2(\boldsymbol{\rho}, \tau) = R_1(\boldsymbol{\rho}, \tau) \cdot h(\boldsymbol{\rho}, \tau) \cdot h(-\boldsymbol{\rho}, -\tau), \quad (3.67)$$

где  $G_1(\mathbf{s}, \omega), G_2(\mathbf{s}, \omega), R_1(\boldsymbol{\rho}, \tau), R_2(\boldsymbol{\rho}, \tau)$  – функции спектральной плотности и корреляционные функции полей на входе и на выходе ПВФ соответственно.

Доказательство соотношений (3.66), (3.67) полностью совпадает с доказательством аналогичных соотношений для стационарных случайных процессов.

Аналогия гармонического разложения и фильтрации случайных полей с гармоническим разложением и фильтрацией случайных процессов позволяет предложить для их моделирования аналогичные алгоритмы.

Пусть требуется построить алгоритмы для моделирования на ЭВМ стационарного однородного по пространству скалярного нормального поля  $\xi(\mathbf{r}, t)$  с заданной корреляционной функцией  $R(\boldsymbol{\rho}, \tau)$  или функцией спектральной плотности  $G(\mathbf{s}, \omega)$ .

Если поле  $\xi(\mathbf{r}, t)$  задано в конечном пространстве, ограниченном пределами  $0 \leq x \leq X, 0 \leq y \leq Y, 0 \leq z \leq Z$ , и рассматривается на конечном интервале времени  $(0, T)$ , то для формирования на ЭВМ дискретных реализаций этого поля можно использовать алгоритм, основанный на каноническом разложении поля в пространственно-временной ряд Фурье [11]:

$$\xi(\mathbf{r}_m, t_n) = \sum_{k=0}^{\infty} A_k \cos(k s_1 \mathbf{r}_m + k \omega_1 t_n) + B_k \sin(k s_1 \mathbf{r}_m + k \omega_1 t_n). \quad (3.68)$$

Здесь  $A_k$  и  $B_k$  – случайные независимые между собой нормально распределённые числа с параметрами  $(0, \sigma_k^2)$  каждое, причём дисперсии  $\sigma_k^2$  определяются из соотношений

$$\begin{aligned} \sigma_0^2 &= \frac{1}{TXYZ} \int_0^R \int_0^T R(\mathbf{r}, \tau) d\mathbf{r} d\tau = \frac{1}{TXYZ} \int_0^X \int_0^Y \int_0^Z \int_0^T R(x, y, z, \tau) dx dy dz d\tau; \\ \sigma_k^2 &= \frac{1}{TXYZ} \int_0^R \int_0^T R(\mathbf{r}, \tau) \cos(k s_1 \mathbf{r} + k \omega_1 \tau) d\mathbf{r} d\tau, \quad k = 1, 2, \dots, \end{aligned} \quad (3.69)$$

где  $\mathbf{R}$  – вектор, изображающий предел интегрирования по пространству,  $\mathbf{R} = (X, Y, Z)$ ;

$k s_1, k \omega_1$  – дискретные частоты гармоник, по которым производится каноническое разложение корреляционной функции в пространственно-временной ряд Фурье,  $k s_1 = \left( \frac{k\pi}{X}, \frac{k\pi}{Y}, \frac{k\pi}{Z} \right)$ ,  $k \omega_1 = \frac{k\pi}{T}$ .

Если область разложения поля во много раз больше его пространственно-временного интервала корреляции, то дисперсии легко выражаются через спектральную функцию поля [11]

$$\sigma_0^2 = \frac{G(0, 0)}{2TXYZ}, \quad \sigma_k^2 = \frac{G(k s_1, k \omega_1)}{TXYZ}, \quad k = 1, 2, \dots \quad (3.70)$$

Формирование дискретных реализаций  $\xi(\mathbf{r}_m, t_n)$  при моделировании случайных полей по данному методу осуществляется путём непосредственного вычисления их значений по формуле (3.68), в которой в качестве  $A_k$  и  $B_k$  берутся выборочные значения нормальных случайных чисел с параметрами  $(0, \sigma_k^2)$ , при этом бесконечный ряд (3.68) приближённо заменяется усечённым рядом. Дисперсии  $\sigma_k^2$  вычисляются предварительно по формулам (3.69) или (3.70).

Рассмотренный алгоритм хотя и не позволяет формировать реализации случайного поля, неограниченные по пространству и по времени, однако подготовительная работа для его получения довольно простая, в особенности при использовании формул (3.70), и этот алгоритм позволяет формировать дискретные значения поля в произвольных точках пространства и времени выбранной

области. При формировании дискретных реализаций поля с постоянным шагом по одной или нескольким координатам для сокращенного вычисления тригонометрических функций целесообразно использовать рекуррентный алгоритм вида

$$\begin{aligned} u_1[n] &= cu_1[n-1] + su_2[n-1], u_1[0] = 0; \\ u_2[n] &= cu_2[n-1] + su_1[n-1], u_2[0] = 1, \end{aligned} \quad (3.71)$$

где  $c = \cos a$ ,  $s = \sin a$ .

Неограниченные дискретные реализации однородного стационарного случайного поля можно формировать с помощью алгоритмов пространственно-временного скользящего суммирования  $\delta$ -поля, аналогичных алгоритмам скользящего суммирования для моделирования случайных процессов. Если  $h(\mathbf{r}, t)$  – импульсная переходная характеристика ПВФ, формирующего из  $\delta$ -поля поле с заданной функцией спектральной плотности  $G(\mathbf{s}, \omega)$ , то, подвергая процесс пространственно-временной фильтрации  $\delta$ -поля дискретизации, получим

$$\xi[i, j, k, l] = \Delta \mathbf{r} \Delta t \sum_p \sum_q \sum_l \sum_m h[p, q, l, m] \cdot x_\delta[i - p, j - q, k - l, n - m], \quad (3.72)$$

где  $\Delta \mathbf{r} \Delta t = \Delta x \Delta y \Delta z \Delta t$  – константа, определяемая выбором шага дискретизации по всем переменным  $x, y, z, t$ ;

$x_\delta[i, j, k, m]$  – дискретное  $\delta$ -поле.

Суммирование в формуле (3.72) осуществляется по всем значениям  $p, q, l, m$ , при которых слагаемые не являются пренебрежимо малыми или равными нулю.

Подготовительная работа при данном методе моделирования заключается в нахождении соответствующей весовой функции  $h(\mathbf{r}, t)$  пространственно-временного формирующего фильтра.

Подготовительная работа и процесс суммирования в алгоритме (3.72) упрощаются, если функцию  $h(x, y, z, t)$  можно представить в виде произведения

$$h(x, y, z, t) = h_1(x)h_2(y)h_3(z)h_4(t). \quad (3.73)$$

В этом случае, как это следует из (3.69), корреляционная функция поля является произведением вида

$$R(x, y, z, \tau) = R_1(x)R_2(y)R_3(z)R_4(\tau). \quad (3.74)$$

где  $R_k(u) = h_k(u) \cdot \overline{h_k(-u)}$ ,  $k = \overline{1, 4}$ .

Если разложение корреляционной функции на множители вида (3.74) в строгом смысле невыполнимо, его можно сделать с некоторой степенью приближения, в частности, положив

$$R(x, y, z, \tau) = R(x, 0, 0, 0)R(0, y, 0, 0)R(0, 0, z, 0)R(0, 0, 0, \tau). \quad (3.75)$$

При разложении на произведение (3.75) пространственных корреляционных функций изотропных случайных полей, у которых  $R_*(\mathbf{p}, \tau) = R(\mathbf{p}, \tau)$ ,  $\rho = \sqrt{x^2 + y^2 + z^2}$ , частичные корреляционные функции  $R_1(x)$ ,  $R_2(y)$  и  $R_3(z)$  бу-

дут, очевидно, одинаковыми. При этом, ввиду приближённости формулы (3.75), пространственная корреляционная функция  $R_*(x, y, z) = R_1(x)R_2(y)R_3(z)$  будет соответствовать, вообще говоря, некоторому неизотропному случайному полю. Так, например, если  $R(\mathbf{p})$  является экспоненциальной функцией вида

$$R(\mathbf{p}) = R(x, y, z) = e^{-|\mathbf{p}|} = e^{-\sqrt{x^2+y^2+z^2}}, \quad (3.76)$$

то согласно (3.75)  $R_1(x) = e^{-|x|}$ ,  $R_2(y) = e^{-|y|}$ ,  $R_3(z) = e^{-|z|}$ . В этом случае заданная корреляционная функция  $R(\mathbf{p})$  аппроксимируется корреляционной функцией

$$R_*(x, y, z) = e^{-(|x|+|y|+|z|)}. \quad (3.77)$$

Случайное поле с корреляционной функцией (3.77) неизотропно. Действительно, если у поля с корреляционной функцией (3.76) поверхность постоянной корреляции (геометрическое место точек пространства, в которых значения поля имеют одинаковую корреляцию со значением поля в некоторой произвольной фиксированной точке пространства) является сферой, то в случае (3.77) поверхность постоянной корреляции есть поверхность куба, вписанного в указанную сферу. Максимальное расстояние между этими поверхностями может служить мерой погрешности аппроксимации.

Примером, в котором разложение (3.75) является точным, может служить корреляционная функция вида

$$R(x, y, z) = e^{-(a^2x^2+b^2y^2+c^2z^2)} = e^{-a^2x^2} e^{-b^2y^2} e^{-c^2z^2}. \quad (3.78)$$

Разложение (3.75) позволяет свести довольно сложный процесс четырёхкратного суммирования в алгоритме (3.72) к повторному применению однократного скользящего суммирования.

Таковы основные принципы моделирования нормальных однородных стационарных случайных полей. Моделирование ненормальных однородных стационарных полей с заданным одномерным законом распределения можно осуществить путём соответствующего нелинейного преобразования нормальных однородных стационарных полей.

В заключение следует заметить, что в этом подразделе были рассмотрены только основные принципы цифрового моделирования случайных полей и даны некоторые возможные моделирующие алгоритмы. Целый ряд вопросов остался незатронутым, например: моделирование векторных (в частности, комплексных), нестационарных, неоднородных, ненормальных случайных полей; вопросы нахождения весовой функции пространственно-временного формирующего фильтра по заданным корреляционно-спектральным характеристикам поля (в частности, возможность применения метода факторизации для многомерных спектральных функций); примеры применения цифровых моделей случайных полей при решении конкретных задач и т. д.

Изложение этих вопросов выходит за рамки учебного пособия. Многие из них являются предметом будущих исследований.

### 3.3.2. Математические модели поля изображения на основе стохастических дифференциальных уравнений в частных производных

Выбор математической модели, отражающей свойства реальных изображений различной природы (видео, инфракрасных, радиолокационных, гидроакустических и др.), является необходимым этапом при разработке устройств предварительной обработки таких изображений [36]. С одной стороны выбираемая модель должна быть достаточно простой, чтобы обеспечить возможность технической реализации устройства обработки, с другой – математическая модель должна как можно полнее отражать свойства реальных изображений, чтобы обеспечить высокое качество обработки.

Исходя из физических соображений плоское полутонное изображение можно представить в виде поля значений яркости. Такое поле можно задать функцией трёх переменных  $u(t, y, x)$ , где  $t$  – время;  $y, x$  – пространственные координаты. Значения функции  $u$  определяют меняющуюся во времени яркость изображения в каждой точке пространства.

Для описания математической модели поля изображения  $u(t, y, x)$  целесообразно воспользоваться аппаратом стохастических дифференциальных уравнений в частных производных, который непосредственно указывает способ генерации реализаций поля, а также позволяет производить анализ вероятностных характеристик получаемых моделей.

Исходя из вышеупомянутых логически обоснованных условий выбора модели изображения, на первом этапе целесообразно ограничиться широким классом хорошо изученных уравнений в частных производных математической физики [42, 70]. Безусловно, специфика отдельных случаев может выходить за рамки вводимого ограничения, что приведёт к постановке и решению новых краевых задач.

Исходя из физических соображений все изображения можно разделить на подвижные и неподвижные. Для неподвижных изображений зависимость функции  $u$  от времени  $t$  будет отсутствовать. В начале рассмотрим некоторые типы моделей неподвижных изображений, а затем – подвижных.

Известно, что к основным уравнениям математической физики относятся три типа линейных дифференциальных уравнений в частных производных второго порядка: гиперболические, параболические, эллиптические. Рассмотрим последовательно в качестве моделей неподвижных изображений краевые задачи для трёх основных типов уравнений математической физики и проанализируем вероятностные характеристики данных моделей.



### Гиперболическое уравнение

В [78] для описания моделей двумерных случайных полей предлагается использовать дифференциальные уравнения в частных производных гиперболического типа. Рассмотрим, как один из вариантов этих моделей, краевую задачу вида

$$\frac{\partial^2 u(y, x)}{\partial y \partial x} + a \frac{\partial u(y, x)}{\partial y} + b \frac{\partial u(y, x)}{\partial x} + abu(y, x) = n_u(y, x), \quad (3.79)$$

$$u(0, x) = g_1, \quad u(y, 0) = g_2,$$

где  $a > 0, b > 0; g_1, g_2 = \text{const}$ .

Здесь  $n_u(y, x)$  – белое гауссовское шумовое поле с известными характеристиками

$$M\{n_u(y, x)\} = 0, \quad (3.80)$$

$$M\{n_u(y_1, x_1)n_u(y_2, x_2)\} = \frac{N_u}{2}\delta(y_2 - y_1)\delta(x_2 - x_1), \quad (3.81)$$

где  $M\{\cdot\}$  – операция математического ожидания.

Для решения уравнения (3.79) воспользуемся методом замены переменных [78]. В результате решение краевой задачи (3.79) будет иметь вид

$$u(y, x) = g_2 e^{-ax} + g_1 e^{-by} (1 - e^{-ax}) + e^{-ax} e^{-by} \int_0^x \int_0^y e^{ax'} e^{by'} n_u(y', x') dy' dx'. \quad (3.82)$$

На рис. 3.8 приведены полутонные изображения реализаций гиперболической модели размером  $50 \times 50$  элементов, полученные в результате моделирования на ЭВМ выражения (3.82) для различных значений параметров  $a$  и  $b$  [45].

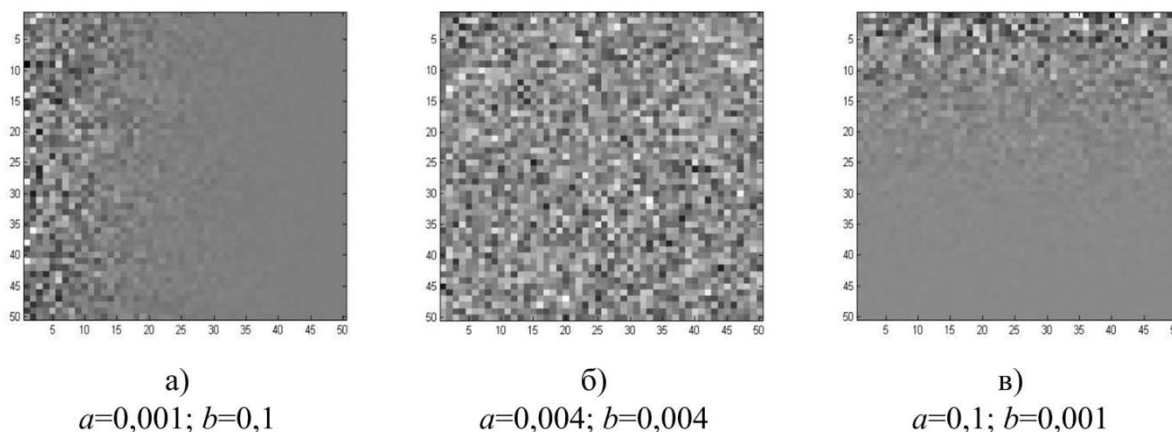


Рис. 3.8. Примеры реализаций гиперболической модели изображения

Поскольку входное воздействие  $n_u(y, x)$  является гауссовым, а уравнение (3.79) – линейным, то и поле  $u(y, x)$  также будет гауссовым. Следовательно, исчерпывающими статистическими характеристиками  $u(y, x)$  являются математическое ожидание и корреляционная функция.

С учётом (3.80) и (3.82) определим математическое ожидание поля  $u(y, x)$ :

$$m(y, x) = M\{u(y, x)\} = g_2 e^{-ax} + g_1 e^{-by} (1 - e^{-ax}). \quad (3.83)$$

Тогда корреляционная функция модели изображения будет определяться как

$$R_u(y_1, x_1, y_2, x_2) = M\{(u(y_1, x_1) - m(y_1, x_1))(u(y_2, x_2) - m(y_2, x_2))\}. \quad (3.84)$$

Подставив (3.82) в (3.84) и выполнив операцию усреднения, получим

$$R_u(y_1, x_1, y_2, x_2) = \frac{N_u}{8ab} (\exp[-b|y_1 - y_2|] - \exp[-b|y_1 + y_2|]) \times \\ \times (\exp[-a|x_1 - x_2|] - \exp[-a|x_1 + x_2|]) = R_y(y_1, y_2) R_x(x_1, x_2). \quad (3.85)$$

Из (3.85) видно, что корреляционная функция  $R_u(y_1, x_1, y_2, x_2)$  факторизуется и может быть представлена в виде произведения двух корреляционных функций по различным координатам, т. е. для данной модели значения поля по координатам  $y$  и  $x$  независимы. Кроме того, выражение (3.85) свидетельствует о наличии причинных связей между значениями поля по обеим координатам, так как корреляционная функция убывает с увеличением расстояния между значениями поля по каждой координате.

Таким образом, гиперболическое уравнение относится к классу каузальных моделей.

### Параболическое уравнение

Рассмотрим одну из краевых задач для уравнения параболического типа

$$\frac{\partial u(y, x)}{\partial y} - a \frac{\partial^2 u(y, x)}{\partial x^2} + bu(y, x) = n_u(y, x), \quad (3.86)$$

$$u(0, x) = g_1, u(y, 0) = g_2, u(y, L_c) = g_3,$$

где  $a > 0, b > 0, L_c$  – длина строки изображения.

Применив метод функций Грина [10], получили решение краевой задачи (3.86), которое определяется выражением

$$u(y, x) = \frac{2}{L_c} \sum_{m=1}^{\infty} \sin\left(\frac{m\pi x}{L_c}\right) \exp[-r_m y] \int_0^y \exp[r_m \eta] \int_0^{L_c} \sin\left(\frac{m\pi \xi}{L_c}\right) n_u(\xi, \eta) d\xi d\eta + \\ + \frac{2g_1}{\pi} \sum_{m=1}^{\infty} \sin\left(\frac{m\pi x}{L_c}\right) \exp[-r_m y] \frac{1 - (-1)^m}{m} + \\ + \frac{2a\pi}{L_c^2} \sum_{m=1}^{\infty} \sin\left(\frac{m\pi x}{L_c}\right) \frac{m(1 - \exp[-r_m y])}{r_m} (g_2 + g_3(-1)^m), \quad (3.87)$$

где  $r_m = b + \frac{am^2\pi^2}{L_c^2}$ .

На рис. 3.9 приведены полутоновые изображения реализаций параболической модели размером  $50 \times 50$  элементов, полученные в результате моделирования на ЭВМ выражения (3.87) для различных значений параметров  $a$  и  $b$  [45].

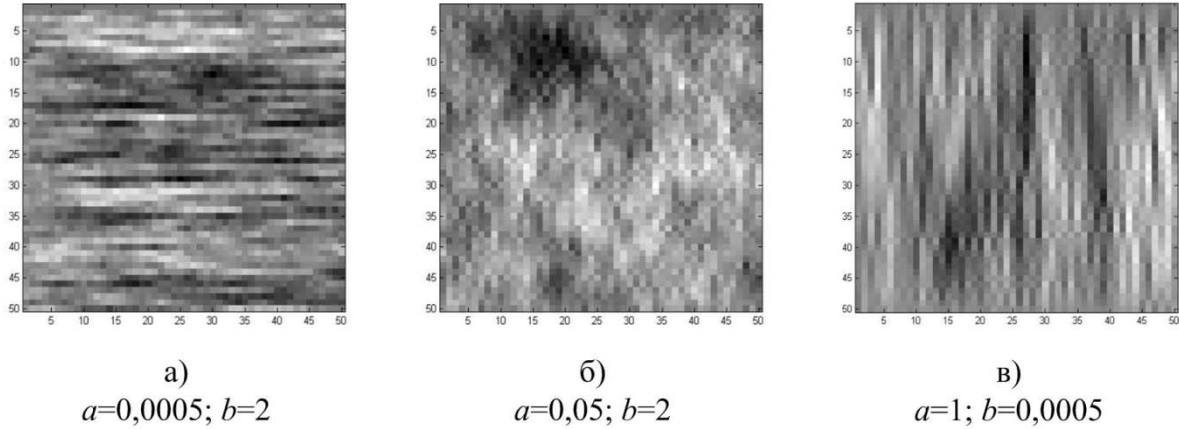


Рис. 3.9 Примеры реализаций параболической модели изображения

Определяя статистические характеристики, получим, что математическое ожидание поля равно сумме второго и третьего членов в правой части (3.87), а корреляционная функция будет иметь вид  $R_u(y_1, x_1, y_2, x_2) =$

$$= \frac{N_u}{2L_c^2} \sum_{m=1}^{\infty} \sin\left(\frac{m\pi x_1}{L_c}\right) \sin\left(\frac{m\pi x_2}{L_c}\right) \frac{\exp[-|y_2 - y_1|r_m] - \exp[-|y_2 + y_1|r_m]}{r_m}. \quad (3.88)$$

Из анализа (3.88) можно сделать вывод об отсутствии у параболических моделей причинных связей между значениями поля изображения по координате  $x$ , так как степень корреляционной зависимости значений поля определяется не столько расстоянием между этими значениями по координате  $x$ , сколько их взаимным расположением на этой координате. Следовательно, параболические уравнения относятся к классу полукаузальных моделей.

### Эллиптическое уравнение

Из широкого круга краевых задач для эллиптических уравнений рассмотрим краевую задачу, которая по своей физической сути может быть использована для описания плоских изображений

$$-\left[\frac{\partial^2 u(y, x)}{\partial y^2} + \frac{\partial^2 u(y, x)}{\partial x^2}\right] + au(y, x) = n_u(y, x), \quad (3.89)$$

$$u(0, x) = g_1, u(y, 0) = g_2, u(L_k, x) = g_3, u(y, L_c) = g_4,$$

где  $L_c$  и  $L_k$  – длина строки и длина кадра изображения соответственно.

Краевая задача (3.89) решена методом Фурье. В результате получено

$$u(y, x) = \frac{4}{L_c L_k} \sum_{m=1}^{\infty} \sum_{l=1}^{\infty} \frac{\sin\left(\frac{m\pi x}{L_c}\right) \sin\left(\frac{l\pi y}{L_k}\right)}{r_{ml}} \left[ \int_0^{L_k} \int_0^{L_c} \sin\left(\frac{m\pi \xi}{L_c}\right) \sin\left(\frac{l\pi \eta}{L_k}\right) n_u(\xi, \eta) d\xi d\eta + \right. \quad (3.90)$$

$$+ \frac{lL_c}{mL_k} \left( (-1)^m - 1 \right) \left( g_1 + g_3 (-1)^l \right) + \frac{mL_c}{lL_k} \left( (-1)^l - 1 \right) \left( g_2 + g_4 (-1)^m \right) \Big],$$

$$\text{где } r_{ml} = \left( \frac{m\pi}{L_c} \right)^2 + \left( \frac{l\pi}{L_k} \right)^2 + a.$$

На рис. 3.10 приведены полутонные изображения реализаций эллиптической модели размером  $50 \times 50$  элементов, полученные в результате моделирования на ЭВМ выражения (3.90) для различных значений параметра  $a$  [45].

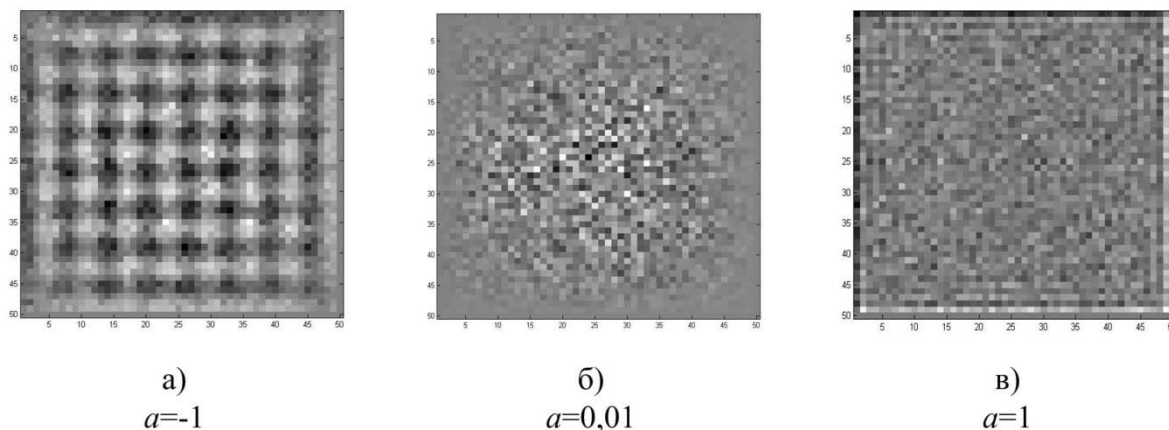


Рис. 3.10. Примеры реализаций эллиптической модели изображения

Определяя по аналогии с предыдущими случаями корреляционную функцию эллиптической модели, получим

$$R_u(y_1, x_1, y_2, x_2) = \frac{N_u}{L_c^2 L_k^2} \sum_{m=1}^{\infty} \sum_{l=1}^{\infty} \frac{\sin\left(\frac{m\pi x_1}{L_c}\right) \sin\left(\frac{m\pi x_2}{L_c}\right)}{r_{ml}} \frac{\sin\left(\frac{l\pi y_1}{L_k}\right) \sin\left(\frac{l\pi y_2}{L_k}\right)}{r_{ml}}. \quad (3.91)$$

Как видно из (3.91), эллиптические модели являются некаузальными по обеим координатам. Для полей, описываемых такими моделями, характерным является то, что наиболее коррелированными являются те значения поля, которые расположены ближе к центру плоскости  $xu$ .

Понятие каузальности изображения тесно связано с его типом. Так, для каузальных изображений наличие причинно-следственных связей во всех направлениях обеспечивает однородность всего изображения, т. е. отсутствие особенностей в каком-либо направлении. К таким изображениям можно отнести, например, радиолокационные изображения рельефа земной поверхности, мелкомасштабные панорамные аэрокосмические снимки различных участков Земли и других планет, различного рода фоновые изображения и др. Отсутствие каузальности приводит к возникновению неоднородностей (особенностей) в изображении на каких-либо участках. Изображениями некаузального типа будут являться, например, одиночные и групповые портреты, изображения крупномасштабных объектов на различных фонах и т. п.

Таким образом, установлено, что с помощью трёх типов моделей можно описать различные по характеру изображения, а также установлена связь между типом изображения и статистическими характеристиками модели.

Рассмотренные выше модели относятся к классу неподвижных изображений. В качестве модели изменяющихся во времени изображений предлагается использовать хорошо изученную краевую задачу математической физики на основе параболического дифференциального уравнения в частных производных

$$\frac{\partial u(t, y, x)}{\partial t} = \alpha \left[ \frac{\partial^2 u(t, y, x)}{\partial y^2} + \frac{\partial^2 u(t, y, x)}{\partial x^2} \right] - \beta u(t, y, x) + n_u(t, y, x), \quad (3.92)$$

$$u(0, y, x) = 0; \quad u(t, 0, x) = g_1, \quad u(t, y, 0) = g_2, \quad u(t, L_k, x) = g_3, \quad u(t, y, L_c) = g_4,$$

где  $\alpha, \beta > 0, t \geq 0, 0 \leq x \leq L_c, 0 \leq y \leq L_k$ .

Здесь  $L_c$  и  $L_k$  – размеры изображения по строке и по кадру соответственно;  $n_u(t, y, x)$  – белое гауссовское шумовое поле с нулевым математическим ожиданием и корреляционной функцией

$$M \{n(t_1, y_1, x_1) n(t_2, y_2, x_2)\} = \frac{N_u}{2} \delta(t_2 - t_1) \delta(y_2 - y_1) \delta(x_2 - x_1). \quad (3.93)$$

Воспользовавшись справочными сведениями из [10], определим решение краевой задачи (3.92), которое в окончательном виде будет определяться выражением

$$\begin{aligned} u(t, y, x) = & \frac{4}{L_c L_k} \sum_{m=1}^{\infty} \sum_{l=1}^{\infty} \sin \frac{m\pi x}{L_c} \sin \frac{l\pi y}{L_k} \exp(-r_{ml} t) \times \\ & \times \left[ \int_0^t \exp(r_{ml} \tau) \int_0^{L_k} \sin \frac{l\pi \eta}{L_k} \int_0^{L_c} \sin \frac{m\pi \varsigma}{L_c} n_u(\tau, \eta, \varsigma) d\varsigma d\eta d\tau - \alpha \frac{[1 - \exp(r_{ml} t)]}{r_{ml}} \times \right. \\ & \left. \times \left[ (g_2 - (-1)^m g_4) (1 - (-1)^l) \frac{m}{l} \frac{L_k}{L_c} + (g_1 - (-1)^l g_3) (1 - (-1)^m) \frac{l}{m} \frac{L_c}{L_k} \right] \right], \end{aligned}$$

$$\text{где } r_{ml} = -\beta - \alpha \pi^2 \left( \frac{m^2}{L_c^2} + \frac{l^2}{L_k^2} \right). \quad (3.94)$$

Корреляционная функция для данной модели при нулевых граничных условиях будет иметь вид

$$\begin{aligned} R_u(t_1, t_2, y_1, y_2, x_1, x_2) = & \frac{N_u}{2 L_c L_k} \sum_{m=1}^{\infty} \sum_{l=1}^{\infty} \sin \frac{m\pi x_1}{L_c} \sin \frac{m\pi x_2}{L_c} \times \\ & \times \sin \frac{l\pi y_1}{L_k} \sin \frac{l\pi y_2}{L_k} \frac{\exp[-|t_2 - t_1| r_{ml}] - \exp[-|t_2 + t_1| r_{ml}]}{r_{ml}}. \end{aligned} \quad (3.95)$$

Из (3.95) следует, что модель (3.92) является каузальной по времени и некаузальной по пространственным координатам, что вполне соответствует физической природе многих реальных изображений.

## 4. Имитационное моделирование систем

### 4.1. Основы имитационного моделирования

#### 4.1.1. Основные понятия имитационного моделирования

**Имитационное моделирование** – это метод исследования, при котором изучаемая система заменяется **моделью**, с достаточной точностью описывающей **реальную систему**, с которой проводятся эксперименты с целью получения информации об этой системе.

Экспериментирование с моделью называют **имитацией** (имитация – это постижение сути явления без эксперимента на реальном объекте).

**Имитационное моделирование** – это частный случай математического моделирования. Существует класс объектов, для которых по различным причинам не разработаны **аналитические модели**, либо не разработаны методы решения полученной модели. В этом случае аналитическая модель заменяется **имитатором** или **имитационной моделью**.

Физическая система заменяется имитатором, в качестве которого могут выступать:

- специальное устройство;
- программа.

**Имитационная модель** – логико-математическое описание объекта, которое может быть использовано для экспериментирования на компьютере в целях проектирования, анализа и оценки функционирования объекта.

Имитационная модель позволяет воспроизвести любой марковский процесс.

Существует три основных подхода к имитационному моделированию (рис. 4.1).

**1. Системная динамика.** Парадигма моделирования, где для исследуемой системы строятся графические диаграммы причинных связей и глобальных влияний одних параметров на другие во времени, а затем созданная на основе этих диаграмм модель имитируется на компьютере. Метод предложен Джейм Форрестером в 1950-х годах [51, 52].

**2. Дискретно-событийное моделирование.** Подход к моделированию, предлагающий абстрагироваться от непрерывной природы событий и рассматривать только основные события моделируемой системы, такие как «ожидание», «обработка заказа», «движение с грузом», «разгрузка». Предложен Джеффри Гордоном в 1960-х годах [5, 9, 67].

**3. Агентное моделирование.** Относительно новое (1990–2000-е годы) направление в имитационном моделировании, которое используется для исследования децентрализованных систем, динамика функционирования которых определяется не глобальными правилами, а наоборот, когда эти глобальные правила и законы являются результатом индивидуальной активности членов группы.



Рис. 4.1. Основные подходы к имитационному моделированию

Имитационное моделирование может проводиться на различных уровнях абстракции (рис. 4.2):

- стратегический уровень (макроуровень, меньше деталей);
- тактический уровень (мезоуровень, средняя детальность);
- оперативный уровень (микроуровень, больше деталей).



Рис. 4.2. Уровни имитационного моделирования

Рассмотрим детальнее второй подход. В дискретно-событийном моделировании функционирование системы представляется как хронологическая последовательность событий. Событие происходит в определённый момент времени и знаменует собой изменение состояния системы.

**Системы дискретно-событийного моделирования** – это, чаще всего, проблемно-ориентированные языки программирования или библиотеки для высокоуровневых языков.

**Требования к имитационной модели:**

- полнота;
- гибкость;
- минимальная длительность разработки;
- блочная структура;
- эффективность воспроизведения на ЭВМ.

Реализация процесса работы системы моделируется на ЭВМ с помощью последовательностей случайных и псевдослучайных чисел.

После завершения накопления статистики производится усреднение результатов моделирования, как правило, по времени или числу реализаций процесса.

При моделировании необходимо учитывать переходные процессы. Длительность переходного процесса составляет не менее 300–500 событий. Для получения установившихся характеристик модель нужно разогнать.

Рекомендуемая степень детализации описания событий определяется допустимым диапазоном отношения средних величин случайных интервалов времени. Обычно соотношение не превышает 100:1. В противном случае нужно строить иерархическую систему, используя в моделях верхнего уровня усредненные параметры нижнего уровня в виде констант.

**Достоинства имитационного моделирования [59]:**

- свобода от каких-либо ограничений на класс решаемых задач;
- наглядность и простота в освоении;
- повышенная устойчивость к сбоям ЭВМ;
- лёгкость интерпретации и контроля промежуточных результатов.

**Недостатки имитационного моделирования [59]:**

- большой расход машинного времени;
- малая точность вероятностных характеристик редких событий;
- трудность получения обобщающих выводов и рекомендаций;
- сложность оптимизации системы;
- вероятностная оценка погрешностей.

#### **4.1.2. Организация и принципы разработки имитационных моделей**

Процесс имитационного моделирования обычно включает следующие этапы, которые могут перекрываться по времени [59]:

- 1) организация разработки модели;
- 2) создание концептуальной модели;
- 3) определение системы;



- 4) формализация описания системы;
- 5) подготовка исходных данных;
- 6) программирование и трансляция модели;
- 7) оценка адекватности модели;
- 8) планирование эксперимента;
- 9) планирование прогонов;
- 10) машинный эксперимент;
- 11) анализ результатов моделирования;
- 12) интерпретация результатов;
- 13) реализация результатов моделирования;
- 14) документирование процесса разработки модели и хода моделирования.

**Организация разработки модели** предполагает установление связей с заказчиком, определение задач и ресурсов, установление взаимодействия, отчётности и контроля, планирование работы.

**Концептуальное моделирование** позволяет сделать предварительное описание системы, установить основные источники информации.

**Определение системы включает:**

- уточнение границ с внешней средой;
- определение характеристик среды и внешних воздействий;
- изучение состава, назначения, внешних и внутренних связей системы;
- выявление ограничений и выбор показателей эффективности;
- постановка задачи на исследование.

**Формализация описания** состоит в построении математической модели системы (структуры и зависимостей между элементами).

**Подготовка исходных данных** состоит в сборе и обработке результатов наблюдений за моделируемой системой.

**Программирование и трансляция модели** предполагает запись модели на языке программирования.

**Оценка адекватности** предполагает проверку:

- полноты учёта основных факторов и ограничений;
- соответствия исходных данных модели реальным;
- наличия в модели необходимых данных;
- корректности моделирующей программы;
- правильности получения конечных результатов из исходных данных;
- осмысленности результатов при нормальных условиях.

**Планирование экспериментов** предполагает определение вариантов исследования и стратегию их перебора.

**Планирование прогонов** необходимо для получения наилучших статистических оценок показателей эффективности.

**Машинный эксперимент** и **анализ результатов** составляет ядро процесса имитационного моделирования.

**Интерпретация результатов** предполагает их перенос на исследуемую систему.

**Реализация результатов** предполагает практическое использование полученных в ходе моделирования рекомендаций и оценку их эффективности

**Документирование** процесса разработки модели и хода моделирования подтверждает успешное завершение проекта и обеспечивает накопление опыта.

Для построения имитационной модели необходимо образовать взаимодействие элементов [59]:

- состояний;
- событий;
- генераторов случайных чисел;
- счётчиков;
- таймера;
- цепей событий;
- цели моделирования;
- блока инициализации;
- условий завершения моделирования;
- методов обработки результатов моделирования.

**Состояние модели** – множество устойчивых значений переменных параметров модели, которое определяется с необходимой и достаточной степенью детальности (например, текущее число заявок в системе, моменты наступления ближайших событий и т. д.) и меняется скачкообразно при наступлении событий.

**События** – это скачкообразные изменения состояния модели, которые могут быть первичными (приход в систему и уход из неё) или вторичными (приём заявки на обслуживание, продвижение в очереди).

**Генераторы случайных чисел** (ГСЧ) генерируют следующие состояния (моменты наступления следующих первичных событий).

**Счётчики** используются для накопления результатов имитационного моделирования (например, суммарное время ожидания, количество обслуженных заявок и т. д.).

**Таймер** – это счётчик модельного (системного) времени. Состояние системы фиксируется в моменты событий определённого рода, другие параметры воспроизводятся за случайные интервалы между событиями.

**Цепи событий** реализуют логику модели в процессе обработки. Выделяют цепи текущих, будущих и задержанных событий. Цепь текущих событий состоит из событий, наступающих в один и тот же момент модельного времени (уход обслуженной заявки, поступление новой заявки на обслуживание, про-

движение очереди). Цепь будущих событий содержит запланированные на последующие моменты времени события с помощью ГСЧ. В цепи задержанных событий находятся события, заблокированные сложившимися на конкретный момент модельного времени условиями моделирования.

*Блок инициализации* устанавливает в исходное состояние элементы модели (счётчики, генераторы псевдослучайных чисел).

*Цель моделирования* определяется показателями качества функционирования системы, которые необходимо получить в результате моделирования.

*Условия завершения моделирования* определяются:

- по *текущему значению счётчика*, связанного с целью моделирования (по заданному значению таймера или заданному числу обслуженных заявок);
- по *комбинации условий*;
- при *достижении заданной точности* одного из показателей.

*Методы обработки результатов моделирования* предполагают получение статистических оценок для искомых показателей (математических ожиданий, дисперсий и др.) и построения на их основе гистограмм, статистических функций распределения.

#### **4.1.3. Алгоритм имитационного моделирования**

Рассмотрим модель системы  $G/G/1/R$  – одноканальной системы с ограниченной ёмкостью накопителя, с произвольным распределением времени между входящими заявками и с произвольным временем обслуживания.

Базовый алгоритм имитационного моделирования предполагает наличие следующих элементов:

- 1) инициализация;
- 2) выбор момента очередного события;
- 3) продвижение таймера;
- 4) накопление счётчиков;
- 5) выбор варианта обработки информации;
- 6) выполнение варианта обработки;
- 7) определение конца протока;
- 8) повторение этапов моделирования, начиная с выбора момента очередного события;
- 9) обработка результатов и вывод их на печать.

Структурная схема имитационного моделирования общего типа  $GI/G/1/R$  представлена на рис. 4.3.

После приведения системы в исходное состояние (очистки счётчиков) формируется с помощью генератора случайных чисел (ГСЧ) момент прибытия очередной заявки. Он должно быть заведомо меньше момента освобождения обслуживающего канала.

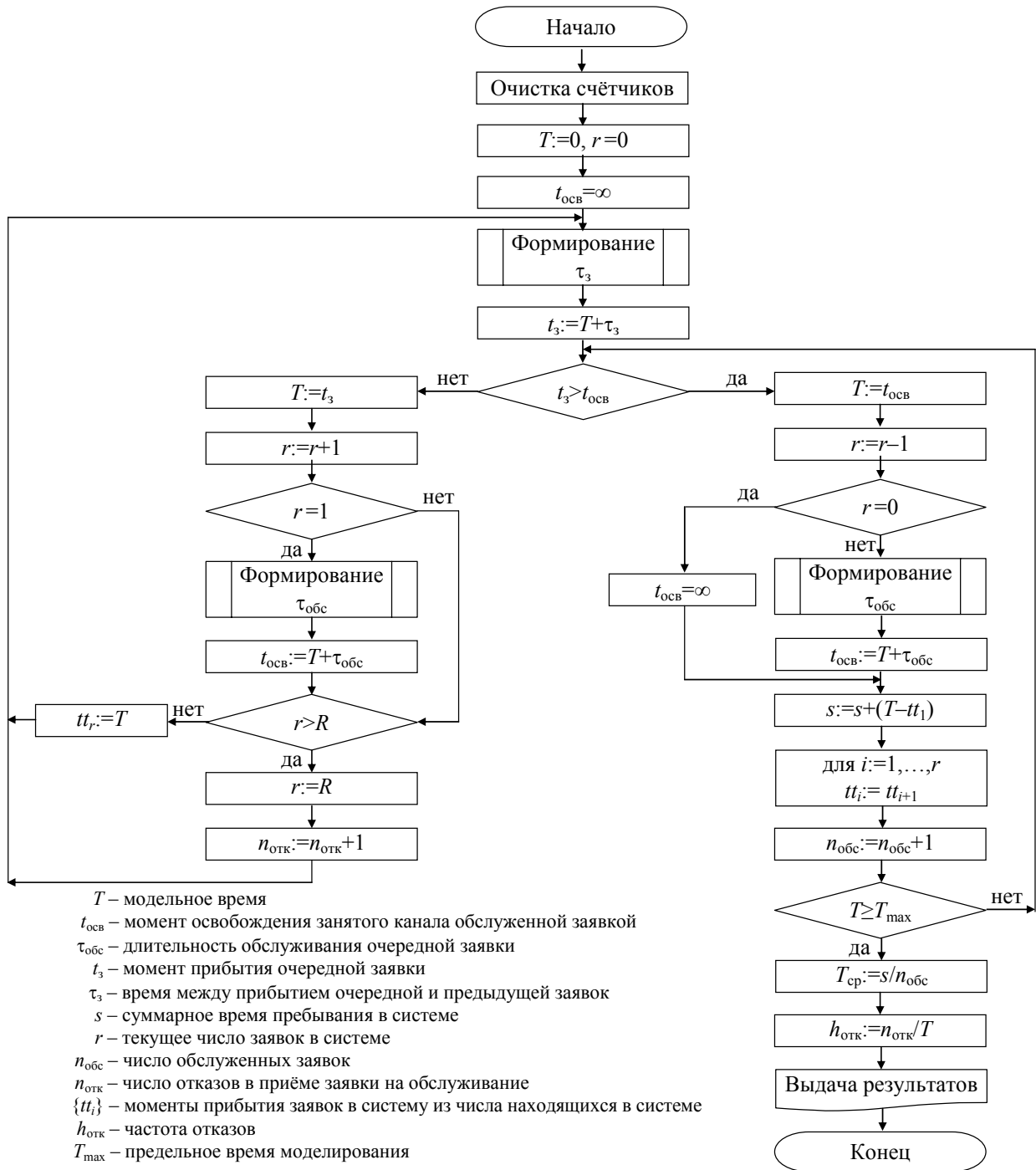


Рис. 4.3. Структурная схема алгоритма имитационного моделирования

Число находящихся в системе заявок  $r$  увеличивается на 1. Если  $r = 1$ , то заявка поступает в свободную систему. Она немедленно принимается на обслуживание, а для канала формируется  $t_{\text{осв}}$ . Если новое число заявок превышает  $R$ , то вновь прибывшая заявка получает отказ. В противном случае в массиве  $\tau$  запоминается время её прибытия в систему. После этого формируется момент прибытия очередной заявки. Если очередным событием в системе оказывается освобождение канала, то число заявок в системе уменьшается на 1. Если  $r = 0$ , то момент освобождения свободного канала полагается равным  $\infty$ . В противном случае из очереди выбирается очередная заявка и формируется новое обращение к ГСЧ в новый момент освобождения.

К счётчику  $s$  добавляется время, проведённое обслуженной заявкой в системе, и выполняется проверка на окончание моделирования по достижению заданного модельного времени  $T_{\max}$ . В случае завершения  $T$  подсчитываются среднее время пребывания в системе  $T_{\text{ср}}$  и частота отказов  $h_{\text{отк}}$ . Результат выводится на печать (экран, принтер, в файл).

Временная диаграмма основных событий в процессе имитационного моделирования представлена на рис. 4.4.

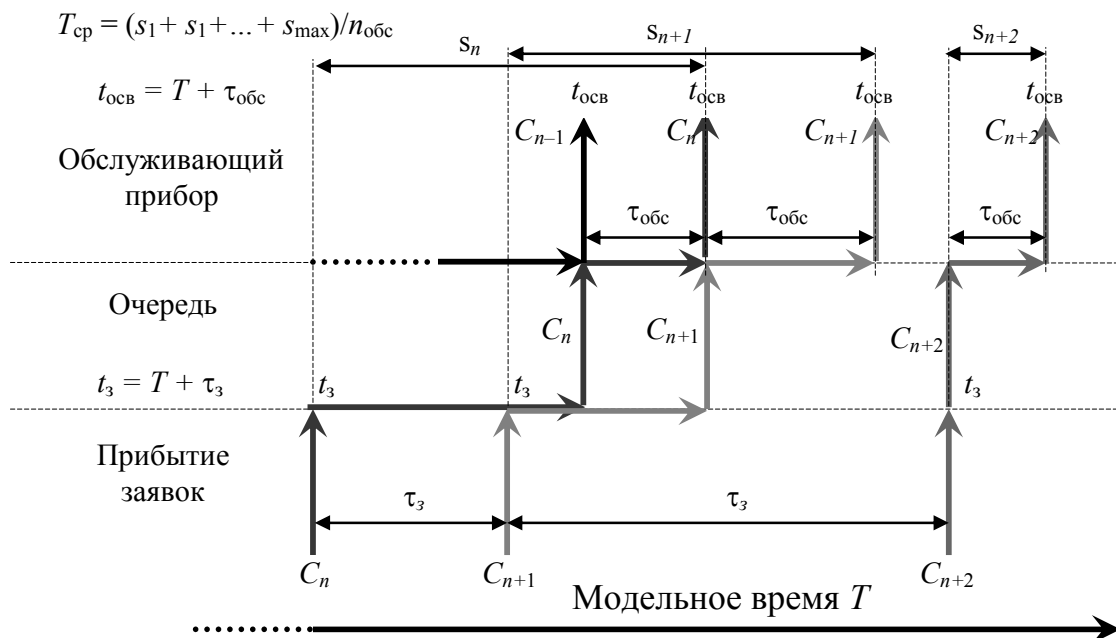


Рис. 4.4. Временная диаграмма имитационного моделирования

В качестве условия завершения моделирования может служить не только окончание модельного времени, но и прохождение через систему заданного число заявок, а также достижение заданной точности получаемого среднего времени пребывания заявки в системе.

При моделировании *многоканальных систем*  $t_{\text{осв}}$  фиксируется для каждого канала отдельно, а с моментом  $t_з$  сравнивается наиболее ранний из них.

При моделировании *приоритетного обслуживания* организуется отдельная очередь заявок каждого приоритета. В случае абсолютного приоритета тип прибывшей заявки должен сравниваться с приоритетом обслуживаемой для необходимости прерывания. Для прерванной заявки, если она помещается в очередь, время её обслуживания может быть сформировано как остаток времени обслуживания на момент прерывания.

При моделировании системы с *временными ограничениями* на пребывание в ней (с «нетерпеливыми заявками») каждую заявку с момента её прихода в систему должен сопровождать момент исчерпания её «терпения». В системе должен фиксироваться наиболее ранний из таких моментов, который наряду с прибытием первой заявки и освобождением первого канала управляет логикой модели. Логика работы модели может меняться от цели моделирования, которая диктует набор вспомогательных счётчиков и манипуляции с их содержимым.

#### 4.1.4. Генерация случайных чисел

Генерация случайных чисел – это воспроизведение на ЭВМ вероятностного процесса, статистические характеристики которого совпадают с характеристиками исходного моделируемого процесса. Для воспроизведения этого процесса используются случайные или псевдослучайные числа.

Программы, формирующие такие случайные последовательности, называют датчиками или *генераторами случайных чисел* (ГСЧ). В саму имитационную модель можно встраивать либо непосредственно первичные данные, либо данные, сформированные с помощью генератора случайных чисел.

Получение случайных чисел с требуемым законом распределения выполняется в два этапа:

- формирование программным или физическим методом числа  $U_i$  равномерно распределённого на интервале  $[0,1]$ ;
- программный переход от  $U_i$  к случайному числу  $X_i$ , которое имеет требуемое распределение  $F(x)$ .

Равномерно распределённое двоичное число в интервале  $[0,1]$  представляется в ЭВМ в двоичной форме в виде  $n$ -разрядной последовательности 0 и 1. Если в любом разряде  $n$ -разрядного двоичного кода с равной вероятностью появляются нули и единицы, то такое число будет равномерно распределено.

**Физические датчики** состоят из  $n$  триггеров со счётным входом, регистрирующих независимые потоки импульсов от счётчика радиоактивных частиц или шумовые выбросы электронной лампы. Такие потоки считаются простейшими. В момент снятия отсчёта триггер может быть с одинаковой вероятностью в 0 или 1. Достоинства – истинная случайность (из-за отсутствия корреляции между смежными разрядами) и отсутствие затрат процессорного времени на генерацию.

**Программные датчики** генерируют *псевдослучайные* числа. Они обеспечивают близкое к равномерному перемешивание разрядов исходных чисел некоторым закономерным способом. *Достоинства*: 1) не нужно дополнительное оборудование; 2) возможность повторить прогон последовательности случайных чисел; 3) однократная проверка ГСЧ после разработки.

Так, как в  $n$ -разрядной сетке количество различных двоичных чисел равно  $2^n$ , отрезки последовательностей различных случайных величин рано или поздно начинают повторяться. Поэтому использовать в одной задаче количество чисел больше длины периода не рекомендуется.

Первым ГСЧ явился метод «*середины произведения*», в котором в качестве очередного числа берётся середина произведения двух  $n$ -разрядных чисел, а вторым сомножителем служат оставшиеся разряды (рис. 4.5).

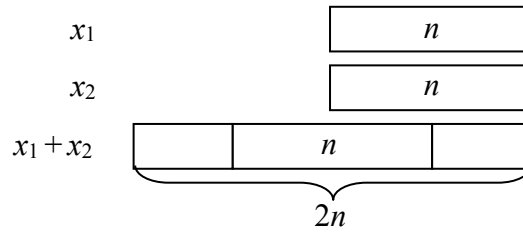


Рис. 4.5. Метод «середины произведения»

Однако данный способ даёт не очень хорошие результаты, поэтому получил развитие другой способ – «линейный конгруэнтный генератор», когда псевдослучайное число вычисляется по формуле

$$x_i = (ax_{i-1} + c)_{\text{mod } M}.$$

Универсальным способом перехода к требуемому распределению является метод обратной функции (рис. 4.6):

$$X = F^{-1}(U).$$

Показательное распределение имеет ФР:

$$F(X) = 1 - e^{-\lambda x}.$$

Тогда  $X = -(1/\lambda) \ln(1-U)$ . Поскольку  $(1-U)$  распределено так же, как и  $U$ , получим:

$$X = -(1/\lambda) \ln U.$$

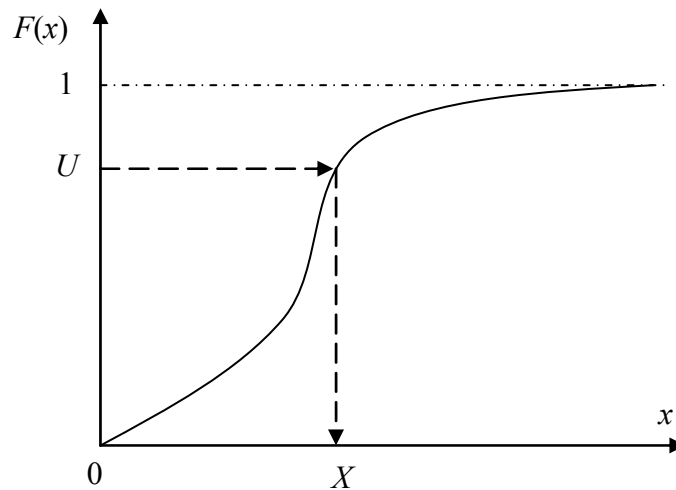


Рис. 4.6. Метод обратной функции

Функция распределения Релея  $F(x) = 1 - e^{-x^2/2\sigma^2}$  приводит к генератору вида

$$X = \sigma \sqrt{-2 \ln U}.$$

Функция распределения Вейбулла  $F(x) = 1 - e^{-x^k/T}$  приводит к генератору вида

$$X = \sqrt[k]{-T \ln(1-U)}.$$

Однако не всегда можно найти обратную функцию. Поэтому часто используются аппроксимации обратной ФР.

Для стандартного нормального распределения

$$\Phi(X) = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{-y^2/2} dy;$$

$$x = \frac{2,30753 + 0,27061U}{1 + 0,99229U + 0,04481U^2} \text{ с погрешностью менее } 3 \cdot 10^{-3};$$

$$x = -\frac{2,515517 + 0,802853U + 0,0103258U^2}{1 + 1,432788U + 0,18363U^2 + 0,001308U^3} \text{ с погрешностью менее } 4,5 \cdot 10^{-4};$$

$$t = \sqrt{-2 \ln U'}, \quad U' = \max\{U, 1 - U\}.$$

*Этапы генерации нормально распределённого числа:*

- 1)  $U \in [0,1]$ ;
- 2) если  $U < 0,5$ ,  $U' = 1 - U$ , иначе  $U' = U$ ;
- 3)  $U = \sqrt{-2 \ln U'}$ ;
- 4) вычисление  $X$  по (\*) или (\*\*);
- 5) если  $U < 0,5$ ,  $\tilde{x} = -x'$ , иначе:  $\tilde{x} = x'$ ;
- 6)  $X = x + G \tilde{x}$ .

В отдельных случаях удобно воспроизводить вероятностную схему распределения:

$$X = \sum_{k=1}^k \left(-\frac{1}{\lambda} \ln U_k\right) = -\frac{1}{\lambda} \sum_{k=1}^r \ln U_k = \frac{1}{\lambda} \ln \prod_{k=1}^r U_k \text{ – распределение Эрланга.}$$

Гиперэкспоненциальное распределение формируется с использованием двух последовательных обращений к ГСЧ:

- 1)  $U_1 \rightarrow 1 \vee 2$ ;
- 2)  $U_2 \rightarrow \exp$ .

Для нахождения нормальных законов часто используется равномерный закон.

При  $m > 6 \rightarrow$  к нормальному закону распределения:

$$W = \sqrt{\frac{12}{n}} \left( \sum_{i=1}^n U_1 - n/2 \right), \quad d = 1/12, \quad z = 1/2.$$

## 4.2. Подготовка и проведение имитационного моделирования

### 4.2.1. Испытания и исследование свойств имитационной модели

После того как имитационная модель реализована на ЭВМ, исследователь должен выполнить последовательность технологических этапов:

- испытания имитационной модели;
- исследование свойств модели;
- планирование экспериментов на ЭВМ;
- эксплуатация модели;
- документирование результатов моделирования.

**Испытания ИМ** включают:

- задание исходной информации моделирования;



- *верификацию модели*, которая состоит в проверке соответствия ИМ тому замыслу, который был заложен при её разработке;
- *проверку адекватности* ИМ объекту моделирования;
- *калибровку модели* для обеспечения адекватности ИМ реальному объекту.

**Исследование свойств ИМ** предполагает последовательное выполнение следующих действий:

- *оценку погрешностей* имитации из-за неидеальности генераторов псевдослучайных чисел, используемых в ИМ;
- *определение времени моделирования*, необходимого для достижения заданной точности моделирования;
- *оценку устойчивости* результатов моделирования при различных изменениях параметров и заданных значениях переменных;
- *оценку чувствительности* результатов моделирования к изменению параметров модели и исходных данных моделирования.

**Планирование экспериментов.** Поскольку «прогоны» вариантов модели обычно дороги, нужно стремиться к тому, чтобы количество параметров моделирования (размерность  $X$ ) и число уровней каждого параметра были минимальными и соответствовали целям эксперимента. Важными являются следующие вопросы:

- определение требуемого размера выборки статистик моделирования, используемых при вычислении значений  $Y$ ;
- определение интервалов изменения параметров моделирования  $X$ ;
- планирование поиска ошибок в имитационном эксперименте.

**Эксплуатация модели.** Есть три основных приёма, позволяющих получить наиболее удачное представление результатов модельного эксперимента и существенно облегчить решение задачи моделирования:

- *исключение резко отклоняющихся значений* при натурных экспериментах с прототипами;
- *выбор системы координат*, в которой полный набор данных моделирования давал бы по возможности прямую линию;
- *анализ регрессионной зависимости* функции отклика модели от параметров моделирования.

**Документирование результатов моделирования.** Тщательное и полное документирование результатов имитационного эксперимента в целях дальнейшего использования этих результатов может значительно увеличить срок жизни модели и вероятность успешной её реализации. При подготовке документации необходимо решить, какие виды документов необходимы, кто их пишет и каким образом, как меняется качество документов по мере использования модели.

Рассмотрим подробнее этапы *испытаний и исследования свойств* имитационных моделей.

### **Испытание имитационной модели**

#### ***Задание исходной информации для моделирования***

Процедура задания исходной информации для моделирования зависит от типа модели (модель функционирующей или проектируемой системы).

При моделировании ***функционирующих систем*** зачастую организуется измерение характеристик поведения системы  $Y^*_Q$ , рабочей нагрузки системы  $G^*$  и параметров функционирования системы  $Q^*$ .

При моделировании ***проектируемых систем***, а также при моделировании функционирующих систем, **когда нельзя организовать измерения**, для формирования исходной информации проводят **измерения на прототипах систем**. Если таких прототипов не существует, то исследователю при построении модели приходится использовать экспертные оценки внешней нагрузки  $G^*$  параметров функционирования  $Q^*$  и характеристик поведения системы  $Y^*_Q$ .

В ходе натурального эксперимента по результатам измерений параметров рабочей нагрузки  $G^*$  формируется *модель рабочей нагрузки  $G$* .

Когда нет возможности определить значения компонент вектора  $G^*$  на основании измерений, приходится полагаться на *субъективные (экспертные) оценки (например, метод Дельфы)*.

### **Верификация имитационной модели**

*Верификация имитационной модели* состоит в доказательстве утверждений соответствия алгоритма её функционирования замыслу моделирования путём формальных и неформальных исследований реализованной программы модели.

*Неформальные исследования* программы модели имеют целью убедить исследователя и заказчика ИМ в соответствии построенной модели замыслу, заложенному конструктором при её разработке. Они представляют ряд процедур проверки правильности алгоритма функционирования модели объекта исследования. Проведение этих процедур входит в состав комплексной отладки ИМ.

Формальные методы верификации программы модели включают в себя:

- использование специальных препроцессоров и постпроцессоров в качестве «читателей» программ;
- замену некоторых стохастических элементов модели детерминированными и проверку на «ожидаемость» результатов моделирования;
- использование теста на «непрерывность» моделирования.

### **Проверка адекватности имитационной модели**

Оценка адекватности модели объекту исследования проводится в случае, когда можно определить значение отклика системы в ходе натуральных испытаний.

Пусть известен отклик реальной системы  $Y^*_Q$  при нагрузке  $G^*$ , параметрах  $Q^*$  и неизвестной из-за сложности и малой изученности протекающих в объекте процессов функции  $Y^*_Q = \varphi^*(Q^*, G^*)$ . Модель представляет собой аппроксимирующую зависимость  $Y_k = \varphi(Q_k, G)$ ,  $k = 1, N$  ( $k$  и  $N$  – соответственно номер и число опытов на модели и реальной системе), найденную в ходе эксплуатации объекта по результатам наблюдений входных воздействий  $Q^*_k$ ,  $k = 1, N$ , и выхода  $Y^*_{Q_k}$  при заданных значениях  $G^*$ .

Наиболее употребительные способы проверки адекватности:

- 1) по средним значениям откликов модели и системы;
- 2) по дисперсиям отклонений откликов модели от среднего значения откликов системы;
- 3) по максимальному значению абсолютных отклонений откликов модели от откликов системы.

### ***Калибровка имитационной модели***

Когда модель оказывается **неадекватной** реальной системе, исследователь переходит к **калибровке ИМ**. **Цель калибровки** состоит в *уменьшении неточностей* формулировки ИМ, обусловленных ошибочной или недостаточно подробной моделью  $\varphi(X, G)$ .

Исследователь может корректировать либо модель внешней нагрузки  $G$ , либо алгоритм функционирования модели  $\varphi(X, G)$ .

В ходе калибровки модели системы можно проводить изменения трёх типов:

- глобальные структурные изменения (например, добавление программ процессов, изменения событий в процессах);
- локальные изменения в модели (например, замена компонент модели внешне эквивалентными, но более точными);
- изменение некоторых параметров, используемых в качестве калибровочных.

*При глобальной модификации* модели используется следующая **методика** из трёх шагов, каждый из которых является по своей природе **итеративным**.

**Шаг 1. Сравнение распределений.** Определяется погрешность моделирования, которая должна быть минимизирована в процессе калибровки.

**Шаг. 2. Балансировка модели.** На каждой итерации процедуры балансировки модели исследователь меняет значения вектора параметров  $Q$  аналогично натурному эксперименту и фиксирует выборки объёма  $N$  откликов модели системы  $\{Y_{nk}\}$ ,  $k = 1, N$ .

**Шаг. 3. Оптимизация модели.** Необязательно стремиться к тому, чтобы по всем параметрам регрессия  $E_{3n}$  была незначима. Достаточно достичь незначимости регрессии  $E_{3n}$  только для наиболее существенных параметров.

## Исследование свойств имитационной модели

Наиболее существенны следующие процедуры исследования модели: *оценка погрешности имитации*, обусловленная наличием в ИМ неидеальных генераторов псевдослучайных чисел; *определение длительности* переходного режима в работе ИМ; *оценка устойчивости* результатов имитации исследуемых процессов; *исследование чувствительности* ИМ, т. е. зависимости изменения  $Y$  от изменения  $X$ .

Рассмотрим порядок действий исследователя при выполнении каждой из перечисленных процедур.

### ***Оценка погрешности имитации из-за генераторов случайных чисел***

В любой вероятностной ИМ сложной системы применяются генераторы псевдослучайных чисел. Каждый из них использует базовый генератор, числа с выхода которого затем преобразуются для получения псевдослучайных величин с заданными законами распределения.

Но такой генератор является источником погрешности имитации. Для проверки качества генераторов псевдослучайных чисел имеется множество тестов.

Существует проблема компромисса между *точностью воспроизведения* псевдослучайных величин согласно выбранному закону их распределения и *сложностью реализации* таких генераторов. Поэтому широко распространённые ГСЧ, использующие *простые алгоритмы* имитации псевдослучайных чисел, в принципе не могут быть хорошего качества. Но применение ГСЧ со *сложными алгоритмами* имитации зачастую не представляется возможным из-за ограничений ресурса времени ЭВМ на моделирование вариантов сложной системы.

Поэтому необходимо оценить эту погрешность. В ряде случаев после определения погрешности, связанной с использованием генераторов псевдослучайных чисел, может оказаться, что точность имитации приемлема и проблема отсутствует.

Как показала практика исследования вычислительных систем на ИМ, зачастую достаточно определить оценки *математического ожидания* и *дисперсий отклонений* компонент вектора откликов  $Y$ .

### ***Определение длительности переходного режима***

Одно из основных отличий имитационных экспериментов от натурных испытаний объекта моделирования заключается в простоте повторения и воспроизведения условий эксперимента, а также в простоте запуска, прерывания и возобновления эксперимента. Это позволяет экспериментатору полностью контролировать модельный эксперимент.

Однако необходимо заботиться о том, как задать начальные условия и когда приступить к сбору данных.

Существуют три способа уменьшения влияния начального периода на динамику моделирования сложной системы:

- 1) использование длинных прогонов модели;
- 2) исключение из рассмотрения начального периода прогона;
- 3) выбор такого начального условия, которое ближе всего к типичному, и тем самым существенное уменьшение длительности переходного режима.

Первый подход можно применять только в случае, если прогон модели не требует много машинного времени.

При втором подходе часть машинного времени тратится бесполезно и из-за сокращения объёма выборки увеличивается дисперсия отклика модели.

### ***Оценка устойчивости результатов имитации***

Под устойчивостью результатов имитации будем понимать степень нечувствительности её к изменению входных условий. Универсальной процедуры подобной проверки не существует. Поэтому создатели моделей вынуждены прибегать к методам «для данного случая», частичным тестам и к здравому смыслу.

В общем случае можно следовать принципу: *чем ближе структура модели к структуре системы и чем выше степень её детальности, тем обширнее область пригодности модели*. Однако иногда в процессе структурной калибровки модели достигается точка, за которой *устойчивость* результатов моделирования может ухудшаться из-за чрезмерной подстройки модели к частным входным условиям, использованным для её калибровки.

Практика моделирования вычислительных систем показала, что устойчивость результатов моделирования можно оценивать *дисперсией значений отклика* (по выбранной компоненте). Если эта дисперсия при увеличении времени моделирования  $T_n$  не увеличивается, значит, результаты моделирования *устойчивы*.

### ***Исследование чувствительности имитационной модели***

Исследователь должен установить *диапазон изменения отклика* модели  $Y$  при изменении каждой компоненты вектора параметров  $X$ . В зависимости от диапазона изменения откликов  $Y$  определяется стратегия планирования экспериментов на ИМ. Если при значительной амплитуде изменений некоторой компоненты вектора параметров модели  $X$  отклик  $Y$  меняется незначительно, то это означает, что точность представления этой компоненты в ИМ не играет существенной роли. Если же отклик модели  $Y$  окажется высокочувствительным к изменению некоторой компоненты вектора  $X$ , то это служит прямым указанием на необходимость представления её в модели с максимально возможной точностью.

Вторым, не менее важным моментом исследования чувствительности модели является проверка зависимости отклика модели  $Y$  от изменений параметров внешней среды  $G$ . Меняя характеристики  $G$  в обе стороны на некоторую величину  $\Delta G$ , исследователь оценивает диапазон изменений вектора отклика модели  $\delta Y$ .

#### 4.2.2. Планирование имитационных экспериментов

**Машинный эксперимент с моделью системы  $S$  при её исследовании и проектировании** проводится с целью получения информации о характеристиках процесса функционирования рассматриваемого объекта.

Эта информация может быть получена как для анализа характеристик, так и для их оптимизации при заданных ограничениях, т. е. для синтеза структуры, алгоритмов и параметров системы  $S$ .

**Основная задача планирования машинных экспериментов** – получение необходимой информации об исследуемой системе  $S$  при ограничениях на ресурсы (затраты машинного времени, памяти и т. п.).

К числу **частных задач**, решаемых при планировании машинных экспериментов, относятся задачи уменьшения затрат машинного времени на моделирование, увеличения точности и достоверности результатов моделирования, проверки адекватности модели и т. д.

**Основная задача планирования машинных экспериментов с моделью  $M_m$**  формулируется следующим образом: необходимо получить информацию об объекте моделирования, заданном в виде моделирующего алгоритма (программы), при минимальных или ограниченных затратах машинных ресурсов на реализацию процесса моделирования.

Таким образом, при машинном моделировании необходимо не только рационально планировать и проектировать саму модель  $M_m$  системы  $S$ , но и процесс её использования, т. е. проведение с ней экспериментов с использованием программно-технических средств ЭВМ.

Рассмотрим основные понятия **теории планирования экспериментов**.

Наиболее подходящей моделью процесса проведения эксперимента является абстрактная схема, называемая «чёрным ящиком». При таком кибернетическом подходе различают входные и выходные переменные:  $x_1, x_2, \dots, x_k$ ;  $y_1, y_2, \dots, y_l$ . В зависимости от того, какую роль играет каждая переменная в проводимом эксперименте, она может являться либо фактором, либо реакцией.

Пусть, например, имеют место только две переменные  $x$  и  $y$ . Тогда, если цель эксперимента – изучение влияния переменной  $x$  на переменную  $y$ , то  $x$  – **фактор**, а  $y$  – **реакция**. В экспериментах с машинными моделями  $M_m$  системы  $S$  фактор является экзогенной или управляемой (входной) переменной, а реакция – эндогенной (выходной) переменной. Каждый фактор  $x_i$ ,  $i = 1 \div k$  может принимать в эксперименте одно из нескольких значений, называемых уровнями. Фиксированный набор уровней факторов определяет одно из возможных состояний рассматриваемой системы. Одновременно этот набор представляет собой условия проведения одного из возможных экспериментов.

Каждому фиксированному набору уровней факторов соответствует определённая **точка в многомерном пространстве**, называемом **факторным пространством**. Эксперименты не могут быть реализованы **во всех точках факторного пространства**, а лишь в принадлежащих допустимой области, как например, это показано для случая двух факторов  $x_1$  и  $x_2$  на рис. 4.7 (плоскость  $x_1O_1x_2$ ).

Существует вполне определённая связь между уровнями факторов и реакцией (откликом) системы, которую можно представить в виде соотношения

$$y_l = \psi_l(x_1, x_2, \dots, x_k), l = 1 \div m.$$

Функцию  $\Psi_l$ , связывающую реакцию с факторами, называют **функцией реакции**, а геометрический образ, соответствующий функции реакции, – **поверхностью реакции** (рис. 4.7).

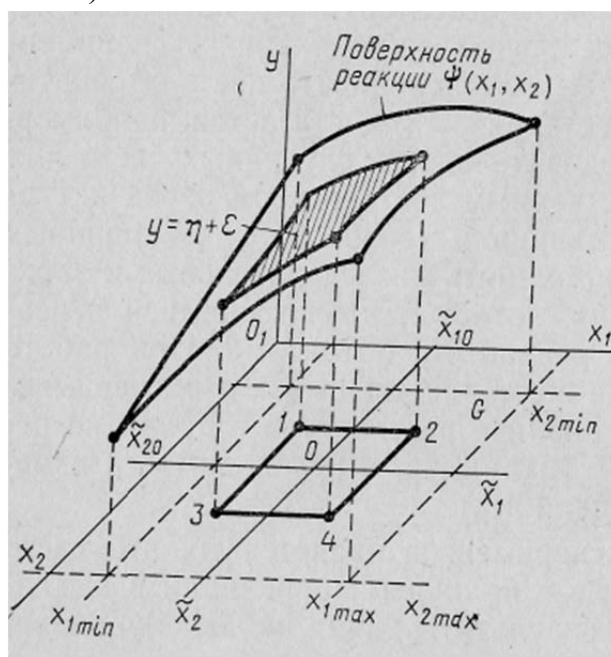


Рис. 4.7. Геометрическое представление поверхности реакции

Исследователю заранее неизвестен вид зависимостей  $\psi_l$ ,  $l = 1 \div m$ , поэтому используют приближённые соотношения:

$$y_l = \varphi_l(x_1, x_2, \dots, x_k), l = 1 \div m.$$

Зависимости  $\varphi_l$  находятся по данным эксперимента, который ставится так, чтобы при минимальных затратах ресурсов (например, минимальном числе испытаний), варьируя значения входных переменных по специально сформулированным правилам, построить математическую модель системы и оценить её характеристики.

При планировании экспериментов необходимо определить основные свойства факторов. Факторы при проведении экспериментов могут быть:

- управляемыми и неуправляемыми;
- наблюдаемыми и ненаблюдаемыми;

- изучаемыми и неизучаемыми;
- количественными и качественными;
- фиксированными и случайными.

**Фактор** называется *управляемым*, если его уровни целенаправленно выбираются исследователем в процессе эксперимента. При машинной реализации модели  $M_m$  исследователь принимает решения, управляя изменением в допустимых пределах различных факторов.

**Фактор** называется *наблюдаемым*, если его значения наблюдаются и регистрируются. Обычно в машинном эксперименте с моделью  $M_m$  наблюдаемые факторы совпадают с управляемыми, так как нерационально управлять фактором, не наблюдая его. Но неуправляемый фактор также можно наблюдать. **Наблюдаемые неуправляемые** факторы получили название *сопутствующих*.

**Фактор** относится к *изучаемым*, если он включен в модель  $M_m$  для изучения свойств системы  $S$ , а не для вспомогательных целей, например, для увеличения точности эксперимента.

**Фактор** будет *количественным*, если его значения – числовые величины, влияющие на реакцию, в противном случае фактор называется *качественным*. Качественным факторам, в отличие от количественных, не соответствует числовая шкала. Однако и для них можно построить условную порядковую шкалу, с помощью которой производится кодирование, устанавливая соответствие между условиями качественного фактора и числами натурального ряда.

Например, в модели системы, формализуемой в виде схемы массового обслуживания, количественными факторами являются:

- интенсивности входящих потоков заявок;
- интенсивности потоков обслуживания;
- ёмкости накопителей;
- количество обслуживающих каналов и т. д.;

а качественными факторами:

- дисциплины постановки в очередь;
- выбора из очереди;
- обслуживания заявок каналами и т. д.

Фактор называется *фиксированным*, если в эксперименте исследуются все интересующие экспериментатора значения фактора, а если экспериментатор исследует только некоторую случайную выборку из совокупности интересующих значений факторов, то фактор называется *случайным*.

В машинных экспериментах с моделями  $M_m$  не бывает неуправляемых или ненаблюдаемых факторов применительно к исследуемой системе  $S$ .

Каждый фактор может принимать в испытании одно или несколько значений, называемых *уровнями*. Причём фактор будет управляемым, если его уровни



целенаправленно выбираются экспериментатором. Для полного определения фактора необходимо указать последовательность операций, с помощью которых устанавливаются его конкретные уровни.

Основными требованиями, предъявляемыми к факторам, являются требования *управляемости фактора* и *непосредственного воздействия на объект*.

Под **управляемостью фактора** понимается возможность установки и поддержания выбранного нужного уровня фактора постоянным в течение всего испытания или изменяющимся в соответствии с заданной программой.

Требование **непосредственного воздействия на объект** предъявляется из-за трудности управления фактором, если он является функцией других факторов.

При проведении машинного эксперимента с моделью  $M_m$  для оценки некоторых характеристик процесса функционирования исследуемой системы  $S$  экспериментатор стремится создать такие условия, которые *способствуют выявлению влияния факторов*, находящихся в функциональной связи с искомой характеристикой.

Для этого необходимо:

- отобрать факторы  $x_i$ ,  $i = 1 \div k$ , влияющие на искомую характеристику, и описать функциональную зависимость;
- установить диапазон изменения факторов  $x_{i \min} \div x_{i \max}$ ;
- определить координаты точек факторного пространства  $\{x_1, x_2, \dots, x_k\}$ , в которых следует проводить эксперимент;
- оценить необходимое число реализаций и их порядок в эксперименте.

#### 4.2.3. Обработка результатов имитационного моделирования

После планирования машинного эксперимента необходимо предусмотреть меры по организации эффективной обработки и представления его результатов.

При выборе методов обработки существенную роль играют три особенности машинного эксперимента с моделью системы  $S$ .

**1. Возможность при моделировании на ЭВМ получать большие выборки** позволяет количественно оценить характеристики процесса функционирования системы, но создаёт проблему хранения промежуточных результатов. Эта проблема решается применением рекуррентных алгоритмов, когда оценки вычисляются по ходу моделирования.

**2. Сложность системы при её моделировании на ЭВМ** приводит к невозможности априорного суждения о характеристиках процесса функционирования системы (например, тип ожидаемого распределения). Поэтому широко используются непараметрические оценки и оценки моментов распределения.

**3. Блочность конструкции машинной модели и раздельное исследование блоков** связано с программной имитацией входных переменных для одной частичной модели по оценкам выходных переменных, полученных на другой частичной модели.

Моделирование вероятностных систем следует рассматривать как статистический эксперимент над моделью системы, отображающей структуру системы и её причинно-следственный механизм.

Необходимо получение доверительных интервалов, а также установление зависимости между временем (числом циклов) моделирования и точностью оценок.

Обозначим через  $\hat{a}(N)$  статистическую оценку параметра  $a$  по данным  $N$  опытов. Оценивание одного и того же параметра, вообще говоря, может быть проведено различными способами. Наилучшими («подходящими») считаются оценки, удовлетворяющие требованиям *состоятельности*, *несмещённости* и *эффективности*.

**1. Оценка называется состоятельной**, если она при неограниченном увеличении числа опытов сходится по вероятности к искомому значению параметра:

$$\lim_{N \rightarrow \infty} P\{|\hat{a}(N) - a| < \varepsilon\} = 1.$$

**2. Оценка является несмещённой**, если её математическое ожидание при любом конечном  $N$  равно истинному значению параметра:

$$M[\hat{a}(N)] = a.$$

**3. Оценка эффективна**, если среди всех возможных она обладает наименьшей дисперсией:

$$M[(\hat{a}_1(N) - a)^2] \leq M[(\hat{a}_i(N) - a)^2].$$

Несмещённой оценкой математического ожидания случайной величины  $X$  является среднее арифметическое результатов  $N$  опытов:

$$\hat{x} = \frac{1}{N} \sum_{i=1}^N X_i.$$

Аналогичными формулами выражаются начальные статистические моменты более высокого порядка:

$$\hat{x}_k = \frac{1}{N} \sum_{i=1}^N X_i^k.$$

Эмпирический контроль сходимости оценок моментов с ростом числа испытаний позволяет убедиться в *существовании* моментов (у распределения Коши все моменты бесконечны).

Многие современные ЭВМ вместо правильного округления младшего из сохраняемых разрядов *отбрасывают* биты, выходящие за разрядную сетку. При суммировании большого количества чисел одного знака это приводит к заметному дрейфу результатов.

Для нейтрализации указанного эффекта можно воспользоваться одним из следующих приёмов:

- работать в формате двойного слова;
- считать средние по сериям наблюдений, а затем определить среднее серий (при этом разница в порядках слагаемых каждой операции сложения уменьшается);
- обрабатывать *центрированные* случайные величины (слагаемые будут знакопеременными, и эффект их усечения взаимно компенсируется).

При расчёте статистических *центральных* моментов приходится вводить поправку, вызванную заменой истинного математического ожидания  $X$  статистическим средним в формулах вида

$$\hat{\mu}_k = \frac{1}{N} \sum_{i=1}^N (X_i - \hat{x})^k, k = 1, 2 \dots$$

В частности, несмещённая оценка для статистической дисперсии

$$\hat{D}_k = \frac{1}{N-1} \sum_{i=1}^N (X_i - \hat{x})^2, k = 1, 2 \dots$$

Статистическое среднеквадратическое отклонение

$$\hat{s} = \sqrt{\hat{D}}.$$

Выборочные средние из-за влияния начальных условий будут, вообще говоря, *смещёнными оценками* истинного значения. Если, например, в начальный момент моделируемая СМО свободна, то несколько первых значений времени ожидания имеют тенденцию быть малыми. Традиционный способ борьбы с этим смещением состоит в *отбрасывании данных за период вхождения в стационарный режим*; однако отнюдь не ясно, какова должна быть длина этого периода, и для гарантированной нейтрализации смещения в ходе моделирования непроизводительно затрачивается много машинного времени.

Все перечисленные оценки *состоятельны* – следовательно, теоретически при существовании стационарного режима и неограниченном возрастании числа наблюдений  $N$  (длительности процесса  $T$ ) сходятся по вероятности к искомым параметрам. Однако при любом конечном  $N$  или  $T$  получаемые оценки *случайны*, в связи с чем возникает вопрос об их качестве (погрешности).

Простейшим способом определения качества оценки является указание её *дисперсии*. Знание дисперсий используется для сопоставления альтернативных оценок одного параметра. В частности, доказано, что *усреднение по системному времени даёт оценки с меньшей дисперсией, чем усреднение по числу наблюдений*.

## 4.3. Программные средства имитационного моделирования

### 4.3.1. Инструментальные средства моделирования

Различают **технические и программные средства ИМ**.

К техническим средствам ИМ относят:

- универсальные цифровые вычислительные системы;
- аналоговые вычислительные машины;
- гибридные моделирующие комплексы;
- стохастические вычислительные машины;
- машины, ориентированные на язык моделирования;
- машины имитационного моделирования;
- распределённые системы моделирования.

**Универсальные ВС** используются в качестве моделирующих систем для программных средств моделирования. Основные требования, предъявляемые к таким ВС, сводятся к наличию большой ёмкости оперативной памяти и высокому быстродействию процессора.

**Аналоговые ВМ** выпускаются с начала развития ВТ и широко применяются для исследования процессов функционирования *непрерывных детерминированных систем*. В АВМ используется структурный принцип управления: отдельные блоки предварительно настраиваются на выполнение определённых операций (суммирование, интегрирование и др.) и соединяются между собой таким образом, чтобы последовательное выполнение операций над переменными, представленными напряжением или током, приводило к решению уравнений, описывающих поведение системы.

**Гибридные моделирующие комплексы (ГМК)** объединяют достоинства дискретных и аналоговых ВМ. В простейшем случае ГМК включает ЭВМ, АВМ, ЦАП, АЦП и БУС (блок управляющих связей). ЦАП и АЦП используются для обеспечения обмена между дискретной и аналоговой частями комплекса. БУС представляет собой средства сопряжения ЭВМ и АВМ по управлению.

**Стохастические вычислительные машины** включают в свой состав электронную аналого-статистическую моделирующую установку (ЭАСМУ), ЭВМ общего назначения, пульт управления и индикации. ЭАСМУ представляет собой стохастическую моделирующую среду, состоящую из однотипных функциональных элементов, каждый из которых может воспроизводить элементарный процесс обслуживания. Структура сетевой модели массового обслуживания набирается из однотипных функциональных элементов.

**Машины, ориентированные на язык моделирования** (GPSS-машина, СИМУЛА-машина) – специализированные машины, архитектура которых ориентирована на определённый язык моделирования. Семантические возможно-

сти языков моделирования служат подходящим средством формализованного описания процессов функционирования сложных систем.

**Машины имитационного моделирования** – это взаимосвязанная совокупность мультипроцессорных систем, предназначенная для анализа и проектирования нестационарных стохастических систем. Идея МИМ основана на том, что исследование систем методом ИМ позволяет естественным образом распараллелить процессы обработки данных. Такие макропроцессы, как имитация системы, сбор и обработка статистики, планирование экспериментов, анализ результатов и оптимизация, диалог с пользователем по управлению моделированием, выполняются параллельно отдельными ВС.

**Распределённые системы моделирования** – наиболее совершенный инструмент исследования и проектирования систем. Данные системы обладают такими структурными свойствами, как интегрированность, распределённость, гибридность, виртуальность, коллективность использования.

**Интегрированность** – возможность автоматизированного решения комплексов задач исследования семантических, синтаксических и динамических свойств системы, а также возможность иерархического решения задач исследования свойств системы на различных уровнях.

**Распределённость** – наличие сетевой организации человеко-машинной среды, в которой выполняются процессы моделирования, распределение основных и сервисных задач моделирования по ресурсам этой сетевой среды.

**Гибридность** – совместное использование эвристических, аналитических, имитационных математических и физических моделей.

**Виртуальность** – организация множества моделирующих комплексов, разделяющих ресурсы сетевой среды.

**Коллективность использования** – организация множества рабочих мест и распределённых контуров их взаимодействия с инструментальными ВС, а также между собой. В качестве инструментальных ВС в них могут использоваться машины имитационного моделирования.

**Программные средства моделирования** разделяют на следующие семейства:

- 1) процедурно ориентированные **алгоритмические языки**;
- 2) проблемно-ориентированные **языки моделирования**;
- 3) автоматизированные **системы моделирования**.

Наибольшие трудозатраты требуются при написании программы на **алгоритмическом языке**, наименьшие – при использовании автоматизированных пакетов. Однако с повышением уровня программных средств уменьшается диапазон моделируемых систем.

**Языки моделирования** имеют более высокий уровень, чем алгоритмические языки программирования.

Типичными операторами **языков моделирования** являются:

- 1) генерировать заявки;
- 2) захватить (определённый прибор);
- 3) занять (определённую ёмкость памяти);
- 4) ждать (истечения определённого интервала времени);
- 5) поместить в очередь;
- 6) освободить (прибор, память);
- 7) исключить (заявку из модели);
- 8) табулировать (собирать и обрабатывать данные по определённой выходной характеристике);
- 9) завершить моделирование и вывести результаты.

**Автоматизированные средства имитационного моделирования** являются средствами программирования наиболее высокого уровня. Они имеют, как правило, универсальную параметрически настраиваемую модель имитации, средства сбора, обработки и представления результатов моделирования, а часто и средства диалога с исследователем, планирования машинных экспериментов и др.

Любой алгоритм имитационного моделирования можно записать на универсальном процедурно ориентированном языке программирования.

***Достоинства универсальных языков программирования:***

- 1) возможность работы на любой ЭВМ;
- 2) нет необходимости в специальном ПО и его изучении;
- 3) логика модели понятна как разработчику, так и программистам;
- 4) использование встроенных функций и подпрограмм штатных систем программирования;
- 5) отсутствие ограничений на состав и логику модели;
- 6) возможность управлять экспериментом и реализовывать методы уменьшения дисперсии.

***Недостатки универсальных языков программирования:***

- 1) возрастание числа элементов и отслеживаемых процессов в модели;
- 2) сложность отладки имитационных моделей из-за взаимозависимости ветвей алгоритма, низкой точности;
- 3) незначительные изменения в системе требуют внесения многочисленных поправок в программу и новой отладки;
- 4) потеря наглядности после составления программы.

Трудности моделирования на **универсальных языках программирования** снимаются при использовании специализированных языков моделирования. Возможность создания языков моделирования основана на том, что количество действий с моделируемыми системами *остается ограниченным* и логика реальных процессов различных систем математически весьма сходна.

### ***Достоинства специализированных языков моделирования:***

- 1) меньшие затраты времени на программирование;
- 2) более эффективные методы выявления ошибок имитации;
- 3) краткость, точность выражения понятий, характеризующих имитационные процессы;
- 4) возможность для некоторой предметной области исследований заранее построить стандартные компоненты;
- 5) автоматическое формирование типов данных, соответствующих принятому способу имитации;
- 6) удобство накопления и представления выходных данных.

### ***Недостатки специализированных языков моделирования:***

- 1) необходимость освоения дополнительных систем программирования;
- 2) ориентация языков на ограниченный класс моделей;
- 3) трудность сбора и обработки результатов, не предусмотренных логикой языка;
- 4) повышение трудоёмкости в реализации модели;
- 5) ограниченные возможности по управлению экспериментом;
- 6) низкая надёжность программ, вызванная скрытой от программиста логикой работы модели и провоцирующими ошибки особенностями языка.

***Автоматизированные средства имитационного моделирования*** являются средствами программирования наиболее высокого уровня. Они имеют, как правило, универсальную параметрически настраиваемую модель имитации, средства сбора, обработки и представления результатов моделирования, а часто и средства диалога с исследователем, планирования машинных экспериментов и др.

Типовой состав системы моделирования приведён на рис. 4.8.



Рис. 4.8. Типовой состав системы моделирования

### 4.3.2. Классификация специализированных программных средств моделирования

**Специализированные языки имитационного моделирования** появились довольно давно, почти одновременно с Алголом и Фортраном, и прошли путь от бурного развития в 70-х годах, когда они ежегодно рождались десятками, до современного стабильного состояния, когда доминирует лишь несколько языков.

В зависимости от классов исследуемых систем выделяют три большие группы языков (систем) моделирования (рис. 4.9):

- 1) языки имитационного моделирования *непрерывных систем*;
- 2) языки имитационного моделирования *дискретных систем*;
- 3) универсальные языки имитационного моделирования (*дискретно-непрерывных систем*).



Рис. 4.9. Классы языков имитационного моделирования

**Языки моделирования непрерывных систем и процессов**, в свою очередь, делятся два типа: на *языки аналогового моделирования* и *языки для решения систем дифференциальных уравнений*, описывающих детерминированные замкнутые непрерывные системы.

В *языках аналогового моделирования* используется компонентная эмуляция аналоговых и гибридных ЭВМ. Например, сумматор заменяется кодом операции суммирования. Взаимодействие между компонентами описывается с помощью блочно-ориентированного языка. Классическим **языком первого типа** является язык **DYNAMO**, разработанный Дж. Форрестером. Язык предназначен для моделирования явлений, представляемых переменными, непрерывными по диапазону своих значений, но дискретными по времени свершения событий.

Языки моделирования непрерывных систем на основе *дифференциальных уравнений* предназначены для моделирования динамических объектов с непрерывным фазовым пространством и непрерывным временем. Как правило, такие объекты описываются с помощью систем дифференциальных (интегро-дифференциальных) уравнений. Системы уравнений могут быть *детерминированными* или *стохастическими*, причём в последнем случае в имитационную систему встраиваются средства статистического моделирования и обработки.



**Примером языка имитационного моделирования второго типа** (стохастическим) является СИМФОР, в котором к возможностям DYNAMO добавлены средства статистического моделирования и обработки. Другим примером может служить пакет «Экспресс-Радиус», разработанный в Институте проблем управления Российской академии наук. В пакете возможности моделирования непрерывных и нелинейных динамических систем дополнены удобным графическим интерфейсом, позволяющим автоматизировать составление программ.

При анализе ВС на системном уровне целесообразно отнести их к классу **дискретных систем**.

В зависимости от подхода к *описанию объектов имитации* и метода *организации квазипараллелизма* языки моделирования *дискретных систем* можно разделить:

- 1) языки, ориентированные на просмотр активностей;
- 2) языки, составляющие расписание событий;
- 3) языки, управляющие процессами;
- 4) языки, организующие взаимодействие транзактов;
- 5) языки, ориентированные на взаимодействие агрегата.

**Языки, ориентированные на просмотр активностей.** В этом случае алгоритм функционирования системы представляется последовательным выполнением активностей. В *управляющей программе моделирования* (УПМ) предусмотрен механизм планирования, который обрабатывает упорядоченный во времени *список моментов завершения* активностей. Когда начинается выполнение некоторой активности, в этот список заносятся соответствующее время завершения и точка возврата в управляющую программу. После выполнения моделирующей программой всех операций возбуждается механизм временного планирования. Он вычеркивает из списка элемент, соответствующий первой по времени завершения активности, устанавливает значение переменной, отображая время моделирования, равным времени завершения соответствующей активности, и передаёт управление в указанную точку управляющей программы. Пример программы – SMPL. Программа-модель составляется из двух частей: проверки условий инициализации активностей и обслуживания активностей.

**Языки, составляющие расписание событий.** Эти языки ориентированы на реализацию *событийного способа имитации*, поэтому программа модели организуется в виде совокупности так называемых процедур обслуживания событий. Выполнение этих процедур синхронизируется во времени списковым механизмом планирования. Каждый элемент списка определяет время события и имя процедуры события. Процедура события определяет, существуют ли условия запуска данной активности, в случае их наличия выполняет необходимые для запуска действия и планирует время её завершения.

Программа, написанная на языке, реализующем событийный способ имитации, зачастую теряет всякое сходство со структурой реальной системы. Это связано с тем, что одним и тем же событиям могут соответствовать активности, аппроксимирующие поведение различных компонент системы. И при интерпретации результатов моделирования исследователю иногда трудно изучать взаимодействие компонент системы по множеству таких объединённых событий в системе. Примеры языков – GASP, **SIMSCRIPT**, CMPL, ДИСМ, МОДИС-ВЕС.

**Языки, управляющие процессами.** Процессный способ имитации получил *наибольшее распространение* из-за того, что в нём сочетаются достоинства имитации активностями и событиями. В этих языках моделирующая программа организуется в виде набора описаний процессов, каждое из которых соответствует одному классу процессов. Описание процесса объединяет активности всех процессов данного класса и имеет форму процедуры, которая выполняется одновременно для всех представителей этого класса, существующих в системе в текущее время.

Программа имитационной модели, написанная на языке, реализующем *процессный способ имитации*, имеет ту же структуру, что и объект моделирования. Каждой компоненте системы соответствует своё описание процесса, что облегчает исследователю интерпретацию результатов моделирования. Примеры языков, ориентированных на процессы – SOL, ASPOL, SIMULA, ДИС, СЛЕНГ-Б, АЛСИМ, СИМУЛА-ИПМ, **СИМУЛА-67**, МПЛ/1, СКИФ-Ф, МК PLSIM,

**Языки, организующие взаимодействие транзактов.** В языках, реализующих транзактный способ имитации, в качестве динамических объектов при описании программы моделей используются транзакты (заявки на обслуживание). Каждому транзакту может принадлежать определённый набор параметров, меняемых соответствующими блоками. Программа-модель на таком языке представляется в виде схемы, отображающей пространственное перемещение транзактов в системе. Затем осуществляется символическое кодирование схемы, при котором каждому блоку ставится в соответствие строка символов, отражающих текст программы. Управляющая программа моделирования создаёт транзакты, передвигает их по определённым блокам и производит действия, связанные с обслуживанием транзакта каждым блоком. По мере прохода транзактов через блоки происходит сдвиг модельного времени. Типичными представителями таких языков являются **GPSS**, МОДЕЛЬ-6, ИМСС, CSS, АСИМ.

**Языки, ориентированные на взаимодействие агрегатов.** В таких языках моделирования в качестве динамических объектов выступают агрегаты. В функции управляющей программы моделирования входит *синхронизация взаимодействия агрегатов*. Она распознает, какое событие произошло в агрегате, управляет выдачей агрегатом выходных сигналов, синхронизирует работу

в агрегатной системе, организует передачу выходных сигналов агрегата на входные полюсы других агрегатов. Сами же агрегаты имитируют поведение компонент сложной системы. Агрегатный метод имитации используется в системах моделирования АИС и САПАС.

В последнее время развиваются языки моделирования **дискретно-непрерывных систем**, такие системы как НЕДИС, GASP-IV, СИМФОР, УСМ.

В зависимости *от степени универсальности* моделируемых систем выделяют:

- универсальные средства моделирования;
- предметно-ориентированные средства моделирования.

Чем больше универсальность средства моделирования, тем более простая схема формализации должна использоваться для разработки модели. Для предоставления возможности учитывать в модели *специфические особенности* систем различных классов и предметных областей используются два подхода:

- 1) в средство моделирования вводится избыточность;
- 2) средство строится как открытая система, доступная к дополнению.

Предметно-ориентированное средство моделирования предоставляет пользователю формализованные возможности, соответствующие предметному мышлению в той области, для которой предполагается её использование.

Приведём в табл. 4.1 с классификацию соответствующих языков и систем моделирования.

Таблица 4.1

#### Классификация систем имитационного моделирования

Вид имитации	Способ организации имитации	Наименование системы моделирования
непрерывные системы	Аппроксимацией процессов	Dynamo
	Системами уравнений	360/System CSMP
дискретные системы	Активностями	SMPL
	Событиями	СИМСКРИПТ, ДИСМ, МОДИС-ВЕС, GASP, CMPL
	Процессами	АЛСИМ, СЛЕНГ-Б, ДИС, SOL, ASPOL, SIMULA, СИМУЛА-ИПМ, СИМУЛА-67, МПЛ/1, СКИФ-Ф, МК PLSIM
	Транзактами	МОДЕЛЬ-6, ИМСС, GPSS, АСИМ, CSS
	Агрегатами	АИС, САПАС
дискретно-непрерывные системы	Событиями	НЕДИС
	Процессами	DISLIN

### 4.3.3. Обзор специализированных языков моделирования

Известно множество систем моделирования. В книге «Языки моделирования» Е. Киндлера (1985) упоминаются в той или иной степени свыше 350 различных систем. Языки Симула и GPSS входят в число **13 наиболее употребляемых языков**.

Наиболее известными являются системы GPSS, SOL, Simula, SMPL, FORSIM, SIMSCRIPT, ПЛИС, Pascal+ (расширенный встроенными средствами моделирования). Ряд языков моделирования, таких как СЛЭНГ, НЕДИС, АР-ГОН и др., был разработан в Советском Союзе.

Языки моделирования в качестве основы используют универсальные языки программирования или другие языки моделирования. Языки SIMSCRIPT, SMPL и FORSIM созданы на основе алгоритмического языка Фортран. Языки SIMULA и SOL созданы на основе языка Алгол. ПЛИС – на GPSS и PL-1.

#### **SIMSCRIPT**

Язык **SIMSCRIPT** является универсальным средством моделирования, созданным на базе языка ФОРТРАН. Центральными понятиями служат обработка списков с компонентами, определённым пользователем, и последовательность событий.

#### **СИМУЛА-67**

Особое место среди языков имитационного моделирования занимает СИМУЛА-67, разработанный в Норвежском вычислительном центре У.И. Далом, Б. Мюрхаугом и К. Ньюгордом. Язык SIMULA-67 – универсальный язык высокого уровня, являющийся расширением языка АЛГОЛ. Возможность его использования для моделирования определяется тем, что в нём имеются средства параллельного оперирования процессами в системном времени и универсальной обработки списочных наборов данных. Однако в языке отсутствуют объекты с заранее определёнными свойствами типа «устройство» или «память». Такие элементы должен определить сам пользователь. Также пользователь определяет все операции по ведению очередей.

В SIMULA-67 впервые получила практическое воплощение *концепция ядра языка* как средства иерархического, структурированного описания класса объектов, – концепция, последующее развитие которой привело к созданию объектно-ориентированного программирования. Термином «объект» в СИМУЛА были обозначены программные компоненты, обладающие собственными локальными данными (атрибутами) и способные выполнять некоторые действия. В роли атрибутов могут выступать прочие программные компоненты, переменные, массивы и ссылки на другие объекты. Действия, выполняемые объектом, задаются с помощью последовательности операторов, называемых сценарием функционирования. Каждый объект обладает системным атрибутом, указыва-

ющим на исполняемый оператор его правил действий, который называется локальным управлением. Во время работы СИМУЛА-программы могут сосуществовать несколько объектов, находящихся на разных стадиях исполнения своих сценариев функционирования, а управляющая программа передаёт управление от одного объекта к другому, активизируя их по заложенному в ней сценарию. Исключительность СИМУЛА заключается в том, что он является весьма удобным средством создания самих языков имитационного моделирования. Средствами СИМУЛА довольно легко реализовать, скажем, GPSS или СЛАМ, не говоря уже о более простых языках моделирования, таких, как DYNAMO. Развитые средства организации и ведения библиотек позволяют легко построить любой язык моделирования как совокупность некоторых классов, аналогичных рассмотренным.

### СЛЭНГ

Язык СЛЭНГ – АЛГОЛ-ориентированный язык системного моделирования с хорошими алгоритмическими возможностями. Он содержит все средства описания систем, имеющиеся в языке SOL, а также использует ряд принципов языка Simula-67. В этот язык введены явные средства обработки списков, средства синхронизации, прерывания и активизации процессов. Он также реализован на ЭВМ типа БЭСМ-6.

### GPSS, GPSS World

Одним из наиболее распространённых языков моделирования дискретных систем является GPSS (*General Purpose Simulation System – общецелевая система моделирования*). Это интерпретирующая языковая система. В составе языка определены структурные элементы типа «устройство», «память» и «логический ключ», однако какие-либо внутренние средства управления ими отсутствуют. Язык основан на принципах *транзактного имитационного моделирования*. С 1961 года язык выдержал множество модификаций для самых различных ОС и ЭВМ и в то же время сохранил почти неизменными внутреннюю организацию и основные блоки. Первоначально GPSS был реализован на больших ЭВМ. В 1984 году появилась версия для ПЭВМ под ОС DOS – GPSS/PC. В 1994 году появилась система GPSS World для OS/2, а в 2000 году она была реализована для Windows.

Основными понятиями языка GPSS являются: транзакт, блок, оператор.

**Транзакт** GPSS – это динамический объект, под которым может подразумеваться клиент, требование, вызов или заявка на обслуживание прибором обслуживания. То есть основное назначение GPSS – это *моделирование систем массового обслуживания*, хотя наличие дополнительных встроенных средств позволяет моделировать и некоторые другие системы (например, распределение ресурсов между потребителями). Транзакты в GPSS могут создаваться (вводиться),

уничтожаться (выводиться), задерживаться, размножаться, сливаться, накапливаться и т. д. Именно благодаря наличию транзакта GPSS обладает такой изящностью и простотой.

**Блок GPSS** представляет собой некоторый самостоятельный элемент моделируемой системы. Каждый блок реализует одну или несколько операций над транзактом, группой транзактов или параметрами транзактов, а совокупность блоков составляет моделирующую программу. Таким образом, GPSS имеет *блочную структуру* и, вообще говоря, легко может быть приспособлен и для структурно-функционального моделирования не очень сложных систем. GPSS достаточно лёгок в освоении, а наличие в нём функций, переменных, стандартных атрибутов, графики и статистических блоков существенно расширяет его возможности.

В конце 80-х интерес к GPSS стал угасать, что можно объяснить двумя обстоятельствами. Во-первых, в эти годы существенно усложнились объекты анализа: они стали в основном иерархическими, с большим количеством взаимосвязей и лучше описываются сетевыми моделями, – а надо признать, что GPSS малоприспособлен для моделирования сетей. Во-вторых, именно в эти годы значительных успехов достигли языки объектно-ориентированного программирования, позволяющие строить исключительно гибкие инструментальные средства имитационного моделирования. Однако в последние годы интерес к GPSS вновь возрос.

В настоящее время язык GPSS и его современная версия GPSS World, разработанная компанией Minuteman Software (США) – одно из наиболее эффективных инструментальных средств имитационного моделирования.

Встроенные средства GPSS World обеспечивают моделирование случайных величин 24-х видов законов распределения. Не составляет труда применить и собственные эмпирические распределения.

Для многих систем программирования наиболее наглядной формой машинно-ориентированного описания решаемой проблемы являются *блок-схемы*. Язык блок-схем принят обычно и при описании дискретных систем перед созданием для их моделирования программ на GPSS. Однако язык блок-схем для GPSS отличается от языка традиционных блок-схем для алгоритмических языков программирования.

GPSS World обладает высокой интерактивностью, диалоговыми возможностями, многозадачностью, средствами наладки, возможностью работы в пошаговом режиме. Для визуализации процесса моделирования предусмотрены динамические окна, со стилизованным отображением объектов любого существующего типа. Для доступа к микросостояниям процесса моделирования предусмотрен механизм кадров.

GPSS World не имеет встроенных графических средств проектирования модели, а также средств анимации модельного эксперимента. Концепция, заложенная при разработке GPSS World, для осуществления данных задач предусматривает использование специальных приложений, при этом средства GPSS World позволяют обеспечить эффективный процесс взаимодействия с такими приложениями (использование потоков данных и возможность вызова исполняемых файлов).

Для расширения возможностей в состав пакета GPSS World встроен язык более низкого уровня **PLUS**, обеспечивающий высокий уровень гибкости создаваемых моделей. Использование средств данного языка, кроме пофакторного дисперсионного анализа, позволяет автоматизировать процесс создания и проведения экспериментов, в том числе оптимизационных.

Возможности современных ВС позволяют проводить весьма значительное количество модельных экспериментов. Для поиска оптимального решения в зависимости от сложности задачи возможно использование как полного перебора всех альтернативных вариантов, так и применение эвристических методов поиска. Средства языка PLUS позволяют сделать как первое, так и второе.

Таким образом, GPSS World отвечает основной части указанных требований и является эффективным инструментальным средством имитационного моделирования функционирования ВС высокой сложности.

## **SOL**

Универсальный язык SOL включает все средства описания систем, присущие языку GPSS, в гораздо более общей и законченной форме. Язык является АЛГОЛ-ориентированным. Центральным в языке SOL служат взаимосвязанные понятия класса (потока) заявок и процесса как алгоритма, описывающего поведение некоторого класса. Описание функционирования модели представляется в виде совокупности взаимодействующих процессов. Процессы оперируют с некоторым набором ресурсов (устройство, память). В языке SOL предусмотрены стандартные средства автоматического сбора статистических данных об использовании ресурсов системы.

## **СЛАМ**

К числу широко известных языков имитационного моделирования относится также язык СЛАМ, разработанный профессором Университета Пердью А. Прицкером в начале 70-х и с тех пор постоянно совершенствуемый созданной в 1973 году компанией Pritsker Corporation. О популярности СЛАМ и вообще о широте использования методов имитационного моделирования можно судить по масштабам деятельности фирмы Прицкера: через сервер компании, расположенный в Индианаполисе, программное обеспечение устанавливали более чем шесть тысяч пользователей в 40 странах мира; в настоящее время

фирма оказывает консультационные работы для двух тысяч клиентов. Организована сеть консалтинговых фирм, связанных между собой через Internet, в 19 странах Азии, Европы, Африки и Южной Америки. Более тысячи преподавателей университетов используют для обучения студентов системы имитационного моделирования, основанные на СЛАМ.

В основе языка СЛАМ лежит простая идея – объединить достоинства GPSS и DYNAMO таким образом, чтобы, допуская раздельное применение этих языков, можно было при необходимости использовать их совместно. Реализация этого принципа на ЭВМ с цифровыми дисплеями хотя и давала некоторые преимущества при моделировании, однако не вносила качественных изменений в процесс моделирования. Переход к графическим интерфейсам раскрыл все преимущества этого принципа.

### ПЛИС, ДИСМ

На базе GPSS и ПЛ/1 была разработана система ПЛИС. Её возможности существенно шире GPSS, поскольку допускают использование всех средств языка ПЛ/1. Другим примером дополнения языка ПЛ/1 средствами имитационного моделирования служит система ДИСМ.

### Simpas

Ещё одним языком имитационного моделирования дискретных систем является Simpas, который, по мнению разработчиков, по своим возможностям полностью перекрывает специализированный язык GPSS, обладая более высоким быстродействием и гибкостью. Система Simpas оформлена как модуль и потому является открытой по сравнению с GPSS/PC. Это даёт возможность исследователю разрабатывать свои процедуры и включать их в Simpas или подстраивать что-то из уже имеющегося в нём под свои нужды. Simpas позволяет использовать возможности Object Pascal. Правда, Simpas предназначен для работы в ОС MS DOS.

## 4.4. Основы построения языка GPSS

### 4.4.1. Объекты языка GPSS

Язык GPSS является языком высокого уровня, ориентированным на описание функционирования дискретных во времени систем с детерминированными или стохастическими параметрами. На основе языка GPSS, улучшенного *встроенным языком низкого уровня PLUS*, построена система GPSS World, предназначенная для моделирования систем с дискретными и непрерывными процессами.

Язык GPSS построен в предположении, что модель сложной системы можно представить в виде совокупности элементов и логических правил их взаимодействия. Абстрактные элементы, описывающие систему, называют объектами.



Объекты подразделяются на 7 категорий и 15 типов (табл. 4.2).

Объект каждого типа характеризуется определённым способом поведения и набором атрибутов. Основная идея моделирования – взаимодействие блоков и транзактов.

Таблица 4.2

### Категории и типы объектов GPSS

Категории	Типы объектов		
Аппаратная	Устройства (ОКУ)	Памяти (МКУ)	Логические ключи
Динамическая	Транзакты		
Операционная	Блоки		
Вычислительная	Переменные	Функции	Генераторы случайных чисел
Статистическая	Очереди	Таблицы	
Запоминающая	Ячейки	Матрицы ячеек	
Группирующая	Числовые группы	Группы транзактов	Списки

**Аппаратные объекты – устройства, памяти (многоканальные устройства – МКУ) и ключи** – абстрактные элементы, на которые может быть разделена реальная моделируемая система. Транзакты, воздействуя на них, могут изменять их состояние и влиять на движение других транзактов.

**Динамические объекты – транзакты** – создаются в определённых точках модели, перемещаются в системном времени и пространстве с помощью планировщика, переходя от блока к блоку и воздействуя на них. Транзакты возникают и уничтожаются, могут расщепляться и сливаться. Входя в блок, транзакт вызывает определённую подпрограмму, которая обрабатывает соответствующее событие. Далее транзакт пытается войти в следующий блок. Продвижение продолжается до тех пор, пока блоку не потребуется выполнить одну из функций:

- удалить транзакт из модели;
- временно заблокировать его в предыдущем блоке до выполнения некоторых условий;
- задержать его на определяемое моделью время.

После этого начинается продвижение другого транзакта и т. д. За один шаг работы управляющей программы производится поочерёдный просмотр всех процессов и имитация тех из них, которые могут быть запущены. Описание траектории транзакта содержит порядок и имена используемых устройств или памятей, временные задержки, логические условия, управляющие продвижением транзактов, точки маршрута, в которых производится сбор данных. Некоторые из объектов должны быть описаны специальными командами. Траектория и задержки могут зависеть от ситуации в системе и от параметров транзакта. Маршруты следования, временные характеристики, условия прекращения моделирования, показатели эффективности задаются пользователем.

**Операционные объекты – блоки** – задают логику функционирования модели и определяют пути продвижения транзактов между объектами аппаратной категории. Блоки GPSS представляют собой подпрограммы на макроассемблере или языке С. Передача управления от блока к блоку реализуется с помощью движения транзактов в модельном времени.

В блоках могут происходить следующие события:

- создание или уничтожение транзактов;
- изменение числового атрибута объекта;
- задержка транзакта;
- изменение маршрута движения транзакта.

В GPSS World реализовано 53 типа блоков в 9 группах (табл. 4.3).

Таблица 4.3

### Классификация блоков GPSS по группам

№ п/п	Группы	Кол-во	Блоки
	<b>Блоки, осуществляющие модификацию атрибутов транзактов</b>	<b>12</b>	
а)	генерирование и уничтожение транзактов		GENERATE, SPLIT, TERMINATE, ASSEMBLE
б)	временная задержка		ADVANCE
в)	синхронизация движения транзактов		MATCH, GATHER
г)	изменение параметров транзактов		ASSIGN, INDEX, MARK, PLUS
д)	изменение приоритета транзакта		PRIORITY
	<b>Блоки, изменяющие последовательность движения транзактов (блоки передачи управления)</b>	<b>5</b>	DISPLACE, TRANSFER, LOOP, TEST, GATE
	<b>Блоки, связанные с группировкой категории</b>	<b>6</b>	ADOPT, ALTER, EXAMINE, JOIN, REMOVE, SCAN
	<b>Блоки, описывающие объекты аппаратной категории</b>	<b>11</b>	
а)	одноканальные устройства		SEIZE, RELEASE, PREEMPT, RETURN, FUNAVAIL, FAVAIL
б)	памяти (многоканальные устройства)		ENTER, LEAVE, SAVAIL, SUNAVAIL
в)	ключи (логические переключатели)		LOGIC
	<b>Блоки, сохраняющие необходимые значения для дальнейшего использования</b>	<b>2</b>	SAVEVALUE, MSAVEVALUE
	<b>Блоки, обеспечивающие получение статистических результатов</b>	<b>3</b>	
а)	очереди		QUEUE, DEPART
б)	таблицы		TABULATE
	<b>Блоки для организации списка пользователя</b>	<b>2</b>	LINK, UNLINK
	<b>Блоки для организации ввода-вывода</b>	<b>5</b>	
а)	открытие/закрытие		OPEN, CLOSE
б)	считывание/запись в файл		READ, WRITE
в)	установка позиции текущей строки		SEEK
	<b>Специальные блоки</b>	<b>7</b>	BUFFER, COUNT, EXECUTE, INTEGRATION, SELECT, TRACE, UNTRACE

Также блоки и СЧА можно сгруппировать по функциональным комплексам (табл. 4.4).

Таблица 4.4

**Группировка блоков GPSS по функциональным комплексам**

№ п/п	Группы	Кол-во	Блоки
1.	<b>Траекторный комплекс</b>	9	
а)	порождающие и уничтожающие транзакты		GENERATE, TERMINATE
б)	изменяющие траекторию		TRANSFER, LOOP, TEST, GATE, DISPLACE, EXECUTE
в)	определяющий временные задержки		ADVANCE
2.	<b>Атрибутный комплекс</b> (средства задания и изменения значений, обрабатываемых моделью)	11	ASSIGN, COUNT, MARK, INDEX, PRIORITY, PLUS, SAVEVALUE, MSAVEVALUE, SELECT, LOGIC, INTEGRATION
3.	<b>Ансамблевый комплекс</b> (формирование и взаимодействие ансамблей и групп транзактов)	10	SPLIT, MATCH, ASSEMBLE, GATHER, ADOPT, ALTER, JOIN, SCAN, EXAMINE, REMOVE
4.	<b>Ресурсный комплекс</b> (работа с устройствами и памятью)	8	SEIZE, RELEASE, FUNAVAIL, FAVAIL, ENTER, LEAVE, SAVAIL, SUNAVAIL
5.	<b>Цепной комплекс</b> (цепи пользователя, событий)	5	PREEMPT, RETURN, LINK, UNLINK, BUFFER
6.	<b>Комплекс управления</b>	5	QUEUE, DEPART, TABULATE, TRACE, UNTRACE
7.	<b>Комплекс ввода-вывода</b>	5	OPEN, CLOSE, SEEK, READ, WRITE

В набор базовых средств языка GPSS входят 53 блока, 25 команд, 35 СЧА, 12 типов операторов языка PLUS. Команды определяют объекты GPSS, т. е. играют роль описаний или некоторым образом влияют на состояние формулируемой модели.

Команды GPSS в соответствии со своим назначением делятся на две группы:

- команды определения объектов GPSS;
- команды управления процессом моделирования.

**Команды определения объектов GPSS**

- 1) B VARIABLE – определить булеву переменную;
- 2) EQU – назначить значение пользовательской переменной;
- 3) FUNCTION – определить функцию;

- 4) FVARIABLE – определить арифметическую переменную (промежуточные значения не округляются в режиме совместимости с GPSS/PC );
- 5) INITIAL – задать начальные или новые значения ключа, сохраняемого значения или матрицы;
- 6) MATRIX – определить матрицу;
- 7) QTABLE – определить таблицу для очереди;
- 8) RMULT – задать установки для первых семи ГСЧ;
- 9) START – установить счётчик завершений и начать моделирование;
- 10) STORAGE – определить память;
- 11) VARIABLE – определить арифметическую переменную (промежуточные значения округляются до целого значения в режиме совместимости с GPSS/PC);
- 12) TABLE – определить таблицу.

### ***Команды управления процессом моделирования***

- 1) CLEAR – сбросить статистику и удалить транзакты;
- 2) CONDUCT – провести PLUS-эксперимент;
- 3) CONTINUE – возобновить прерванное моделирование;
- 4) EXIT – закончить сеанс с GPSS World;
- 5) HALT – остановить моделирование и удалить все отсроченные команды;
- 6) INCLUDE – прочитать и транслировать дополнительный файл;
- 7) INTEGRATE – автоматически проинтегрировать переменную по времени;
- 8) REPORT – задать имя файла для отчёта или выдать отчёт немедленно;
- 9) RESET – сбросить статистики моделирования;
- 10) SHOW – вычислить и высветить выражение;
- 11) STEP – сделать установленное число попыток входа в блок;
- 12) STOP – задать условия останова по числу попыток входа в блок.

**Объекты вычислительной категории – переменные, функции, ГСЧ** – служат для описания взаимосвязей между компонентами модели в виде математических соотношений. **Переменные** – сложные выражения, включающие константы, системные числовые атрибуты (СЧА), библиотечные функции, арифметические и логические операции, переменные пользователя. **Выражения** могут применяться в переменных и операторах GPSS. В переменных выражения определяются командами GPSS, в операторах GPSS – определяются как часть языка PLUS. Каждому объекту соответствуют атрибуты, описывающие его состояние в данный момент времени. Они доступны для использования в течение всего процесса моделирования и называются **СЧА**. **Булевы переменные** позволяют проверять в одном блоке GPSS одновременно несколько условий, исходя из состояния или значения этих условий и их атрибутов.

**Функции** задаются табличным способом и определяют непрерывные или дискретные функциональные зависимости между аргументом функции и зависимым значением функции. Кроме того, имеется 24 встроенных **ГСЧ**.

**Статистические объекты – очереди и таблицы** – предназначены для сбора статистических данных в местах ожидания транзактов.

**Объекты запоминающей категории – ячейки и матрицы** – обеспечивают обращение к сохраняемым значениям числовой информации. Любой транзакт может записать и считать эту информацию. Матрицы поддерживают 6 измерений.

**Объекты группирующей категории – числовые группы, группы транзактов и цены** – предоставляют возможность пользователю обращаться к групповым атрибутам транзактов.

#### 4.4.2. Основные блоки GPSS и построение простейших моделей

Итак, всего существует 53 различных блока GPSS. Приведём их перечень и назначение (табл. 4.5).

В таблице используется символ **О** для обозначения логического оператора или оператора отношения.

Таблица 4.5

Назначение блоков GPSS

№	Блок	Описание
1.	ADOPT A	изменить ансамбль транзакта
2.	ADVANCE A,B	задержать транзакт на заданное время в цепи будущих событий
3.	ALTER <b>О</b> A,B,C,D,E,F,G	проверить и модифицировать транзакт в группе
4.	ASSEMBLE A	ждать накопления транзактов и разрушить копии
5.	ASSIGN A,B,C	изменить параметр транзакта
6.	BUFFER (без операндов)	начать сначала просмотр цепи текущих событий
7.	CLOSE A,B,C	конец потока данных
8.	COUNT <b>О</b> A,B,C,D,E	подсчитать объекты и записать их количество в параметр транзакта
9.	DEPART A,B	уменьшить очередь
10.	DISPLACE A,B,C,D	изменить транзакту следующий блок
11.	ENTER A,B	занять или ожидать занятия нескольких единиц памяти
12.	EXAMINE A,B,C	проверить принадлежность к группе
13.	EXECUTE A	выполнить действие, определённое указанным здесь блоком
14.	FAVAIL A	снять запрет на вход транзактов в устройство
15.	FUNAVAIL A,B,C,D,E,F,G,H	запретить вход транзактов в устройство
16.	GATE <b>О</b> A,B	проверить объект и изменить маршрут транзакта
17.	GATHER A	ждать транзакты одного семейства до сбора требуемого их количества
18.	GENERATE A,B,C,D,E	создать транзакт и поместить его в ЦБС
19.	INDEX A,B	изменить параметр транзакта
20.	INTEGRATION A,B	включить или выключить интегрирование переменной
21.	JOIN A,B	ввести элемент в числовую группу или группу транзактов

№	Блок	Описание
22.	LEAVE A,B	освободить часть памяти
23.	LINK A,B,C	перевести транзакт в ПЦ
24.	LOGIC O A	изменить состояние логического ключа
25.	LOOP A,B	уменьшить значение параметра на единицу и перейти к указанному блоку, если результат больше нуля
26.	MARK A	ввести в параметр транзакта значение модельного времени
27.	MATCH A	ждать входа связанного транзакта в сопряжённый блок MATCH
28.	MSAVEVALUE A,B,C,D	изменить элемент матрицы
29.	OPEN A,B,C	открыть поток данных
30.	PLUS A,B	вычислить PLUS-выражение и записать его в параметр транзакта
31.	PREEMPT A,B,C,D,E	сменить владельца устройства
32.	PRIORITY A,B	изменить приоритет транзакта
33.	QUEUE A,B	увеличить длину очереди
34.	READ A,B,C	прочитать следующую строку из потока данных
35.	RELEASE A	освободить устройство
36.	REMOVE O A,B,C,D,E,F	удалить элемент из числовой или транзактной группы
37.	RETURN A	освободить устройство от транзакта, если оно занято через PREEMPT
38.	SAVAIL A	снять запрет на вход транзактов в память
39.	SAVEVALUE A,B	изменить сохраняемое значение
40.	SCAN O A,B,C,D,E,F	проверить группу транзактов и записать число членов группы в параметр активного транзакта
41.	SEEK A,B	сменить указатель строки в потоке данных
42.	SEIZE A	занять устройство или ждать этой возможности
43.	SELECT O A,B,C,D,E,F	записать в параметр номер выбранного объекта
44.	SPLIT A,B,C	создать копии транзакта
45.	SUNAVAIL A	запретить вход транзактов в память
46.	TABULATE A,B	изменить содержимое таблицы
47.	TERMINATE A	вывести транзакт из модели, уменьшить счётчик завершений карты START
48.	TEST O A,B,C	проверить арифметическое условие и изменить маршрут транзактов
49.	TRACE (без)	начать трассировку активного транзакта
50.	TRANSFER A,B,C,D	перейти к указанному блоку
51.	UNLINK O A,B,C,D,E,F	вывести транзакт из цепи пользователя
52.	UNTRACE (без)	закончить трассировку активного транзакта
53.	WRITE A,B,C,D	направить значение в поток данных

**Рассмотрим подробнее некоторые группы блоков GPSS.**

### **1. Поступление транзакта в модель.**

**GENERATE [A], [B], [C], [D], [E], [F]** (Генерировать)

A – средний интервал времени между последовательными поступлениями транзактов в модель;

B – модификатор (интервал или функция), изменяющий значения интервала генерации транзактов по сравнению с интервалом, указанным операндом A;

C – время поступления первой транзакции (по умолчанию 0);

D – ограничитель количества создаваемых транзакций;

E – уровень приоритета (целое число от 0 до 127);

F – количество числовых параметров (до 100) у создаваемых транзакций.

A и B могут быть именем, положительным числом, выражением в скобках, СЧА.

A–B – нижняя граница.

A+B – верхняя граница.

Более сложные интервалы времени поступления транзактов (не по равномерному закону) задаются модификатором-функцией.

Смещение интервалов C – это момент времени, в который должен появиться первый транзакт.

D – граничное значение общего числа транзактов, которые могут войти в модель через данный блок в течение времени моделирования. Если значение не определено, блок остается активным в течение всего времени моделирования.

E – класс приоритета каждого из транзактов. Чем больше число, тем выше приоритет.

Обязательно должен быть задан либо операнд A, либо D.

### **2. Удаление транзактов из модели.**

Для удаления транзактов служит блок:

**TERMINATE [A]** (Завершить)

B этот момент освобождается память, выделенная под транзакт.

Операнд A – величина уменьшения специального счётчика завершения.

### **3. Занятие и освобождение одноканального устройства.**

**SEIZE A** (Занять)

Транзакция, поступающая в блок, либо занимает прибор A, либо ставится в очередь, если прибор занят.

A – идентификатор (число или имя) занимаемого прибора.

**RELEASE A** (Освободить)

Транзакция освобождает прибор, указанный в аргументе A.

A – идентификатор (число или имя) освобождаемого прибора.

#### **4. Имитация обслуживания путём задержки во времени.**

##### **ADVANCE A, [B] (Задержать)**

Задерживает транзакцию на время, определяемое аргументами A и B.

A – среднее время задержки или вероятностное распределение из встроенной библиотеки процедур, заключённое в круглые скобки;

B – величина полуинтервала равномерно распределённого интервала задержки или модификатор таблично заданной функции (по умолчанию – ноль).

#### **5. Сбор статистики.**

##### ***Регистрация очереди:***

##### **QUEUE A, [B] (Стать в очередь)**

Ставит транзакцию в очередь, имя которой указано в A. Блок QUEUE создаёт явно заданную очередь, которую можно использовать для управления транзакциями и сбора статистики.

A – идентификатор (число или имя) очереди;

B – количество элементов, на которое должна увеличиться длина очереди; [по умолчанию – один].

##### **DEPART A, [B] (Покинуть очередь)**

Удаляет транзакцию из очереди, имя которой указано в аргументе A.

A – идентификатор (число или имя) очереди;

B – количество элементов, на которое должна уменьшиться длина очереди; [по умолчанию – один].

##### ***Статистические таблицы:***

##### **Name TABLE A, B, C, D**

Используется для получения плотности распределения и её интегральных носительных частот, среднего и стандартного отклонения аргументов.

A – элемент данных, чьё частотное распределение будет табулироваться (имя, выражение в скобках, СЧА);

B – верхний предел первого интервала;

C – ширина частотного интервала (разница между верхней и нижней границей каждого частотного класса);

D – число частотных интервалов.

Для сбора данных транзакт должен войти в блок TABULATE с тем же именем таблицы, которое определено в блоке TABLE.

##### **TABULATE A, [B]**

A – имя таблицы;

B – весовой коэффициент, число единиц, которое должно быть занесено.

##### **Name QTABLE A,B,C,D**

Используется для получения распределения только времени пребывания транзакта в очереди.



A – имя очереди;

B,C,D – такие же, как в TABLE.

## **6. Занятие и освобождение многоканального устройства (МКУ).**

### **ENTER A, [B] (Войти)**

Занимает B приборов в МКУ A (по умолчанию – один).

A – идентификатор (число или имя) многоканального устройства;

B – количество занимаемых приборов многоканального устройства;

[по умолчанию – один].

### **LEAVE A, [B] (Выйти)**

Освобождает B приборов из A возможных (по умолчанию – один).

A – идентификатор (число или имя) многоканального устройства;

B – количество освобождаемых приборов многоканального устройства;

[по умолчанию – один].

### **Name STORAGE A**

Предварительно перед использованием блоков ENTER и LEAVE с помощью команды STORAGE определяется количество приборов в МКУ.

## **7. Изменение маршрута движения транзактов.**

### ***Проверка состояния булевой переменной:***

#### **TEST X A, B, [C] (Проверить)**

Сравнивает значения указанных СЧА и управляет направлением движения транзакта на основе результатов сравнения.

X – условный оператор (L – меньше, LE – меньше или равно, E – равно, NE – не равно, G – больше, GE – больше или равно).

A и B – проверяемое и контрольное значение (имя, число, строка, выражение в скобках, СЧА, СЧА\*параметр)

C – имя (метка) блока назначения C.

Два режима:

- отказа во входе (если операнд C не задан);
- разрешения во входе и альтернативном выходе (если C задан).

### ***Проверка состояния логического ключа:***

#### **GATE X A, [B] (Впустить)**

Изменяет движение транзакта по состоянию проверяемого объекта – логического ключа, устройства или МКУ.

X – условный оператор (NU – устройство, заданное A, свободно; U – устройство, заданное операндом A, занято);

A – имя или № проверяемого устройства;

B – номер следующего блока для входящего транзакта, когда условный оператор имеет значение «ложь».

Два режима:

- отказа во входе;
- разрешения во входе и альтернативном выходе.

GATE U Zrk – в случае незанятости устройства Zrk транзакт не будет пропущен блоком GATE.

GATE NU Bat, Oper – в случае занятости устройства Bat транзакт будет направлен к блоку Oper.

***Передача транзакта в другой блок:***

**TRANSFER [A],[B],[C],[D]** (Перемещать)

A – режим выбора

- 1) режим безусловной передачи;
- 2) режим статистической передачи;
- 3) режимы BOTH (оба), ALL (все) и PICK (выборочный);
- 4) режимы функциональный FN, параметрический P, подпрограммный SBR, одновременный SIM

Наиболее используемыми являются первый и второй режим:

TRANSFER ,B – *безусловная передача*;

A – не используется;

B – имя блока, к которому направляется активный транзакт.

TRANSFER A,[B],C – *статистическая передача*;

A – частота (вероятность) передачи транзакта в блок C;

B – имя блока B; [по умолчанию – следующий по порядку блок];

C – имя блока C.

***Организация циклов:***

**LOOP A,B** (Зациклить)

Модифицирует параметр и управляет движением активного транзакта по модифицированному параметру.

***Перемещение транзакта к новому блоку:***

**DISPLACE A, B, [C], [D]**

Находит любой транзакт и перемещает его в любой блок с сохранением интервала модельного времени.

**8. Изменение значений параметров транзактов.**

***Задание значений параметров:***

ASSIGN A, B, [C]

***Вычисление выражения и запись его в параметр:***

PLUS A, [B]

***Изменение значения параметра транзакта:***

INDEX A,B

***Занесение значения абсолютного модельного времени:***

MARK [A]

## 9. Установка устройства в готовность и неготовность.

### *Перевод устройства в состояние неготовности:*

FUNAVAIL A,B,C,D,E,F,G,H

A – имя или номер устройства;

B–H – необязательные параметры, задающие режимы обработки транзактов и метки блоков для перенаправления транзактов.

### *Перевод устройства в состояние готовности:*

FAVAIL A

Рассмотрим применение основных блоков GPSS на примере.

**Пример.** В систему с одноканальным устройством обслуживания поступают заявки. Интервал времени поступления заявок распределен по экспоненциальному закону  $f(x) = \frac{1}{s} \cdot e^{-(x-m)/s}$ ,  $x \geq m$  с параметрами  $m = 0$  с,  $s = 1$  с. Время обслуживания также распределено по экспоненте с параметрами  $m = 0$  с,  $s = 0,8$  с. Необходимо промоделировать функционирование системы в течение 10 часов (36000 с). Модель должна обеспечить сбор статистики об очереди.

\*\*\*\*\*

; Задание таблицы

VrObr TABLEM1, 800,50,10 ; Определение таблицы

; Моделирование процесса обслуживания заявок

GENERATE (Exponential(1,0,1)) ; Генерация заявок

QUEUE Ocher ; Постановка в очередь

SEIZE Ustr ; Попытка занять устройство

DEPART Ocher ; При успехе – окончание ожидания

ADVANCE (Exponential(2,0,0.8)) ; Задержка на время обслуживания

RELEASE Ustr ; Освобождение устройства

TERMINATE ; Конец истории транзакта

; Сегмент задания времени моделирования

GENERATE 36000,,1 ; Момент окончания

TERMINATE 1 ; Конец моделирования

Комментарии могут открываться звёздочкой в начале строки или продолжать текст оператора после точки с запятой. Комментарий должен быть на латинице.

Имеется 2 сегмента – в каждом есть пара GENERATE-TERMINATE, но только один TERMINATE с операндом, вычитаемым из счётчика завершения.

### 4.4.3. Объекты вычислительной категории

К объектам вычислительной категории относят переменные, функции и ГСЧ (генераторы случайных чисел). Переменная – это обобщённое понятие, которое в конкретном случае представляет собой выражение. Выражение может включать следующие элементы:

- 1) константы;
- 2) системные числовые атрибуты (СЧА);

- 3) арифметические операции;
- 4) логические операции;
- 5) библиотечные функции;
- 6) переменные пользователя.

**Константы** – числа, представляются в одном из трёх типов данных – целочисленном, вещественном или строковом.

**СЧА** предназначены для представления информации, автоматически регистрируемой и корректируемой в процессе моделирования. По способу определения имен СЧА можно разделить на три группы:

- **СЧА объектов** (имя СЧА состоит из двух частей: *групповое имя*, идентифицирующее тип объекта и тип информации об объекте, *имя конкретного члена группы*);
- **матричные СЧА** (частный случай первой), могут содержать до трёх идентификаторов;
- **атомарные СЧА** A1, AC1, C1, M1, PR, TG1, XN1, Z1 (не требуют указания номера или имени объекта).

СЧА подразделяются на следующие типы:

- СЧА модели (RNj, C1, AC1, TG1, Z1);
- СЧА транзактов;
- СЧА блоков;
- СЧА памяти;
- СЧА устройств;
- СЧА очередей;
- СЧА ячеек и матриц ячеек;
- СЧА вычислительных объектов;
- СЧА групп, цепей и ключей;
- СЧА таблиц.

**Общими для модели являются атрибуты системного времени:**

AC1 – абсолютное системное время (таймер);

C1 – относительное системное время (с последнего RESET);

TG1 – остаток счётчика завершений.

**Арифметические, условные и логические операторы** предназначены для указания операций, которые должны выполняться над элементами выражения:

«^» – возведение в степень, «#» – умножение, «/» – деление, «\» – целочисленное деление, «@» – целый остаток, «-» – вычитание, «+» – сложение, «>=» – 'GE', «<=» – 'LE', «>» – 'G', «<» – 'L', «=» – 'E', «!=» – 'NE', «&» – 'AND', «|» – 'OR', «NOT».

**Библиотечные математические функции.**

Библиотека GPSS World имеет следующие математические функции: ABS(A), ATN(A), COS(A), EXP(A), INT(A), LOG(A), SIN(A), SQR(A), TAN(A).

### Библиотечные ГСЧ.

ГСЧ предназначены для моделирования случайных величин и случайных событий. В GPSS World встроена возможность генерации 24-х типов распределений непрерывных и случайных величин (табл. 4.6). Для обращения к вероятностному распределению необходимо указать имя оператора и его параметры, заключённые в круглые скобки и отделённые друг от друга запятой:

**<Имя процедуры>(G,A,B, ...)**

где **G** – номер генератора равномерно распределённых случайных чисел (от 1 до 999) – используется в качестве аргумента для формирования случайных величин с заданным законом распределения. Остальные параметры **A, B, ...,** задают непосредственно параметры вероятностного распределения.

Таблица 4.6

#### Моделируемые в GPSS распределения

№ п/п	Распределение	Оператор	Тип
1.	Бета-распределение	Beta(RN <sub>j</sub> ,min,max,a,b)	Вещественный
2.	Биномиальное	Binomial(RN <sub>j</sub> ,t,p)	Целый
3.	Дискретно-равномерное	DUniform(RN <sub>j</sub> ,min,max)	Целый
4.	Экспоненциальное	Exponential(RN <sub>j</sub> ,m,s)	Вещественный
5.	Экстремального значения типа А	ExtValA(RN <sub>j</sub> ,m,s)	Вещественный
6.	Экстремального значения типа В	ExtValB(RN <sub>j</sub> ,m,s)	Вещественный
7.	Геометрическое	Geometric(RN <sub>j</sub> ,p)	Целый
8.	Гамма	Gamma(RN <sub>j</sub> ,m,s,a)	Вещественный
9.	Обратное Гаусса	InvGauss(RN <sub>j</sub> ,m,s,a)	Вещественный
10.	Обратное Вейбулла	InvWeibull(RN <sub>j</sub> ,m,s,a)	Вещественный
11.	Лапласа	Laplace(RN <sub>j</sub> ,m,s)	Вещественный
12.	Логлапласово	LogLaplace(RN <sub>j</sub> ,m,s)	Вещественный
13.	Отрицательное биномиальное	NegBinom(RN <sub>j</sub> ,c,p)	Целый
14.	Логистическое	Logistic(RN <sub>j</sub> ,m,s)	Вещественный
15.	Логлогистическое	LogLogis(RN <sub>j</sub> ,m,s,a)	Вещественный
16.	Логнормальное	LogNormal(RN <sub>j</sub> ,m,s,a)	Вещественный
17.	Парето	Pareto(RN <sub>j</sub> ,m,s)	Вещественный
18.	Пирсона типа 5	Pearson5(RN <sub>j</sub> ,m,s,a)	Вещественный
19.	Нормальное	Normal(RN <sub>j</sub> ,m,s)	Вещественный
20.	Пуассона	Poisson(RN <sub>j</sub> ,v)	Целый
21.	Пирсона типа 6	Pearson6(RN <sub>j</sub> ,m,s,a,b)	Вещественный
22.	Треугольное	Triangular(RN <sub>j</sub> ,min,max,mode)	Вещественный
23.	Равномерное	Uniform(RN <sub>j</sub> ,min,max)	Вещественный
24.	Вейбулла	Weibull(RN <sub>j</sub> ,m,s,a)	Вещественный

## 4.5. Технология моделирования в среде GPSS World

### 4.5.1. Разработка моделей в среде GPSS World

Система имитационного моделирования GPSS World включает [1]:

- **язык GPSS** – высокоуровневый язык имитационного моделирования;
- **язык PLUS** (Programming Language Under Simulation) – встроенный в GPSS язык программирования низкого уровня;
- **транслятор** – программа для трансляции (перевода) с языка высокого уровня на язык компьютера, создающая объекты «Процесс моделирования».

Объектами системы GPSS World являются:

1) **«Модель»** или **«GPSS-модель»** – программа, написанная на языке GPSS и представляющая собой последовательность операторов, описывающих логику работы моделируемой системы, каждый из которых реализует некоторую конкретную функцию;

2) **«Процесс моделирования»** – непосредственно исполняемый объект, создаваемый в результате трансляции объекта «GPSS-модель»; реализация «процесса моделирования» заключается в перемещении в модели некоторых подвижных объектов, называемых *транзактами*;

3) **«Отчёт»** – создаётся автоматически по завершении процесса моделирования и содержит результаты моделирования;

4) **«Текстовый объект»** – текстовые файлы, используемые для упрощения разработки больших моделей и формирования библиотеки исходных текстов.

Три первых объекта являются основными и всегда используются при имитационном моделировании.

**Главное окно** GPSS World состоит из нескольких компонентов:

- строка заголовка;
- основное меню;
- панель инструментов;
- поле клиента для набора текста модели;
- строка состояния (разделена на три части – подсказки, сообщения об ошибках, модельное время).

Большинство действий запускается из *Главного окна* с помощью Основного меню, включающего следующие группы команд: *File* (Файл); *Edit* (Правка); *Search* (Поиск); *View* (Вид); *Command* (Команда); *Help* (Помощь); *Window* (Окно).

**Пункт меню File (Файл)** включает стандартные команды Windows-приложения: 1. *New* (Создать); 2. *Open* (Открыть); 3. *Close* (Закрыть); 4. *Save* (Сохранить); 5. *Save As* (Сохранить Как); 6. *Print* (Печать); 7. *Internet*; 8. *Recent File* (Последний файл); 9. *Exit* (Выйти).

**Пункт меню Edit (Правка)** включает команды:

- Undo (Отменить);
- Cut (Вырезать);
- Copy (Копировать);
- Paste (Вставить);
- Insert Line (Вставить строку);
- Delete Line (Удалить строку);
- Font... (Шрифт);
- Expression Window... (Окно выражения);
- Plot Window... (Окно графика);
- Insert GPSS Blocs (Вставить блоки GPSS);
- Insert Experiment (Вставить эксперимент);
- Settings (Установки).

Остановимся на **установках параметров моделирования (Edit/Settings)**.

Всего имеется 5 вкладок:

- Simulation (Моделирование) – позволяет настроить опции процесса моделирования (например, интервал опроса, размер блока параметров, глубина вычисления, размер стека, максимальная требуемая память и др.);
- Reports (Отчёты) – позволяет настроить перечень информации, включаемой по умолчанию в отчёт, генерируемый путём запуска модели;
- Random Numbers (Случайные числа) – позволяет назначить использование определённых генераторов случайных чисел;
- Function Keys (Функциональные клавиши) – позволяет назначить операции функциональным клавишам и изменить интервалы табуляции;
- Expressions (Выражения) – позволяет добавить выражения в окно Expression.

**Пункт меню Search (Поиск)** помогает перемещать курсор в тексте объекта и включает команды:

- Find/Replace (Найти/Заменить);
- Go to Line... (Перейти к строке);
- Next Bookmark (Следующая закладка);
- Mark (Установить метку);
- UnMark (Удалить метку);
- UnMark All (Удалить все метки);
- Select to Bookmark (Выделить до отметки);
- Next Error (Следующая ошибка);
- Previous Error (Предыдущая ошибка).

**Пункт меню View (Вид)** включает команды:

Notices (Заметки) – используется для управления содержимым нескольких окон;

Toolbar (Панель инструментов) – показывает или скрывает панель инструментов;

Entity Details (Детальное представление элемента);

Simulation Clock (Часы моделирования) – устанавливает или сбрасывает один из атрибутов объекта «Процесс моделирования».

**Пункт меню Command (Команда)** предназначен для создания и управления объектами «Процесс моделирования» и включает команды:

Create Simulation (Создать выполняемую модель) – запускает транслятор;

Retranslate (Перетранслировать) – предназначена для повторной трансляции;

Repeat Last Command (Повторить последнюю команду);

CONDUCT (Управление) – даёт возможность проведения эксперимента;

START (Пуск) – обеспечивает запуск оттранслированной программы;

STEP1 (Шаг 1) – обеспечивает пошаговое выполнение оттранслированной программы;

HALT (Останов) – прерывает процесс моделирования;

CONTINUE (Продолжить) – обеспечивает продолжение процесса моделирования;

CLEAR (Очистить) – возвращение моделирования в первоначальное состояние;

RESET (Сброс) – сброс статистики в начальное состояние;

SHOW... (Показать) – вызывает диалоговое окно Show Command, обеспечивающее просмотр искомых параметров в окне Journal;

Custom... (Выборочно) открывает диалоговое окно Simulation Command для ввода выборочных команд управления во время моделирования.

**Пункт меню Window (Окно)** включает команды:

– Cascade (Каскад) – расположение окон каскадом;

– Tile (Мозаика) – расположение окон мозаикой;

– Simulation Window (Окно моделирования) – вызов динамических окон;

– Simulation Snapshot (Снимок моделирования) – вызов моментальных кадров (снимков), характеризующих состояние объектов.

**Пункт меню Help (Помощь)** включает команды: Help Topics (Вызов справки) и About GPSS World (О системе GPSS World).

### **Представление времени в процессе имитации**

При имитационном моделировании выделяют три представления времени: реальное, модельное (системное) и машинное время.

**Реальное время** – это время, в котором происходит функционирование моделируемой системы в реальной жизни.

**Модельное время** – это время, в котором происходит функционирование моделируемой системы при проведении имитационного моделирования на ЭВМ.



В системном времени выполняются:

- переход моделируемой системы из одного состояния в другое;
- синхронизация работы всех компонент имитационной модели;
- управление ходом имитационных экспериментов;
- параллельная реализация событий в моделируемой системе.

**Машинное время** – время, отражающее затраты времени ЭВМ на проведение имитационного моделирования.

### **Динамические объекты системы**

Динамические объекты системы – транзакты, перемещающиеся от блока к блоку. Поведение требования определяется его атрибутами, параметрами.

Каждое требование может иметь **любое число параметров**. Значение любого параметра активного требования может быть возвращено через стандартный числовой атрибут P<номер требования> или P\$<имя требования>

**Параметрами могут быть:**

- приоритет;
- метка времени;
- номер ансамбля;
- индикатор задержки;
- индикатор трассировки;
- текущий блок;
- следующий блок.

### **Цепи событий**

Динамика продвижения транзактов в модели GPSS основана на схеме событий, при этом событием считается любое изменение состояния моделируемой системы. **Цепи** (упорядоченные списки) определяют внутреннюю организацию GPSS и определяют фактическую последовательность событий в модели.

**Цепь текущих событий** (Current Events Chain – CEC) – цепь транзактов, соответствующих событиям, которые планируются к продвижению в одном или нескольких блоках в течение текущего или ближайшего модельного времени.

**Цепь будущих событий** (Future Events Chain – FEC) – цепь транзактов, соответствующих событиям, время наступления которых больше текущего времени и которые находятся в состоянии ожидания событий в будущем.

**Цепь повторений** (Retry Chain) – цепь транзактов, которые ожидают готовности канала обслуживания.

**Цепь зависания** (Pending Chain) – это цепь транзактов, ожидающих возможности прервать устройство в режиме прерывания.

**Цепь прерывания** (Interrupt Chain) – это цепь прерванных транзактов.

**Цепь задержки** (Delay Chain) – приоритетная цепь транзактов, ожидающих возможности монопольного занятия канала обслуживания.

При освобождении устройства транзакты последовательно выбираются из цепей Pending, Interrupt, Delay.

**Цепь пользователя** (User Chain) – цепь транзактов, удалённых пользователем из цепи текущих событий, которые временно не планируются к продвижению. Управление цепями пользователя осуществляется с помощью блоков LINK (включение транзакта в цепь) и UNLINK (извлечение транзакта из цепи).

При моделировании транзакты хранятся в цепях. Существует пять видов цепей:

- текущих событий (цепь транзактов, соответствующих событиям, время наступления которых меньше или равно текущему времени);
- будущих событий (цепь транзактов, соответствующих событиям, время наступления которых больше текущего времени);
- задержки устройства или памяти (цепь транзактов, ожидающих занятия устройства или памяти);
- отложенных прерываний устройства (цепь транзактов, ожидающих занятия устройства по приоритету);
- пользователя (цепь транзактов, удалённых пользователем из цепи текущих событий и помещённых в цепь временно неактивных транзактов).

#### 4.5.2. Окна GPSS

Процедура моделирования в GPSS World не ограничивается разработкой программы, её компиляцией и получением итогового отчёта. Богатый набор графических окон среды моделирования позволяет наблюдать весь процесс имитации и взаимодействовать с этим процессом.

**Итак, в пункте меню Window (Окно) имеются важные вкладки:**

- Simulation Window (Окно моделирования);
- Simulation Snapshot (Снимок моделирования).

**Графические окна GPSS World.**

Существует 10 типов динамических «**Окон моделирования**» (по числу типов объектов), которые можно открыть в любой момент процесса моделирования:

- 1) *Blocks Window* (Окно блоков) – для установки контрольных точек и отслеживания передвижения транзактов;
- 2) *Expressions Window* (Окно выражений) – для наблюдения за изменениями количества Plus-выражений;
- 3) *Facilities Window* (Окно каналов обслуживания);
- 4) *Logicswitches Window* (Окно логических переключателей);
- 5) *Matrix Window* (Окно матрицы);
- 6) *Plot Window* (Окно гистограммы) – вывод любого количества многоцветных графиков;
- 7) *Queues Window* (Окно очередей);

- 8) *Savevalues Window* (Окно сохраняемых величин);
- 9) *Storages Window* (Окно накопителей);
- 10) *Table Window* (Окно таблицы) – динамическая гистограмма для наблюдения за сбором данных, поиска выбросов и оценки сходимости.

Эти десять окон обновляются в процессе моделирования.

#### **Моментальные снимки событий.**

Кроме динамических окон в GPSS имеются окна снимков, позволяющие фиксировать мгновенное состояние.

При моделировании событие, связанное конкретным требованием, может сохраняться в одном из снимков. Эти снимки называются *цепями*.

С помощью «*снимков моделирования*» можно получить следующие кадры состояния:

- 1) снимок цепи текущих событий (CEC Snapshot);
- 2) снимок цепи будущих событий (FEC Snapshot);
- 3) снимок любого транзакта в системе (Transaction Snapshot);
- 4) снимок числовых групп (Numeric Groups Snapshot);
- 5) снимок группы транзактов (Transaction Groups Snapshot);
- 6) снимок цепей пользователей (Userchains Snapshot).

### **4.5.3. Этапы моделирования в GPSS World**

1. Работа начинается с составления программы [59]. Посредством команды **Include** можно вставлять ранее отлаженные фрагменты.

2. Компиляция организуется по команде **Command/Create Simulation**. При обнаружении ошибок создаётся их циклический список. Просмотр списка обеспечивают команды **Next Error** и **Previous Error**. Глубина стека ошибок ограничена, поэтому одного прогона исправления ошибок может быть недостаточно.

3. При успехе компиляции активизируются **интерактивные команды** и становятся доступными **окна имитации**.

4. После этого к модели можно добавить любые предложения: **блоки (кроме GENERATE)** и **описания PLUS-процедур**. Интерактивно вставленный блок – временный, и разрушается сразу после входа в него транзакта (это так называемый режим «ручного моделирования»).

5. После успешного создания исполняемой модели (**Имитации**) через меню **Command/Start** задаётся начальное значение счётчика, которое вместе с имеющим ненулевой декремент блоком **TERMINATE** определяет длительность моделирования.

6. Её можно ограничить также по счётчику сгенерированных транзактов (операнд **D** блока **GENERATE**) или по времени – с помощью отдельного сегмента модели.

7. Команда **START** определяет окончание моделирования: её счётчик должен уменьшиться до нуля.

8. Начатое моделирование можно прервать командой **HALT**, кнопка которой имеется во всех динамических окнах. После её выполнения можно выбирать блоки и задавать контрольные остановки в них (кнопка красного светофора), удалять заданные остановки (перечёркнутый красный светофор).

9. Один шаг модели (вход активного транзакта в следующий блок) выполняется по нажатию кнопки с изображением лестницы. Продолжение моделирования с использованием команды **STEP** позволяет автоматически остановить его точно после заданного числа входов активного транзакта в блоки модели.

10. Контролировать промежуточные результаты можно через окна **Графики и Выражений**. Окно **Таблиц** позволяет наблюдать сходимость исследуемых распределений – гистограммы, среднего значения и СКО. В меню **Windows** поддерживается список всех открытых окон, включая отчёты по прогонам предыдущих версий модели.

11. Для типовых имитационных моделей собираемая по умолчанию статистика вполне достаточна. Однако существует возможность с помощью команд **TABLE** и **QTABLE** получить более подробные сведения в виде таблиц (гистограмм) распределений. Сбор данных обеспечивается блоками **TABULATE** и **DEPART** соответственно.

12. Ещё более детальные сведения можно собирать и обрабатывать с помощью скалярных и матричных сохраняемых значений и **PLUS**-процедур.

Необходимо отметить некоторые ограничения бесплатной студенческой версии, используемой в ходе обучения:

- модели ограничены 150-ю блоками **GPSS**;
- нельзя открыть модели, созданные в коммерческой версии;
- не поддерживается режим совместимости с **GPSS/PC**.

## **5. Моделирование боевых действий и систем военного назначения**

### **5.1. Теоретические основы моделирования боевых действий**

#### **5.1.1. Сущность, роль и значение моделирования боевых действий**

Моделирование играет важнейшую роль в познании окружающего мира и во всех областях практической деятельности человека. Велико значение моделирования и в военной сфере.

Эффективность и успешность боевых действий предопределяется не только соотношением сил и средств и чёткой организацией обеспечения, но и качеством принимаемых решений, умелым прогнозом развития боевой ситуации.

В повышении качества решений существенная роль отводится моделированию боевых действий и систем военного назначения. Значительна роль моделирования также в процессе обучения военных специалистов и при проведении научных исследований.

Для выработки и принятия управленческих решений любого уровня необходимо предвидеть возможные последствия их реализаций. Выбрать из нескольких альтернативных вариантов выполнения боевой задачи единственный, который в наибольшей степени соответствует складывающейся обстановке и обеспечивает достижение цели боевых действий с максимальным качеством, можно на основе оценки значений различных количественных показателей, характеризующих эти варианты. Иными словами, необходимо «измерять» варианты, определяя значения конкретных показателей качества боевых действий. При этом возникает очевидная проблема определения характеристик явлений и процессов, не произошедших ещё в действительности. Единственный путь решения данной проблемы проходит через моделирование боевых действий.

Первые математические модели боевых действий были построены ещё до революции [17]: в 1914 году русский капитан первого ранга Осипов описал дифференциальными уравнениями процесс сражения между двумя эскадрами (позже это сделал Ланчестер и с тех пор такая модель носит его имя). Класс такого рода моделей впоследствии стали называть моделями динамики средних.

Прогнозированием действий противника и своих ответных действий командиры подразделений и частей в той или иной мере занимались практически с зарождением тактики борьбы в воздухе. Но тогда это было умозрительное моделирование, которое в современных условиях уже не всегда приводит к объективным выводам из-за большого числа факторов, влияющих на характер борьбы, и невозможности оценить влияние ряда факторов в количественном измерении.

В современном понимании моделирование боевых действий стало оформляться в 1950-х годах (после войны в Корее). В США достаточно активно выполнялось моделирование боевых действий авиации в интересах принятия решений в ходе войны во Вьетнаме. И сегодня оно стало неотъемлемым элементом тактической подготовки командиров, военных специалистов [17].

Эффективность и успешность боевых действий предопределяется не только соотношением сил и средств и чёткой организацией обеспечения, но и качеством принимаемых решений, умелым прогнозом развития боевой ситуации. В повышении качества решений существенная роль отводится моделированию боевых действий. Значительна роль моделирования также и в процессе обучения военных специалистов, и при проведении научных исследований.

Моделирование предполагает выполнение различных операций с моделями. Общие вопросы теории моделирования, такие как понятие модели, классификация моделей, основные этапы моделирования, подробно рассмотрены в подразд. 1.2.

Моделирование наиболее эффективно в тех случаях, когда проводить исследования на реальных объектах практически невозможно или экономически нецелесообразно (полёты космических кораблей, работа ядерных реакторов, макроэкономические процессы, закономерности возникновения и распространения эпидемических заболеваний и т. п.). В полной мере это относится и к ведению боевых действий различного масштаба.

Моделирование боевых действий позволяет получить данные о динамике развития боевой ситуации и вероятных результатах своих действий и противника при тех или иных исходных данных, характеризующих выполнение боевой задачи. Умелое моделирование боевых действий позволяет достаточно уверенно прогнозировать характер действий противника, его возможности по противодействию и с учётом этого выбирать рациональный вариант своих действий.

Опыт войн и вооружённых конфликтов бесспорно свидетельствует о том, что наряду с соотношением сил и средств сторон исход боевых действий во многом предопределяется качеством управления. Моделирование боевых действий позволяет повысить достоверность прогнозов возможных вариантов развития боевой ситуации, результатов действий сторон, выработать рекомендации по планированию и порядку выполнения боевых задач. За счёт этого повышается обоснованность принимаемых решений, являющихся основой управления.

Кроме того, моделирование является основным инструментом, позволяющим познавать закономерности боевых действий, выявлять наиболее существенные факторы, влияющие на их результаты, определять требуемые свойства новых средств вооружённой борьбы.

Модели боевых действий описывают организованное противоборство сторон, поэтому в таких моделях обязательно учитывается противодействие про-

тивника. Другой характерной чертой моделей боевых действий является нелинейный характер влияния состава привлекаемых сил сторон и других факторов на получаемые результаты (за счёт воздействия, главным образом, элементов хитрости и обмана в решениях командиров сторон и случайного фактора в процессе боевых действий).

В зависимости от целей моделирования и многих других факторов модели боевых действий значительно отличаются друг от друга. Однако в большинстве моделей имеются некоторые общие компоненты, отражающие такие процессы, как разведка противника, его поражение (нанесение удара), перемещение сил сторон; управление силами, участвующими в боевых действиях; обеспечение боевых действий.

Имитация боевых действий в процессе моделирования осуществляется с использованием различных методов.

### **5.1.2. Классификация моделей и методы моделирования боевых действий**

Классификация моделей может быть довольно обширной и сложной. Встречаются различные варианты классификации, при которых используются различные классификационные признаки и различная трактовка схожих признаков. Ниже рассматривается вариант классификации моделей боевых действий [17], в основе которого лежат пять признаков (рис. 5.1): масштаб моделируемых боевых действий; целевая направленность моделирования; характер конечных результатов моделирования; характер учёта времени моделируемых процессов; способ построения образа оригинала боевых действий (применяемый метод моделирования).

По масштабу модели различаются стратегические, оперативные, тактические.

Существенной особенностью стратегических моделей является наличие в них большого числа логических связей. Значительный размер таких моделей не позволяет рассматривать детально боевые единицы, вследствие чего в таких моделях вводится обобщённое (эталонное) боевое средство.

Оперативные модели достаточно детальны в отображении реальных процессов, однако возможности боевых средств различных видов представляются в них своими основными характеристиками. Модели этого типа в основном предназначены для решения распределительных задач (распределение сил, ресурсов, объектов и т. д.).

Тактические модели наиболее подробно отражают реальный процесс, в них учитываются, как правило, все существенные характеристики боевых средств. Такие модели используются для разработки тактики применения подразделений и частей, разработки планов действий в конкретной боевой обстановке, разработки требований к перспективной технике.

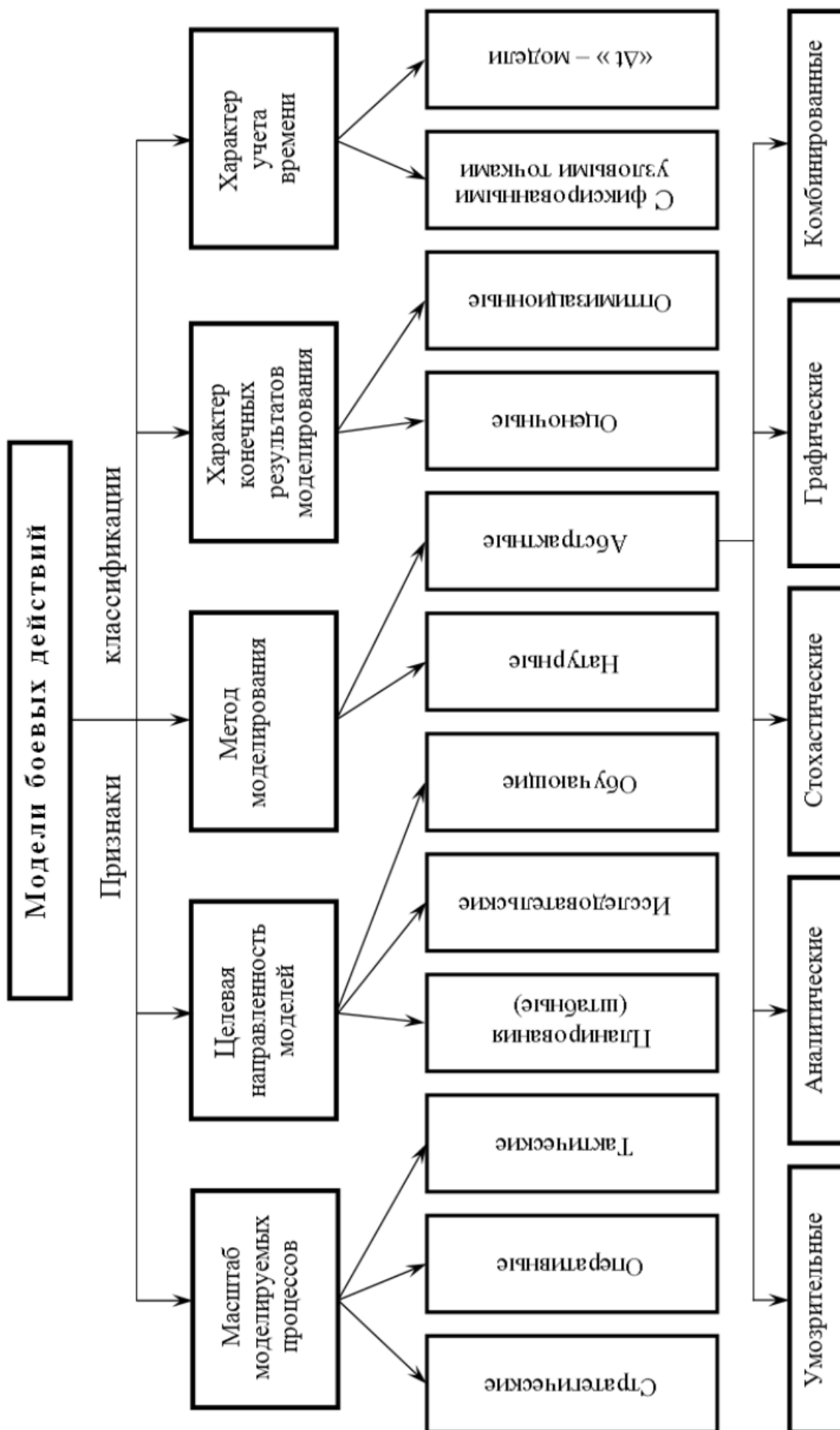


Рис. 5.1.1. Классификация моделей боевых действий



По целевой направленности модели подразделяются на модели планирования, исследовательские и обучающие.

Модели планирования – это, как правило, специализированные по конкретной задаче модели, позволяющие получить в документированном виде план действий (работ, мероприятий) в рамках заданных условий (плановая таблица полётов, план-график боевых действий). Такие модели также иногда называются штабными.

Исследовательские модели наиболее подробны в учёте всех факторов, влияющих на динамику и результат боевых действий, и позволяют анализировать характер их влияния во всём диапазоне возможных условий.

Обучающие модели отличаются от исследовательских и моделей планирования более полным банком оперативно-тактической информации и специфической организацией алгоритма, обеспечивающего выработку у обучаемых необходимых умений и навыков и приобретение ими новых знаний.

В зависимости от характера конечных результатов моделирования все модели можно подразделить на две группы – оценочные и оптимизационные.

В оценочных моделях для заданного варианта действий определяется их результат (потери сторон), т. е. в таких моделях имитируется (проигрывается) процесс боевых действий до получения конечного результата.

Оптимизационные модели построены таким образом, что в качестве результата выдают искомый (оптимальный) вариант действий, наивыгоднейший параметр или структуру системы, которые являются наилучшими в определённом диапазоне условий и при выбранных ограничениях и критериях. Оптимизационные модели сложнее в разработке, в них необходимо закладывать алгоритм и критерии, по которым осуществляется оптимизация.

По характеру учёта времени модели принято подразделять на модели с узловыми точками и  $\Delta t$ -модели.

В моделях с узловыми точками описание моделируемого процесса (состояния и взаимодействия его элементов) даётся только на характерные моменты времени (например, в процессе отражения налёта авиации противника такими узловыми точками могут быть моменты обнаружения воздушной цели, принятия решения на её уничтожение, пуска ракет и поражения цели).

В  $\Delta t$ -моделях состояние всех объектов, динамика процесса описываются через равные промежутки времени  $\Delta t$ .

Метод моделирования определяется способом формирования образа боевых действий. В зависимости от применяемых методов моделирования модели подразделяются на две группы:

- натурные модели;
- абстрактные модели.

Натурные модели основаны на использовании физических аналогов реальных объектов и явлений, характерных для процесса боевых действий. Примерами таких моделей являются лётно-тактические и исследовательские учения, полёты на боевое применение средств поражения по аналогам типовых объектов. Натурное моделирование связано со значительными затратами материальных средств.

В абстрактных моделях используются условные средства отображения образов реальных объектов и явлений. В моделях этой группы могут применяться следующие методы моделирования: умозрительный, аналитический, стохастический, графический, комбинированный.

При умозрительном моделировании образы, связанные с процессом и результатами боевых действий, формируются в сознании человека. Это самый упрощённый метод, хотя и один из самых распространённых. Осознанно или неосознанно метод неизбежно применяется всеми, кто связан с моделированием.

Аналитическое моделирование состоит в исследовании выбранных процессов с помощью совокупности математических соотношений (формул, уравнений, неравенств, логических условий, операторов и т. д.), описывающих динамику и ожидаемые результаты боевых действий. Эта совокупность соотношений объединяется в единое целое общим алгоритмом, определяющим последовательность моделирования.

Математический аппарат, используемый в аналитических моделях, может строиться на алгебраических, трансцендентных, дифференциальных уравнениях и их системах.

Стохастическое моделирование предполагает изучение возможных исходов случайных событий методом статистических испытаний. Для этого принимается гипотеза о характере закона распределения, которому подчиняется исследуемая случайная величина.

Графическое моделирование воспроизводит некоторые объекты и процессы в виде схем на картах, листах бумаги, на планшетах, дисплеях и т. п. Этот метод часто применяется при отображении основных элементов оперативно-тактической обстановки, при построении траекторий движения летательных аппаратов применительно к различным этапам боевого полёта в процессе ведения воздушного боя, нанесения авиационного удара и выполнения других задач.

Комбинированное моделирование выполняется сочетанием нескольких различных методов, в результате чего возможно, к примеру, графоаналитическое, натурно-аналитическое, аналитико-стохастическое моделирование. Сочетание нескольких методов моделирования применяется в тех случаях, когда отдельные фрагменты общего процесса боевых действий удобнее представлять одним методом, а другие – иными.

Детализация этих процессов в модели зависит от её предназначения. В некоторых случаях названные процессы могут быть представлены специальными подмоделями.

### **5.1.3. Основные цели и задачи моделирования боевых действий.**

#### **Общий порядок моделирования**

Моделирование боевых действий подразделений и частей может осуществляться по трём целевым направлениям:

- в интересах успешного выполнения боевой задачи;
- для обучения личного состава;
- в интересах научных исследований.

Каждому направлению соответствует определённый набор возможных целей моделирования. Перечислим основные цели моделирования боевых действий подразделений и частей.

#### **1. В интересах успешного выполнения боевой задачи:**

- оценка качества различных вариантов выполнения боевой задачи и выбор оптимального варианта;
- определение путей повышения эффективности выполнения поставленной боевой задачи;
- разработка способов и тактических приёмов выполнения боевой задачи.

#### **2. При обучении личного состава:**

- повышение уровня тактического мышления личного состава;
- формирование навыков выполнения расчётов;
- создание базы типовых вариантов выполнения боевой задачи (для последующего принятия решения по аналогии и обострения интуиции).

#### **3. При выполнении научных исследований:**

- исследование закономерностей боевых действий в новых условиях;
- прогнозирование характера боевых действий в новых условиях;
- разработка требований к новой боевой технике.

В интересах успешного выполнения боевой задачи в общем случае могут решаться три основные задачи моделирования (табл. 5.1).

Первая – так называемая «прямая» задача. Она предполагает определение значений основных показателей, характеризующих качество рассматриваемого варианта боевых действий. Исходными данными для моделирования являются выделяемые силы и средства, цель действий и порядок её достижения.

Вторая – так называемая «центральная» задача – предполагает определение такого варианта боевых действий из рассматриваемых, при котором достигается экстремальное значение показателя, выбранного в качестве критерия оптимальности. При этом для всех вариантов неизменными являются привлекаемые к боевым действиям силы и средства.

Третья – так называемая «обратная» задача. При её решении осуществляется определение потребных сил и средств для достижения заданного результата (заданных значений иных показателей качества боевых действий) при выбранном варианте выполнения боевой задачи.

*Таблица 5.1*

**Основные задачи моделирования боевых действий**

Задача моделирования	Исходные данные	Варьируется	Определяется
Прямая	– привлекаемые силы и средства; – цель и вариант боевых действий	–	значения показателей качества боевых действий (результат, потери, эффективность и т. п.)
Центральная	– привлекаемые силы и средства; – показатель качества боевых действий, выбранный в качестве критерия	возможные варианты боевых действий	вариант боевых действий, соответствующий экстремальному (заданному) значению выбранного критерия
Обратная	– заданный результат (заданные значения иных показателей); – вариант боевых действий	–	потребный состав сил и средств

Содержание моделирования определяется его целями, а также тем, какой процесс моделируется и какой метод моделирования используется. Однако независимо от моделируемого процесса и применяемого метода в моделировании боевых действий можно выделить ряд общих для всех случаев этапов:

- 1) подготовка исходных данных, выбор методов моделирования;
- 2) формирование вариантов действий;
- 3) преобразование исходных данных в результирующую информацию и определение возможного развития динамики боевых действий по разработанным вариантам;
- 4) оценка качества вариантов действий, анализ результатов моделирования и выбор оптимального варианта.

Все исходные данные можно разбить на три группы: о противнике, о своих силах, об условиях боевых действий.

Данные о противнике и своих силах содержат, как правило, информацию о составе, дислокации, боевых возможностях и тактике применения сил. В зависимости от уровня моделируемого процесса эта информация может иметь различную детализацию. Чем ниже уровень моделирования, тем конкретнее и детальнее должна быть информация.

Данные об условиях боевых действий содержат информацию о метеообстановке, времени суток, радиационной обстановке, рельефе местности и другую информацию, характеризующую возможное влияние ряда факторов на боевые действия.

Содержание и методика работы по формированию возможных вариантов, моделированию динамики действий по этим вариантам определяются характером боевой задачи. Эти вопросы будут подробно рассмотрены в последующих темах нашей дисциплины.

Важным и ответственным этапом моделирования является оценка качества вариантов боевых действий и выбор оптимального варианта. Необходимо правильно выбрать (сформировать) критерии для этой оценки. Для этого могут использоваться показатели, характеризующие качество вариантов выполнения боевой задачи (результат, результативность, успешность, эффективность и др.). Любой из этих показателей может быть выбран в качестве критерия оптимальности моделируемых вариантов.

В общем случае число критериев, выбранных для оптимизации варианта, может быть произвольным. В простейшем случае выбор оптимального варианта может осуществляться по одному критерию. В этом случае оптимальным считается тот вариант, которому соответствует экстремальное значение выбранного критерия оптимальности.

Например, если в качестве единственного критерия оптимальности выбирается количество (математическое ожидание числа) уничтожаемых объектов противника, то вариант выполнения задачи, при котором объектов уничтожается больше всего, и будет оптимальным.

Однако однокритериальный подход к выбору варианта боевых действий не обеспечивает в большинстве случаев требуемую обоснованность принимаемого решения. Системный анализ предполагает необходимость учёта нескольких, как правило, противоречивых факторов, влияющих на качество рассматриваемых вариантов боевых действий. Поэтому предпочтительнее сравнивать варианты по нескольким критериям, важным для оценки каждого из них. Объединить различные критерии в один общий не всегда представляется возможным. Например, при выборе варианта выполнения какой-либо задачи придётся учитывать и количество поражаемых объектов противника, и время, затрачиваемое на достижение результата, и величину своих потерь, и другие показатели.

В подобных случаях приходится прибегать к многокритериальной оптимизации. Существует целый ряд математических методов многокритериальной оптимизации. Ряд из них направлен на поиск компромисса, некоторые другие – на свёртку критериев в один интегральный критерий оптимальности. Эти методы описаны в специальной литературе по прикладной математике. На практике при моделировании в интересах выполнения боевой задачи можно использовать несложные методы, основанные на выборке двух наиболее важных критериев, переводе ряда критериев в ограничения, учёте относительной важности критериев.

Метод оптимизации по двум наиболее важным критериям предполагает выполнение ранжировки всех отобранных критериев по важности. Выбираются два наиболее важных критерия. Применительно к одному из них определяется область вариантов, в которой значения данного критерия отличаются от экстремального на величину, не превышающую 5–10%. Далее из этой области выбирается тот вариант, которому соответствует экстремальное значение второго критерия. Сущность этого метода поясняет рис. 5.2.

Допустим, в результате ранжировки наибольшая важность была присвоена эффективности выполнения боевой задачи. Вторым по важности стал достигаемый результат. В допустимую область, в пределах которой эффективность не опускается ниже 90 % от максимальной, попали варианты № 4, 5, 6 и 7. Из этих четырёх вариантов вариант № 6 позволяет достичь максимального результата. Следовательно, из девяти вариантов оптимальным является шестой.

Сущность метода перевода критериев в ограничения заключается в следующем. Из всех критериев выбирается один наиболее важный. Остальные критерии переводятся в ограничения. Далее из всех вариантов выбираются те, при которых значения указанных критериев удовлетворяют заданным ограничениям. И уже из этой группы выбирается вариант, которому соответствует экстремальное значение наиболее важного критерия. Иллюстрация данного метода приведена в табл. 5.2.

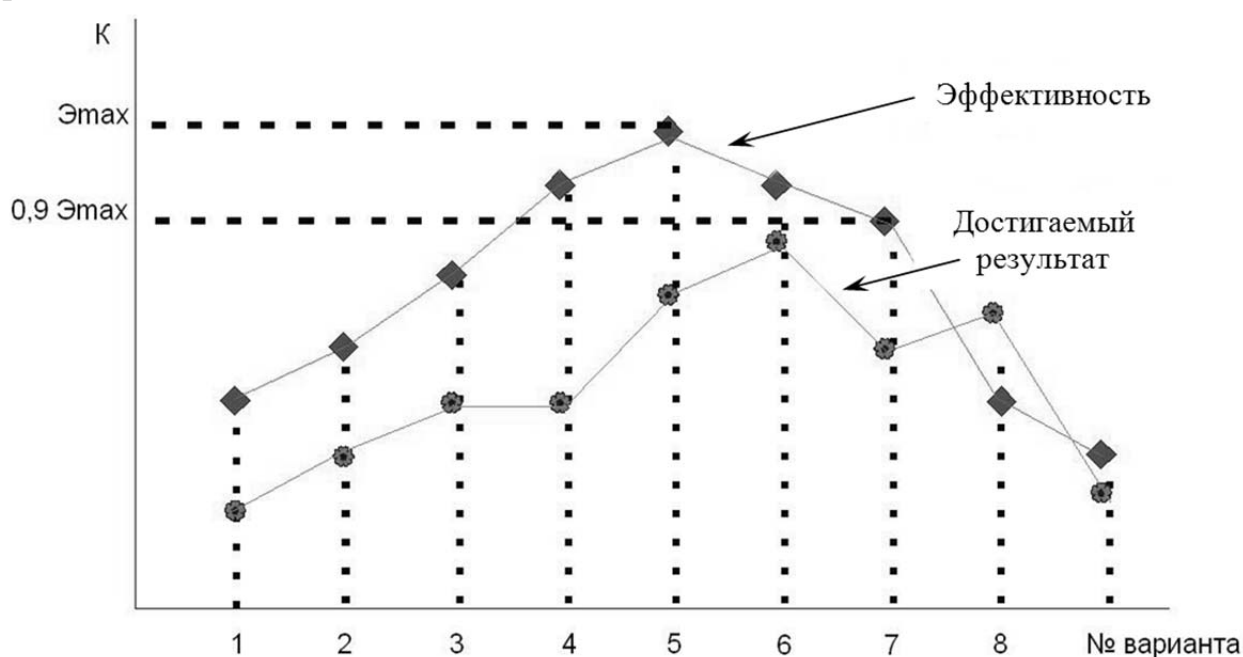


Рис. 5.2. Оптимизация по двум наиболее важным критериям

Наиболее важным критерием в данном примере выбрана эффективность выполнения боевой задачи. Два других критерия переведены в ограничения:

- относительный ущерб, наносимый противнику, должен быть  $\geq 0,5$ ;
- свои потери не должны превышать двух самолетов.

Этим ограничениям удовлетворяют только варианты № 3 и 6, но при третьем варианте достигается большая, по сравнению с шестым, эффективность действий. Следовательно, из рассмотренных шести вариантов оптимальным является третий.

Таблица 5.2

**Решение задачи методом перевода критериев в ограничения**

№ варианта	1	2	3	4	5	6
Эффективность	0,6	0,8	0,75	0,65	0,9	0,67
Относительный ущерб	0,47	0,55	0,52	0,63	0,6	0,52
Свои потери	1	4	2	3	3	1

В сущности, относительная важность критериев лежит в основе нескольких методов многокритериальной оптимизации. Ниже рассматривается один из случаев, при котором выполняется свёртка критериев и вычисляются значения интегрального критерия оптимальности рассматриваемых вариантов. Для этого используется формула

$$K_{\text{опт}_j} = \sum_{i=1}^{n_{\text{кр}\uparrow}} \gamma_i \frac{K_{ij}}{K_{i\max}} + \sum_{i=n_{\text{кр}\uparrow}+1}^{n_{\text{кр}\downarrow}} \gamma_i \frac{K_{j\min}}{K_{ij}}, \quad (5.1)$$

где  $K_{\text{опт}_j}$  – значение интегрального критерия оптимальности, вычисляемого для  $j$ -го варианта;

$\gamma_i$  – относительная важность  $i$ -го критерия;

$K_{ij}$  – значение  $i$ -го критерия для  $j$ -го варианта;

$n_{\text{кр}\uparrow}$  – количество критериев, положительные приращения которых улучшают вариант;

$n_{\text{кр}\downarrow}$  – количество критериев, которые улучшают вариант при отрицательных приращениях;

$K_{i\max}, K_{i\min}$  – максимальное и минимальное значение  $i$ -го критерия, которое он может принимать при всех вариантах.

Пример многокритериальной оптимизации, выполненный данным методом, приведён в табл. 5.3.

Таблица 5.3

**Решение задачи методом многокритериальной оптимизации**

Критерии	Относительная важность	Варианты		
		1	2	3
Эффективность	0,25	0,5	0,4	0,3
Уничтожено объектов противника	0,4	20	25	29
Свои потери	0,35	10	9	9
Интегральный критерий оптимальности		0,84	0,895	0,9

В данном примере для выбора оптимального варианта используются три критерия: эффективность выполнения боевой задачи, математические ожидания числа уничтоженных объектов противника и числа своих потерь. В результате вычислений по формуле (5.1) получилось, что максимальному значению интегрального критерия оптимальности соответствует вариант № 3. Следовательно, из рассмотренных вариантов оптимальным является третий.

Этот метод очень удобен, однако его точность целиком зависит от обоснованности значений относительной важности применяемых критериев. Значения важности критериев могут определяться волевым решением командира (начальника), либо по результатам экспертной оценки нескольких специалистов (должностных лиц).

#### **5.1.4. Оценка точности моделей и достоверности результатов моделирования**

Основной причиной, вызывающей ошибки при моделировании боевых действий, является нарушение принципа адекватности модели реальным процессам.

Определение адекватности модели, т. е. соответствия аналога реальному процессу, – один из труднейших методологических вопросов моделирования, особенно в случаях, когда проведение натурных экспериментов невозможно или экономически нецелесообразно. Это положение в значительной степени относится к моделированию боевых действий авиации.

Для проверки адекватности модели могут использоваться различные методы.

Один из методов заключается в оценке результатов, полученных в модели при задании её параметрам предельных значений. К оценке результатов моделирования привлекаются специалисты, непосредственно связанные с работой реальной системы, эксперты в данной области.

Другой метод оценки адекватности модели состоит в проверке преобразований информации от входа к выходу.

Кроме нарушения принципа адекватности существуют ещё три группы причин, влияющих на точность моделирования:

- неточность входной информации (исходных данных);
- методические ошибки, связанные с упрощением модели и погрешностями применяемых методов;
- ошибки расчёта (при методе статистических испытаний ввиду малости выборки, при численном интегрировании – за счёт величины шага, округлений и т. д.).

Ошибки входной информации появляются из-за неточности измерительных инструментов, отсутствия достоверных данных и т. д.

Положительное число, заведомо превышающее ошибку измерения по абсолютному значению, называется предельной абсолютной погрешностью. Отно-



шение предельной погрешности к абсолютному значению измеряемой величины называется предельной относительной погрешностью.

Если методическая ошибка возникает вследствие невозможности описать влияние каких-либо факторов аналитическими зависимостями, то для контроля модели и определения или хотя бы оценки методической ошибки приходится пользоваться какими-либо другими методами (испытания физической или смешанной модели, оценка при крайних допущениях и т. д.).

Методическая ошибка возникает и в том случае, когда ряд факторов учитывается, но описывается их некими средними значениями.

Выбор количества реализаций в стохастической модели зависит от требований, предъявляемых к результатам моделирования. При применении метода статистических испытаний всегда следует проанализировать такой вопрос, как возможность упрощения модели. Это увеличивает методическую ошибку, но сокращает затраты времени на получение одной реализации и тем самым при заданном машинном времени позволяет увеличить число реализаций и, следовательно, уменьшить ошибку расчёта. Таким образом, можно найти оптимальную сложность модели, обеспечивающую минимальную величину суммарной ошибки при заданном машинном времени.

Все виды ошибок принято характеризовать среднеквадратическими отклонениями.

Суммарная среднеквадратическая ошибка в определении результирующего параметра моделирования (значения искомого критерия  $K$ ) вычисляется по следующей формуле

$$\sigma_K = \sqrt{\sum_{i=1}^n \left( \frac{dK}{dA_i} \right)^2 (\sigma A_i)^2 + (\sigma M)^2 + (\sigma p)^2},$$

где  $A_i$  – исходные данные, влияющие на величину искомого критерия;

$n$  – количество исходных данных;

$\sigma A_i$  – среднеквадратические ошибки определения исходных данных;

$\sigma M$  – среднеквадратическая методическая ошибка;

$\sigma p$  – среднеквадратическая ошибка расчёта.

Во всех случаях построения моделей следует выбирать оптимальное сочетание сложности модели (определяющей методическую ошибку) и метода расчёта (определяющего ошибку расчёта) с точностью исходных данных. Если определяющей ошибкой является ошибка из-за неточности определения исходных данных, то стремиться к уменьшению остальных не имеет смысла.

Оценка точности модели и результатов моделирования – необходимые составляющие этапов разработки модели и анализа полученных результатов.

В общем случае результат проверки считается удовлетворительным, если удаётся положительно ответить на следующие вопросы:

1. Отражает ли модель чётко и ясно известные данные и ситуации?
2. Можно ли по модели определить причины известных явлений?
3. Даёт ли изменение основных параметров моделируемого процесса логичные результаты?

Принимая решения по результатам любого моделирования, следует помнить, что достоверность полученного результата не выше достоверности модели. Кроме того, необходимо иметь в виду, что моделирование, несмотря на очевидную привлекательность, не является всесильным методом исследования, обладает определёнными противоречиями и недостатками.

Основное противоречие моделирования заключается в том, что его результаты не являются безупречными и требуют корректной трактовки на основе творческого подхода. Чаще всего абсолютные значения найденных показателей обладают определёнными погрешностями, и бездумное их использование может ввести в заблуждение. Как правило, результаты моделирования боевых действий позволяют достаточно корректно выполнить лишь сравнительную оценку различных вариантов выполнения боевой задачи. Процесс создания модели, экспериментирования с ней и анализа результатов в равной мере выступает и искусством, и наукой. Поэтому важно, чтобы исследователь или руководитель (командир), пользующийся результатами моделирования, представлял себе смысл вводимых допущений в модель, сильные и слабые стороны метода, его преимущества и недостатки.

В качестве недостатков моделирования как метода отмечают следующие:

- разработка хорошей модели часто требует больших затрат времени, наличия высококвалифицированных специалистов и обходится дорого (для создания моделей сложных систем и процессов может потребоваться не один год работы);

- в действительности создаваемые модели не в полной мере отражают реальное положение вещей, игнорирование этого может привести к получению неправильных результатов и принятию неверного решения;

- поскольку модели в принципе неточны, при моделировании процессов мы всегда получаем приближённые результаты, измерить величину ошибки, как правило, не удаётся.

После ознакомления с основными положениями, касающимися моделирования боевых действий, сформировано представление о сущности, роли и месте моделирования, выполняемом при подготовке подразделений и частей к боевым действиям, при обучении личного состава и при проведении научных исследований, а также применяемым при этом методам. Более подробно были рассмотрены основные задачи и содержание моделирования в интересах успешного выполнения боевой задачи.

## **5.2. Моделирование сложных систем военного назначения**

### **5.2.1. Имитационно-моделирующий комплекс как основа построения автоматизированных учебно-тренировочных средств военного назначения**

Современные информационные технологии широко применяются в ходе боевой подготовки как в Вооружённых Силах России, так и в иностранных армиях. Одним из традиционных направлений компьютеризации процесса подготовки военнослужащих является создание виртуальной боевой обстановки в ходе занятий.

Исторически выделилось два пути развития систем моделирования обстановки в интересах боевой подготовки: предназначенные для индивидуальной подготовки и коллективные. К первому классу относятся различные тренажёры, как индивидуальные, так и коллективные, позволяющие сократить расходы на эксплуатацию ВВТ в ходе боевой подготовки, ко второму – так называемые автоматизированные системы моделирования боевой обстановки (АСМБО), создающие единую обстановку для должностных лиц органов управлений соединений и объединений. Данное разделение обусловлено отличием в решаемых задачах. Так, если в первом случае необходимо создать динамичную детализированную обстановку для конкретного обучаемого, то во втором акцент делается на имитацию боевых возможностей подразделений, частей, соединений и объединений Вооружённых Сил, а результаты моделирования представляются в виде графиков, схем (карт), таблиц и т. д. Кроме того, если применение тренажёров ограничено вопросами боевой подготовки, то АСМБО могут применяться в боевой обстановке для поиска оптимального решения на бой (операцию), то есть должны обеспечивать быстроедействие выше, чем у систем реального времени [25].

Анализируя направления и динамику развития СМВН вооружённых сил США и объединённых вооружённых сил НАТО в последние годы, следует отметить тенденции к утверждению комплексного подхода к разработке СМВН в масштабах вооружённых сил США и объединённых вооружённых сил НАТО, хотя ранее их разработка велась по программам, определяемым видами вооружённых сил.

К базовым принципам, лежащим в основе СМВН вооружённых сил США и объединённых вооружённых сил НАТО, следует отнести:

- обеспечение экономической эффективности моделирования;
- согласованность систем моделирования с существующими АСУ войсками и оружием;
- совместимость систем моделирования различного уровня и назначения;
- использование апробированных коммерческих решений при разработке систем моделирования;
- защищённость систем моделирования от несанкционированного воздействия.

Продуктом деятельности СМВН в ходе боевой подготовки является синтезированная (виртуальная) обстановка (Synthetic Environment) с достаточной степенью точности воспроизводящая оперативную обстановку. Она представляет собой совокупность реальных и виртуальных объектов, взаимодействующих в едином представлении окружающего мира.

Основа синтезированной обстановки – компьютерные модели сил (средств) (Computer Generated Forces). Это общий термин, обозначающий моделируемые объекты: воинские формирования, образцы ВВТ, их группы и т. д. Общей чертой компьютерных моделей является реализация их поведения в соответствии со складывающейся обстановкой без непосредственного участия человека.

При проведении КШУ и военных игр применяется несколько другой вид компьютерных моделей – SAF (Semi-Automated Forces). Его отличительной чертой является возможность динамической реконфигурации человеком компьютерной модели в реальном времени, что обеспечивает более реалистичное поведение моделируемого воинского формирования. Например, на односторонних учениях в состав группы руководства может вводиться специальная группа подыгрыша, состоящая из опытных офицеров, реагирующих на ошибки обучаемых ОУ. Также для обеспечения зависимости состояния оперативной обстановки от действий подчинённых (приданных) частей и подразделений сценарий учения может изменяться в соответствии с результатами выполнения последними практических упражнений, например, стрельб.

К основным направлениям совершенствования систем моделирования в интересах боевой подготовки вооружённых сил США и объединённых вооружённых сил НАТО относятся:

1. Приоритетное развитие систем моделирования с распределённым преобразованием информации. В рамках данного направления реализуются как тренажёры, так и АСМБО – от образца ВВТ (танка, истребителя, БМП) до системы подготовки ОУ объединения. Экспертами отмечается переход на качественно новый уровень боевой подготовки с применением подобных систем. Например, в печати сообщалось о создании компьютерного тренажёра для отработки действий в составе системы противовоздушной обороны, позволяющего включать в виртуальное сражение «много-на-много» произвольное число боевых единиц. Это, в частности, позволяет организовать подготовку боевых расчётов ПЗРК в составе подразделения (части, соединения).

2. Оперативно-техническое сопряжение СМВН с действующими и перспективными АСУ войсками и оружием. Существенный недостаток первых образцов АСМБО заключался в необходимости создания для обучающихся должностных лиц ОУ специализированных АРМ, размещённых в едином узле. Так, ранее основным местом проведения компьютерных КШУ был компьютерный центр в г. Рамштайн (Германия). Это приводило, во-первых, к увеличению чис-

ленности персонала, обслуживающего мероприятие оперативной подготовки, во-вторых, требовался ресурс времени, необходимый на освоение ими интерфейса программных приложений.

В сочетании с первым направлением задействование в оперативной подготовке существующих АСУ войсками и оружием позволяет устранить недостатки и увеличить число участников учения.

3. Повышение реалистичности создаваемой условной обстановки. В рамках данного направления обеспечивается повышение детализации моделируемой обстановки (концепция Train the Way You Fight). Детализация моделей определяется количеством учитываемых факторов. Следствием этого является усложнение отдельных элементов агрегативных моделей. Эффективным способом сокращения ресурсозатрат на моделирование обстановки, реализуемым в СМВН вооружённых сил США и объединённых вооружённых силы НАТО, считается накопление и повторное использование программных компонентов. На этой основе разрабатываются сложные системы, обеспечивающие согласованное по пространству и времени моделирование действий разнородных сил и средств группировки войск.

Как известно, боевая подготовка (рис. 5.3) – это не только организованный процесс воинского обучения и воспитания личного состава, слаживания подразделений, воинских частей и соединений, но и подготовка и слаживание органов управления (штабов) [35].

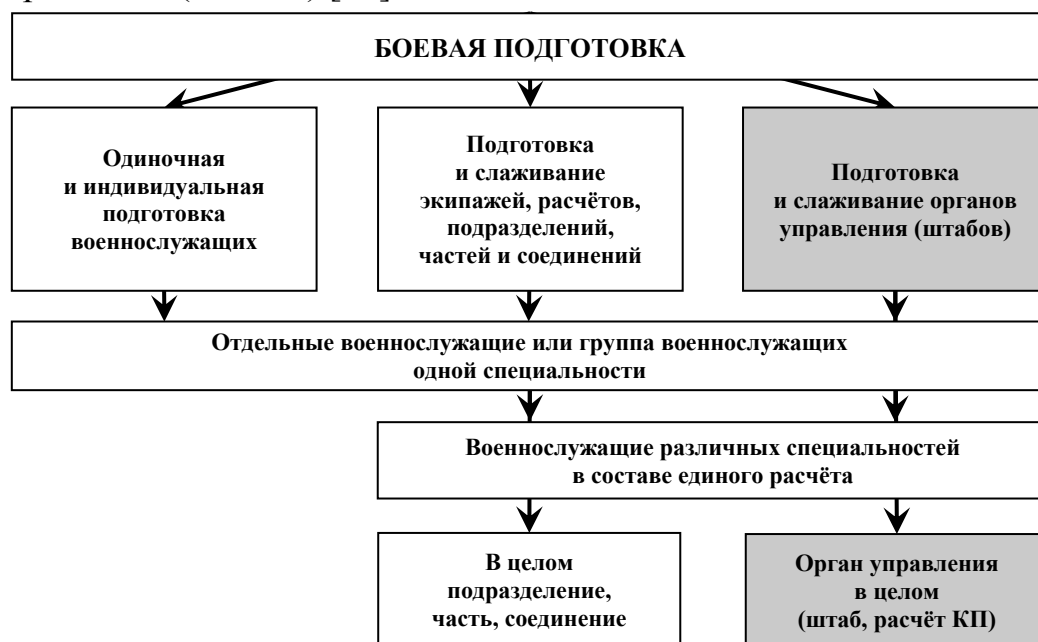


Рис. 5.3. Структура боевой подготовки по объектам обучающих воздействий

В настоящее время в ВКС идёт процесс разработки и внедрения в учебный процесс новых форм и способов обучения на базе современных компьютерных систем и тренажёров. Создаваемые УТК ПВС позволяют автоматизировать процесс обучения и значительно повысить его эффективность по первым двум

направлениям боевой подготовки (рис. 5.3). Однако при внедрении в процесс подготовки новых ТСО незаслуженно обходится вниманием необходимость автоматизации процесса подготовки и слаживания органов управления (штабов).

Постоянная готовность органов управления (штабов) к управлению частями (соединениями) в повседневной деятельности и при выполнении задач по боевому предназначению достигается высокой и всесторонней их подготовкой.

Подготовка органов управления (штабов) – это обучение их эффективному выполнению задач управления подчинёнными частями (соединениями) в мирное и военное время [41].

Таким образом, проблема создания и внедрения в учебный процесс современных ТСО, использующих информационные технологии, позволяющих автоматизировать процесс боевой подготовки командиров и штабов, является актуальной и требует пристального внимания как разработчиков подобных систем, так и заказчика – Министерства обороны РФ.

Анализ существующей системы подготовки командиров и штабов как в вузе, так и в частях (соединениях) выявил существенные недостатки технического оснащения учебно-материальной базы:

1. В настоящее время не существует единой АОС, в которой реализованы основные формы обучения органов управления (штабов) по планированию, подготовке войск (сил) и управлению ими при ведении боевых действий частей (соединений).

2. Мероприятия по боевой подготовке органов управления (штабов) проводятся, как правило, на действующих командных пунктах (запасной командный пункт) с использованием режимов тренажа систем АСУ, стоящих на вооружении, что не является в полной мере автоматизацией процесса обучения.

3. Применяемые ТСО не являются комплексами, а представляют собой несвязанные составные части – УТК, автоматизированные системы контроля подготовленности и т. п.

4. Специализированные ТСО разрабатываются производителями, не имеющими достаточных знаний в предметной области и не участвующими в организации учебного и воспитательного процесса и, соответственно, не имеющими возможности создать и апробировать полноценную АОС специального назначения.

5. Коммерческие ТСО нового поколения имеют известные сложности при внедрении и эксплуатации из-за закрытости своей архитектуры.

В результате научных исследований определены подходы к формированию методологических основ непрерывной подготовки командиров и штабов частей (соединения) с использованием инновационных технологий интерактивного обучения. Результатом явилась концепция построения автоматизированного комплекса учебно-тренировочного командного пункта, а также основные принципы и требования к его разработке.

Военные игры, как и КШУ, носят комплексный характер и проводятся по важнейшим направлениям оперативной, оперативно-тактической и тактической подготовки с целью дать обучающимся практику в исполнении конкретных должностей при планировании, организации и ведении боевых действий и их всестороннем обеспечении как индивидуально, так и в составе органа управления.

Для начала рассмотрим разработанную модифицированную модель двухсторонней военной игры (рис. 5.4), являющуюся результатом интеграции системы моделирования в общепринятую структуру обобщённой модели военной игры.

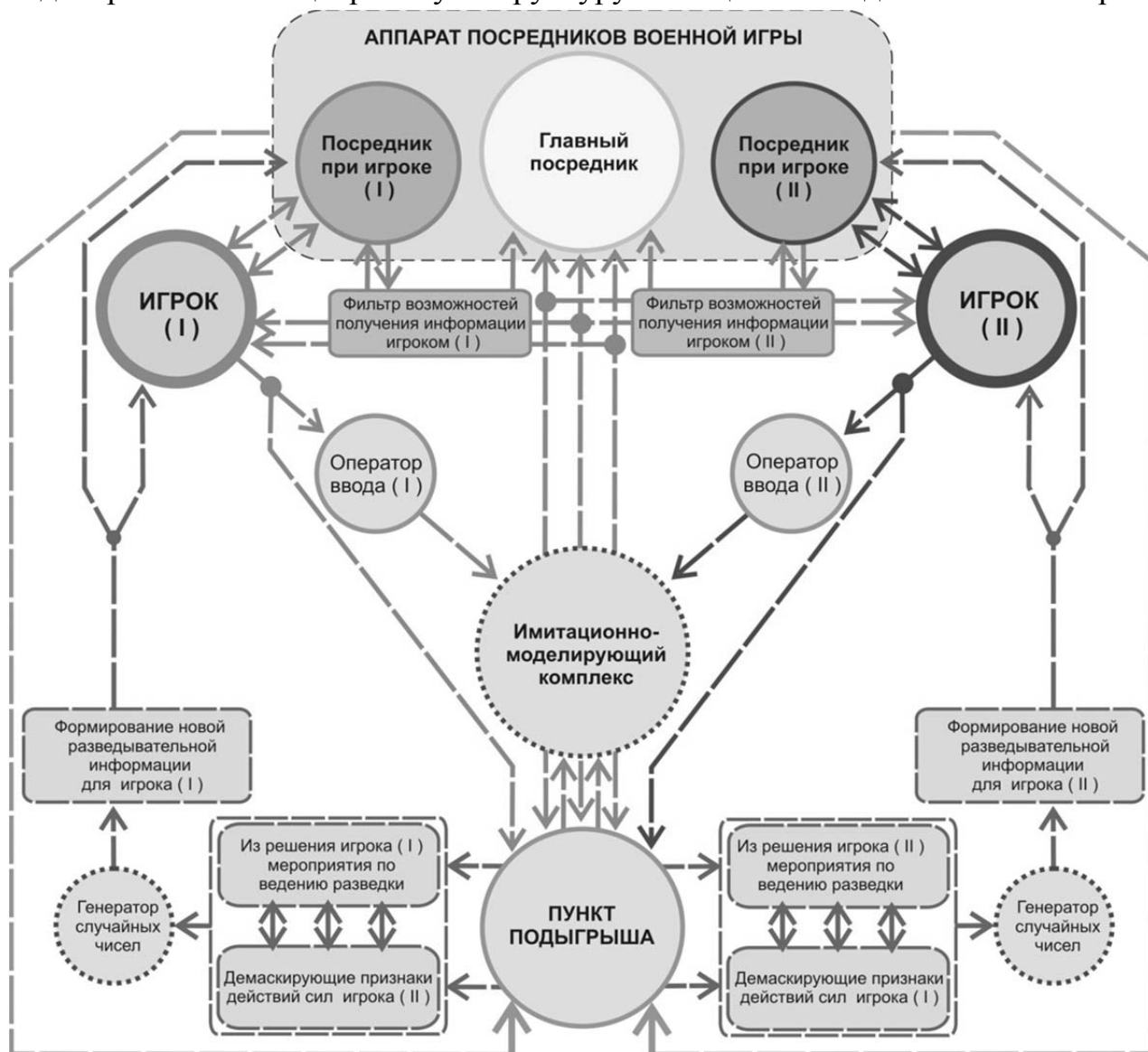


Рис. 5.4. Модифицированная модель двухсторонней военной игры

На данной схеме центральное место занимает ИМК, позволяющий:

- задавать динамику военной игры в целом;
- осуществлять поддержку принятия решения обучаемых лиц боевого расчёта;
- имитировать действия вероятного противника в режиме односторонней игры;
- устранить субъективный фактор в оценке действий лиц боевого расчёта и результатов работы обучаемых органов управления со стороны организаторов военной игры (посредников).

Дополнительными элементами, обеспечивающими успешную интеграцию ИМК, являются операторы ввода стороны I и II, осуществляющие корректный ввод командной информации соответствующих игроков в единое информационное пространство ИМК, а также блоки весовой обработки и генерации новой разведывательной информации (генераторы случайных чисел), позволяющие генерировать поступающую разведывательную информацию сторонам, адекватную проведённым ими мероприятиям по ведению разведки и степени маскировки объектов противника.

На основе анализа разработанной модифицированной модели двухсторонней военной игры и процессов, протекающих при её функционировании, были вырабатаны предназначение, принципы построения, структура и основные требования к разработке АК УТКП, который позволит проводить практическую подготовку органов управления на качественно новом учебном и методическом уровне.

АК УТКП предназначен: для автоматизированного обучения органов управления (штабов) по планированию, подготовке войск (сил) и управлению ими при ведении боевых действий частей (соединений) на базе учебно-тренировочного командного пункта.

Основные принципы разработки АК УТКП:

- вуз участвует в создании АК УТКП на всех этапах: формирование технического задания, методическое сопровождение, испытание и оценка эффективности применения;
- АК УТКП является «сквозным» средством непрерывной подготовки в системе «вуз – части (соединения)» для всех уровней подготовки;
- программное обеспечение, содержание которого не является исключительной прерогативой Поставщика, строится по открытой архитектуре;
- Заказчик имеет право формировать «закрытые» для Поставщика разделы;
- вуз формирует необходимые учебные курсы, интегрируемые в общую структуру АК УТКП самостоятельно;
- архитектура АК УТКП обеспечивает преемственность материала и гибкое его редактирование в интересах Заказчика.

Приведённая на рис. 5.5 структура АК УТКП наглядно иллюстрирует типичную АОС [68], имеющую, однако, ряд существенных отличительных черт.

Во-первых, обязательным условием построения АК УТКП является наличие системообразующего элемента – имитационно-моделирующего комплекса, который позволяет реализовывать на высоком уровне достоверности все основные формы обучения органов управления (штабов): командно-штабные учения и тренировки, а также предоставляет возможность использовать комплекс в научной работе вузов и учебных центров.



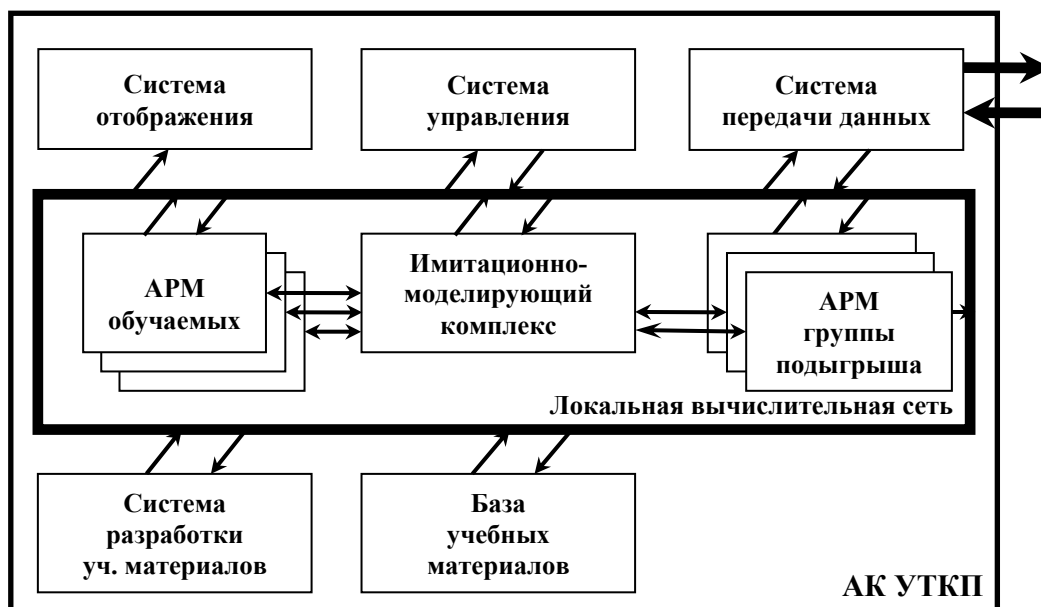


Рис. 5.5. Структура автоматизированного комплекса учебно-тренировочного командного пункта

Имитационно-моделирующий комплекс предназначен для моделирования боевых действий и оценки их эффективности на фоне противоборства разнородных и разнородных группировок войск (сил) как с учётом человеческого фактора (принятие решений лицами боевого расчёта), так и без него.

Во-вторых, наличие группы подыгрыша позволяет проводить как односторонние, так и двусторонние военные игры, а также предоставляет возможность гибкого управления учебным процессом.

В-третьих, система разработки учебных материалов, реализованная в виде интерактивного конструктора объектного моделирования [44], позволит на базе вуза разрабатывать множество учебных тактических задач и сценариев боевых действий, оттачивать их на практических и групповых занятиях с последующим внедрением в курсы боевой подготовки органов управления (штабов) частей (соединений).

И наконец, высокоскоростная система передачи данных позволяет интегрировать АК УТКП в единую многоуровневую АОС подготовки военных специалистов, структура которой показана на рис. 5.6.

Из рисунка видно, что предложенная концепция построения системы включает в себя уже разработанные и внедряемые УТК ПВС уровня подразделений, замыкающихся на АК УТКП органа управления воинской части, что позволяет проводить комплексную подготовку и слаживание экипажей, расчётов, подразделений, органов управления на уровне части. Масштабируемость предлагаемой АОС [28] даёт возможность проводить боевую подготовку различной степени сложности на уровне соединения, а при необходимости выходить и на уровень объединения (командования).

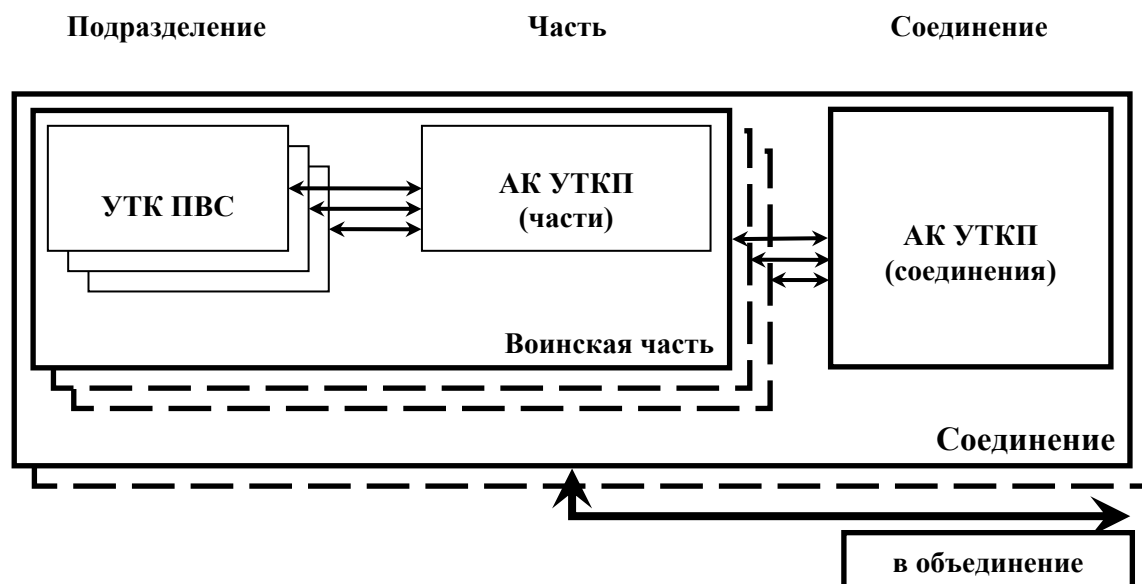


Рис. 5.6. Единая автоматизированная обучающая система подготовки военных специалистов

В ходе исследований проведён анализ размещения лиц боевого расчёта на КП соединений и объединений ВКС, влияние размещения на эффективность совместной боевой работы, а также размещение средств визуального отображения информации, оконечных устройств средств связи, оборудования АРМ. В результате обоснован и предложен вариант размещения АРМ лиц боевого расчёта и ТСО учебно-тренировочного командного пункта уровня соединения, а также АРМ группы подыгрыша и технического обеспечения работы ИМК [47].

Самостоятельным научным результатом можно считать формулирование основных положений о ИМК – функциональном ядре АК УТКП, полученных в ходе проведённых исследований. Предназначение ИМК, решаемые задачи, основные требования к комплексу, режимы его функционирования, а также требуемые результаты моделирования и формы их представления приведены в [46].

В плане расширения кругозора и углубления знаний в области применения имитационно-моделирующих комплексов при построении автоматизированных учебно-тренировочных средств военного назначения, целесообразно самостоятельно познакомиться с научными работами [46, 49].

### 5.2.2. Моделирование автоматизированных систем радиовидения воздушно-космического базирования

#### Структура модели и имитатора сигналов автоматизированной системы радиовидения

Радиовидение – наблюдение объектов в радиодиапазоне волн с детальностью оптических систем. В отличие от оптических систем, системы радиовидения дают возможность получать изображение объектов независимо от метеоусловий и естественной освещённости, на значительном удалении и одновременно

в широкой зоне обзора, а также объектов, невидимых в оптическом диапазоне волн. История развития систем радиовидения началась в 50-х годах прошлого века практически одновременно в СССР и США сначала применительно к задачам воздушной и космической разведки. Такие системы получили название РСА. Первые научные работы (монографии), обобщающие опыт научных исследований и разработки РСА, опубликованы в 1970 году [3].

В настоящее время радиовидение переживает этап бурного развития: значительно повышаются тактико-технические характеристики (разрешающая способность до 10 см, режимы селекции движущихся целей, интерферометрическая съёмка трёхмерных (рельефных) изображений, полное поляризационное зондирование и приём, многопозиционные РСА), резко расширяются области применения РСА (в том числе РСА с обратным (инверсным) синтезированием апертуры антенны для получения изображений воздушных и космических объектов). Практически все новые системы используют цифровые методы обработки сигналов, что позволяет работать в реальном масштабе времени и автоматизировать процедуры формирования и обработки РЛИ.

При проведении теоретических и экспериментальных исследований РСА широко используется метод математического моделирования таких РЛС, проводимый с помощью универсальных ЭВМ. Использование моделирования на ЭВМ позволяет решить ряд теоретических задач, существенно сократить время и стоимость проведения экспериментальных исследований по созданию и совершенствованию автоматизированных систем радиовидения, а порой провести исследования, которые невозможны на натурных макетах.

На основании опыта создания математических моделей РСА можно предложить структурную схему такой модели (рис. 5.7). Вся модель разбита на отдельные блоки, каждый из которых представлен на рисунке в виде прямоугольника. Следует сразу заметить, что реализация каждого блока может осуществляться как в форме части единой программы РСА, так и в форме самостоятельных программ (обычно подпрограмм). При проведении широкого круга исследований второй способ является более удобным, так как позволяет компоновать из подпрограмм модели РСА различной сложности при минимальных перестройках программы.

К числу основных блоков модели РСА следует отнести блоки:

- а) задания исходных данных и определения параметров РСА;
- б) моделирования траектории движения ФЦА;
- в) формирования траекторного сигнала;
- г) коррекции фазы сигнала;
- д) обработки сигнала по азимуту;
- е) обеспечения заданного вида обзора;

- ж) формирования РЛИ;
- з) оценки результатов моделирования.

Каждый из блоков состоит из отдельных частей, некоторые из них (наиболее важные) выделены на рис. 5.7.

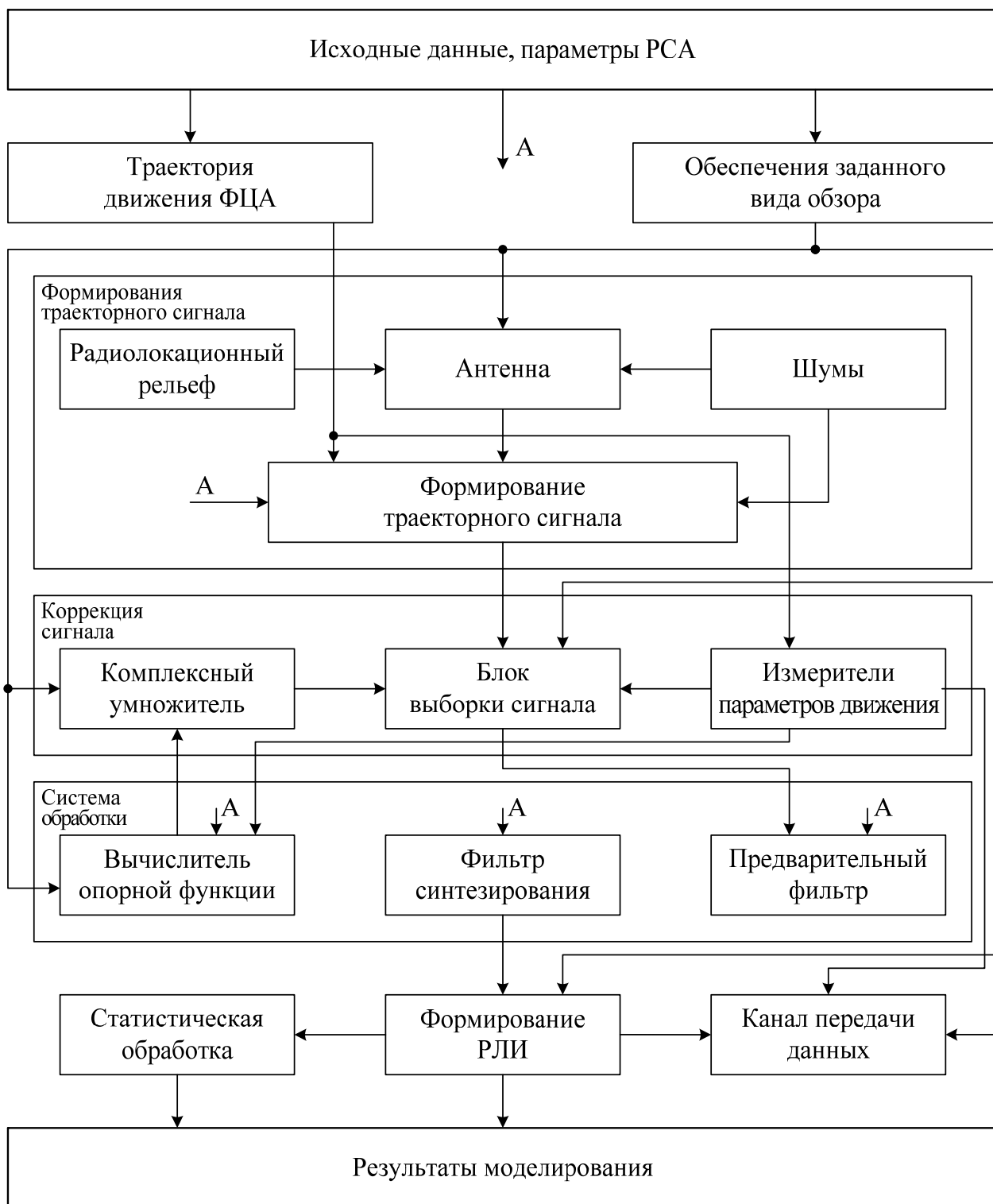


Рис. 5.7. Структура модели автоматизированной системы радиовидения с РСА

Очень важным моментом в процессе моделирования является выбор исходных данных. В общем случае в качестве исходных данных в модель вводятся:

- а) параметры движения носителя РСА (высота  $h_0$  и скорость  $V_{\text{ц}}$ );
- б) параметры движения объекта наблюдения (при обратном синтезировании апертуры антенны);
- в) вид траектории движения ФЦА;
- г) параметры радиолокационной станции (длина волны  $\lambda$ , период зондирования  $T_3$ );
- д) наклонная дальность  $r_0$  до зоны обзора;
- е) вид обзора;
- ж) ширина зоны обзора;
- з) способ обработки;
- и) размер интервала синтезирования  $L$ .

К выбору исходных данных следует относиться с большим вниманием, так как ошибки в выборе исходных данных приводят к получению неверных результатов и ставят под сомнение правильность созданной модели.

На основе исходных данных и параметров РСА рассчитывается траектория движения ФЦА. Моделирование полёта носителя РСА в реальных условиях является самостоятельной задачей, пути решения которой изложены в ряде работ [12, 55]. Однако широкий круг задач может решаться без учёта реальных условий полёта конкретного носителя РСА, что значительно упрощает моделирование. В этом случае траектория ФЦА может быть описана аналитическими зависимостями, но следует предусмотреть возможность описания как прямолинейных, так и криволинейных траекторий.

В блоке обеспечения заданного вида обзора устанавливаются законы управления реальной антенной и опорной функцией, выбирается способ обработки траекторного сигнала и вариант формирования РЛИ.

В блоке формирования траекторного сигнала на основе рассчитанной траектории ФЦА и вида обзора формируется сигнал на входе системы обработки. Местность и цели, в том числе и протяжённые (радиолокационный рельеф), создаются обычно в виде набора имитирующих точечных целей, параметры переломления и координаты которых задаются в соответствии с задачами, решаемыми при моделировании. Часто появляется необходимость создания фоновой обстановки в виде выборок шума («некогерентный фон») с определённым законом распределения. Здесь же при необходимости учитывается ДНА.

На выходе блока формирования траекторного сигнала синфазная и квадратурная составляющие, а точнее – отсчёты этого сигнала, которые после распределения по дальности и азимуту заносятся в память ЭВМ.

При реализации в модели всенаправленного обзора необходимо, чтобы отсчёты траекторного сигнала поступали в систему обработки в соответствии с правилом, определяемым на основе данных, получаемых из блока обеспечения заданного вида обзора. Эта операция осуществляется в блоке коррекции фазы сигнала. Здесь же сигнал корректируется. Коррекция включает:

- а) сдвиг спектра траекторного сигнала в область низких частот, осуществляемый за счёт нелинейного преобразования, проводимого в блоке коррекции с использованием сигнала цифрового гетеродина, настройка которого происходит на основе данных полученных из блоков ИПД и обеспечения заданного вида обзора;

- б) компенсацию траекторных нестабильностей и упругих колебаний конструкции носителя РСА, которая тоже может быть сведена к процессу гетеродирования;

- в) демодуляцию сигнала, которая сводится к умножению сигнала на опорную функцию.

При моделировании системы компенсация траекторных нестабильностей и ИПД возникает необходимость моделирования навигационных датчиков, которое может быть сведено к моделированию ошибок этих датчиков, т. е. моделированию случайных процессов с заданными законом распределения и корреляционной функцией.

Упорядоченный и скорректированный сигнал поступает в систему обработки, где осуществляется обработка траекторного сигнала. В наиболее общем случае этот блок включает:

- а) вычислитель опорной функции;

- б) предварительный фильтр;

- в) фильтр синтеза.

Полученные в результате обработки отсчёты выходного сигнала поступают в блок формирования РЛИ, где в соответствии с законом, определяемым в блоке обзора, создаётся условное РЛИ. Здесь же осуществляется некогерентное накопление изображений.

Для получения количественной оценки часто необходимо применять статистическую обработку и специальные методы оценки результатов моделирования. Результаты моделирования будут представлены в виде условных РЛИ, таблиц, графиков и т. д.

Построение такой универсальной модели позволяет осуществить практически все виды исследований, необходимых при проектировании РСА. Однако следует заметить, что при решении конкретных задач не имеет смысла создавать столь развитую модель, так как это затрудняет анализ полученных результатов и требует длительного времени для отладки модели.

## **Моделирование процессов формирования изображений в автоматизированных системах радиовидения воздушно-космического базирования в телескопическом режиме**

Современные автоматизированные системы радиовидения при решении задач мониторинга земной поверхности используют цифровые алгоритмы обработки сигналов и управления режимами работы, что существенно расширяет возможности их практического применения. Цифровое управление параметрами системы даёт возможность реализации различных видов обзора земной поверхности: переднебокового, телескопического (прожекторного), секторного и др. Любой из этих видов обзора позволяет производить многократный просмотр участка местности, представляющего интерес для оператора, однако наиболее целесообразным для этого является телескопический обзор [3].

При телескопическом обзоре РЛИ формируется в виде отдельного кадра в окрестности выбранной точки – центральной точки участка местности (рис. 5.8), положение которого неизменно ( $x_{ЦТ} = \text{const}$ ,  $y_{ЦТ} = \text{const}$ ). Размер кадра РЛИ определяется раскрывом диаграммы направленности антенны, а ось диаграммы направленности отслеживает центральную точку кадра РЛИ. При этом закон управления диаграммой направленности антенны

$$\psi_a(t) = \arctg((x_{ЦТ} - V_n t) / y_{ЦТ}),$$

где  $x_{ЦТ}, y_{ЦТ}$  – пространственные координаты центральной точки, введённые в соответствии с рис. 5.8;

$\gamma_0$  – угол между осью диаграммы направленности антенны и нормалью к земной поверхности;

$V_n$  – путевая скорость летательного аппарата;

$t$  – текущее время;

$h_0$  – высота полёта;

$L$  – длина интервала синтезирования апертуры.

В данном случае время синтезирования апертуры  $T_c = L / V_n$ , а общее время наблюдения  $T = m T_c$ , где  $m$  – количество периодов синтезирования.

Именно при телескопическом обзоре удаётся добиться условий, при которых время накопления информации значительно больше длительности синтезирования, т. е.  $T \gg T_c$ , что позволяет существенно повысить качество (разрешение) формируемого РЛИ. Принцип формирования изображения в телескопическом режиме предполагает последовательное многократное зондирование одного и того же участка местности в различных условиях (рис. 5.8). Это соответствует разбиению общего времени накопления информации  $T$  на равные отрезки, соответствующие длительности периода синтезирования апертуры  $T_c$ . Данное об-

стоятельство позволяет применить для решения задачи формирования РЛИ научно-методический аппарат оптимального восстановления случайных полей в дискретном времени [53]. При этом шагом дискретизации по времени будет являться длительность интервала синтезирования апертуры  $T_c$ . Необходимо отметить, что в данной постановке понятия «восстановление» и «формирование» РЛИ совпадают.

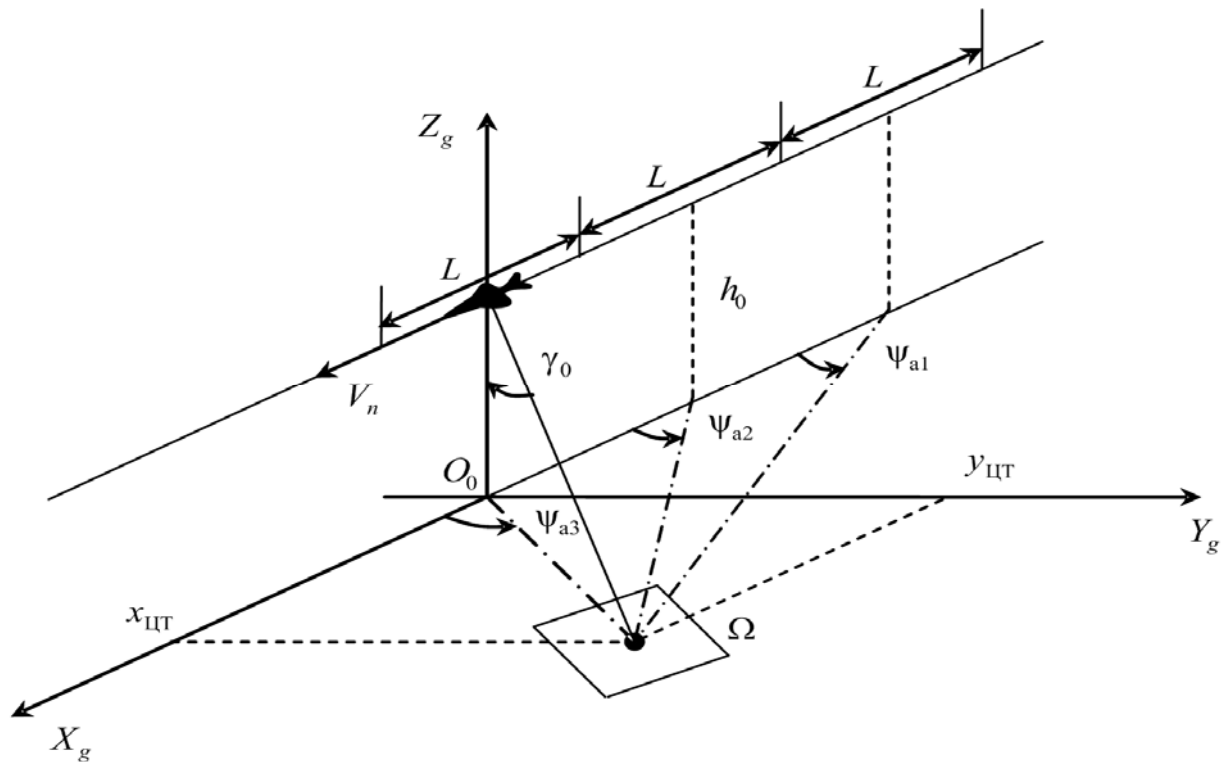


Рис. 5.8. Телескопический режим работы системы радиовидения воздушно-космического базирования

С учётом выше изложенного, сигнал на входе системы формирования РЛИ в телескопическом режиме может быть записан в виде

$$\xi_j(t) = S_j(t, u(\mathbf{r})) + n_j(t) = \int_{\Omega} F_j(t, \mathbf{r}) u(\mathbf{r}) d\mathbf{r} + n_j(t), \quad j = \overline{1, m}, \quad t \in [0, T_c]. \quad (5.2)$$

Здесь

$$u(\mathbf{r}) = u_j(\mathbf{r}) = u_{j-1}(\mathbf{r}) \quad (5.3)$$

неизменный за время накопления информации удельный коэффициент рассеяния, характеризующий неподвижное во времени РЛИ зондируемой поверхности  $\Omega$ ;

$m = \frac{T}{T_c}$  – количество этапов обработки, определяемое числом интервалов

наблюдения (периодов синтезирования апертуры) за общее время накопления информации  $T$ ;  $\mathbf{r} = [x, y]^T$  – вектор пространственных координат в области  $\Omega$ ;

$S_j(t, u(\mathbf{r}))$  – полезный сигнал на  $j$ -м интервале наблюдения, который в данном случае является линейным функционалом от пространственной реализации  $u(\mathbf{r})$ ;



$F_j(t, \mathbf{r})$  – значения зондирующего сигнала  $F(t)$  в каждой точке зондируемой поверхности  $\Omega$  на  $j$ -м интервале наблюдения;  $n_j(t)$  – гауссовский белый (как на интервале синтеза  $T_c$ , так и на всем интервале наблюдения  $T$ ) шум (ГБШ) на  $j$ -м интервале наблюдения с нулевым математическим ожиданием и корреляционной функцией  $R_{ij}(t_1, t_2) = \frac{N}{2} \delta(t_2 - t_1) \delta_{ij}$ , где  $\delta_{ij}$  – символ Кронекера.

Конкретизируя общий алгоритм оптимального восстановления случайных полей в дискретном времени [53] применительно к задаче формирования изображений в системах активного зондирования при телескопическом обзоре (5.2), (5.3) получим соответствующие уравнения для оценки

$$\hat{u}_j(\mathbf{r}) = M \{u(\mathbf{r}) | \xi_1(t) \dots \xi_m(t)\}$$

и корреляционной функции ошибки восстановления

$$R_j(\mathbf{r}, \mathbf{r}') = M \left\{ \left[ u(\mathbf{r}) - \hat{u}_j(\mathbf{r}) \right] \left[ u(\mathbf{r}') - \hat{u}_j(\mathbf{r}') \right] | \xi_1(t) \dots \xi_m(t) \right\} :$$

$$\hat{u}_j(\mathbf{r}) = \hat{u}_{j-1}(\mathbf{r}) + \frac{2}{N} \int_0^{T_c} \int_{\Omega} R_j(\mathbf{r}, \mathbf{r}') F_j(t, \mathbf{r}') \left( \xi_j(t) - \int_{\Omega} F_j(t, \mathbf{r}'') \hat{u}_{j-1}(\mathbf{r}'') d\mathbf{r}'' \right) d\mathbf{r}' dt, \quad (5.4)$$

$$R_j(\mathbf{r}, \mathbf{r}') = \left[ \frac{2}{N} \int_0^{T_c} F_j(t, \mathbf{r}) F_j(t, \mathbf{r}') dt + R_{j-1}^{-1}(\mathbf{r}, \mathbf{r}') \right]^{-1}, \quad j = \overline{1, m}. \quad (5.5)$$

Здесь  $\hat{u}_0(\mathbf{r}) = m_u(\mathbf{r})$  и  $R_0(\mathbf{r}, \mathbf{r}') = R_u(\mathbf{r}, \mathbf{r}')$  – априорные математическое ожидание и корреляционная функция модели исходного изображения  $u(\mathbf{r})$  соответственно [36].

Как видно из полученных выражений (5.4), (5.5), процедура формирования РЛИ в телескопическом режиме осуществляется рекуррентно во времени с шагом, равным интервалу синтеза  $T_c$ .

Структурная схема устройства, реализующего алгоритм (5.4), (5.5), представлена на рис. 5.9.

Основными блоками устройства обработки являются:

- блок формирования сигналов (БФС), генерирующий значения зондирующего сигнала  $F(t)$  в каждой точке зондируемой поверхности  $\Omega$  на  $j$ -м интервале наблюдения с использованием априорных сведений о зондирующем сигнале и навигационной информации;

- блок вычисления пространственно распределённой корреляционной функции ошибки восстановления  $\mathbf{R}$ , на каждом  $j$ -м этапе обработки реализующий процесс вычисления  $R_j(\mathbf{r}, \mathbf{r}')$  в соответствии с выражением (5.5);

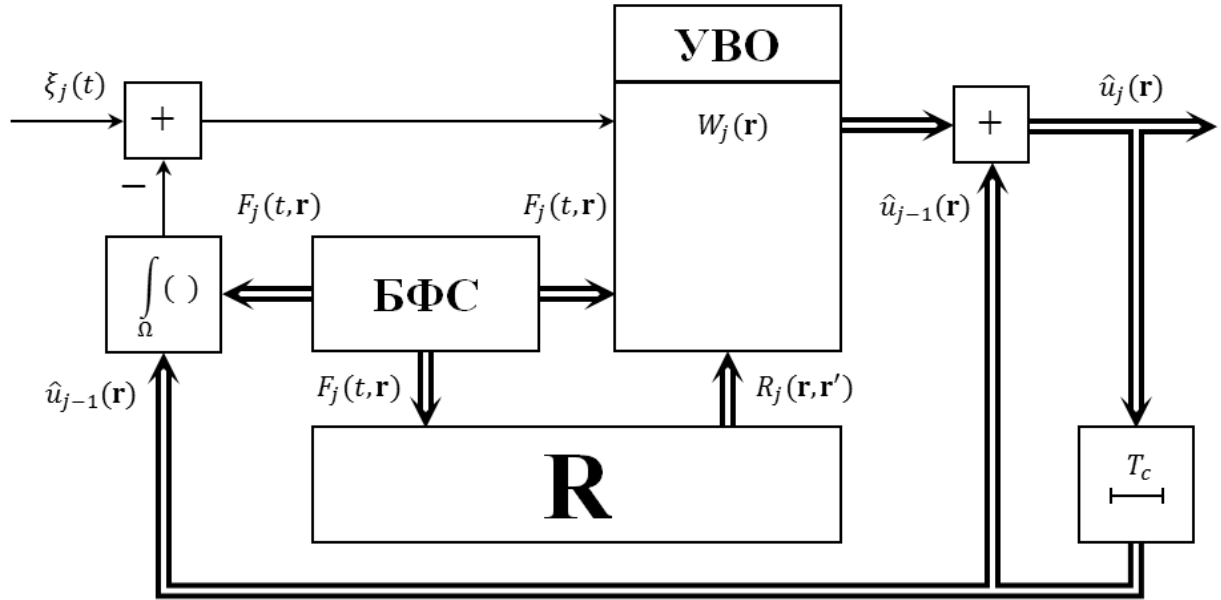


Рис. 5.9. Структурная схема устройства оптимального восстановления изображений в системах радиовидения воздушно-космического базирования при телескопическом обзоре

– устройство весовой обработки (УВО), которое на каждом  $j$ -м этапе обработки формирует корректирующую поправку  $W_j(\mathbf{r})$  для оценки случайного поля РЛИ, полученной на предыдущем этапе обработки  $\hat{u}_{j-1}(\mathbf{r})$ :

$$W_j(\mathbf{r}) = \frac{2}{N} \int_0^{T_c} \int_{\Omega} R_j(\mathbf{r}, \mathbf{r}') F_j(t, \mathbf{r}') \left( \xi_j(t) - \int_{\Omega} F_j(t, \mathbf{r}'') \hat{u}_{j-1}(\mathbf{r}'') d\mathbf{r}'' \right) d\mathbf{r}' dt. \quad (5.6)$$

Из рис. 5.9 видно, что рекуррентная структура формирования оценки предполагает на каждом  $j$ -м этапе ещё и пространственно-временную обработку принимаемого сигнала  $\xi_j(t)$  по правилу (5.6). Двойными линиями на схеме показана связь по всем значениям пространственной координаты  $\mathbf{r}$ .

В заключение следует отметить, что в качестве основного метода обработки траекторного сигнала в существующих РСА используется субоптимальный метод согласованного приёма [37], где в отличие от оптимальной процедуры восстановления предполагается получение оценки поля изображения земной поверхности в виде выходного сигнала согласованного фильтра

$$\begin{aligned} \hat{u}_{\text{с.ф}}(t, \mathbf{r}) &= \int_0^t h_{\text{с.ф}}(\tau, \mathbf{r}) \xi(\tau) d\tau = \\ &= \int_0^t \int_0^t F(\tau_2, \mathbf{r}) R_n^{-1}(\tau_1, \tau_2) \xi(\tau_1) d\tau_1 d\tau_2 = \frac{1}{N} \int_0^t F(\tau, \mathbf{r}) \xi(\tau) d\tau, \end{aligned} \quad (5.7)$$

а получаемое изображение  $\hat{u}_{\text{с.ф}}(t, \mathbf{r})$  полностью характеризуется функцией неопределённости,

$$\Psi(t, \mathbf{r}, \mathbf{r}') = \frac{2}{N} \int_0^t F(\tau, \mathbf{r}) F(\tau, \mathbf{r}') d\tau, \quad (5.8)$$

которая представляет собой (в отсутствие помех) изображение точечного отражающего объекта. Протяжённость функции неопределённости по пространственным координатам характеризует разрешающую способность системы зондирования. Так, например, размер элемента разрешения вдоль направления синтеза определяется протяжённостью раскрыва антенны в этом направлении и равен её половине, а в перпендикулярном направлении (по дальности) – обратно пропорционален ширине спектра зондирующего сигнала.

В [34] показано, что при статистической регуляризации решения задачи восстановления на основе критерия минимума среднего квадрата ошибки импульсная характеристика корректирующего фильтра будет определяться корреляционной функцией ошибки восстановления  $R_j(t, \mathbf{r}, \mathbf{r}'')$  (в телескопическом режиме определяемая выражением (5.5)), а обобщённая функция неопределённости примет вид

$$\Psi_{0j}(t, \mathbf{r}, \mathbf{r}') = \int_{\Omega} R_j(\mathbf{r}, \mathbf{r}'') \Psi(t, \mathbf{r}', \mathbf{r}'') d\mathbf{r}''. \quad (5.9)$$

Таким образом, разработанный алгоритм оптимального восстановления изображений в системах радиовидения воздушно-космического базирования в телескопическом режиме (5.4), (5.5) позволяет проводить рекуррентную обработку наблюдаемого сигнала  $\xi_j(t)$ , при которой с ростом количества этапов обработки  $j \rightarrow \infty$  происходит уменьшение протяжённости обобщённой функции неопределённости по пространственным координатам  $\Psi_{0j}(t, \mathbf{r}, \mathbf{r}')$ , что позволяет повысить разрешающую способность РЛС землеобзора [40].

Итак, подводя итог вышесказанному, следует отметить, что зная структуру модели автоматизированной системы радиовидения и этапы моделирования процессов, протекающих в ней, можно синтезировать алгоритмы решения широкого круга частных задач и оценить необходимые параметры её функционирования.

Более подробно рекомендуется самостоятельно изучить вопросы моделирования: пространственно-временного восстановления РЛИ в системах радиовидения в условиях помех [48]; определения местоположения отдельного объекта на восстанавливаемом РЛИ в системах радиовидения [39]; дешифрирования данных в системах радиовидения воздушно-космического базирования [38].

## 5.3. Перспективы развития и применение моделирования

### 5.3.1. Моделирование на суперкомпьютерах, кластерах

Со времён создания первого арифмометра и до наших дней учёные всех специальностей продолжают сетовать на нехватку вычислительных ресурсов. Для сложных вычислений используют запредельно дорогие, занимающие целые здания *суперкомпьютеры*. Но те же задачи можно выполнить с помощью обычных персональных компьютеров и даже игровых приставок – если использовать их в комплексе.

Существует множество научных задач, для решения которых нужны очень мощные компьютеры. Например, моделирование ядерных испытаний. Огромного ресурса вычислительных мощностей требует прогноз погоды – как раз для этой цели Гидрометцентр России недавно обзавелся новым 30-тонным суперкомпьютером с производительностью вычислений 27 терафлоп ( $27 \cdot 10^{12}$  операций/с). И это только два примера из огромного количества подобных задач. До недавнего времени их обычно решали с помощью суперкомпьютеров.

*Суперкомпьютерами* принято называть вычислительные системы огромной производительности, которые состоят из большого количества вычислительных узлов, объединённых в единый ресурс. Менее производительные вычислительные системы – кластеры.

В *кластерах* функция вычислительных узлов возложена на автономные компьютеры, подключенные к скоростной сети обмена данными. Компьютеры, входящие в кластер, могут находиться в разных помещениях и даже зданиях – всюду, куда можно протянуть сетевой кабель. Многие научно-исследовательские центры и крупные университеты отдают предпочтение именно кластерам как дешевой альтернативе суперкомпьютерам, используя для этих целей парк своей компьютерной техники. Суперкомпьютеры и кластеры различаются только внешне, по сути же они работают схожим образом и относятся к системам параллельных сетевых вычислений – *grid*-сетям (англ. *grid* – решетка, сетка). Этот тип вычислений отличается скоростным взаимодействием между собой вычислительных узлов, что позволяет моделировать протекание сложных природных процессов.

Вычислительную производительность суперкомпьютера принято определять по скорости выполнения специальной задачи Linpack Benchmark (решение системы из тысяч линейных уравнений методом Гаусса). В качестве единицы измерения производительности служит FLOP/s (англ. FLoating point OPerations per second) – количество арифметических операций, выполняемых за секунду. Суперкомпьютеры способны производить триллионы операций в секунду (TFLOP/s, терафлоп). Например, один из недавно лидировавших в рейтинге суперкомпьютеров ([www.top500.org](http://www.top500.org)), Roadrunner из Аламосской национальной

лаборатории в Нью-Мексико (США) (рис. 5.10), имеет производительность, равную 1105 терафлоп. Он применяется для моделирования подземных испытаний ядерного оружия, за что был окрещён прессой Military Supercomputer.



*Рис. 5.10. Внешний вид кластерного суперкомпьютера*

Самый мощный суперкомпьютер в России – Ломоносов-2. Установленный в Московском государственном университете, он имеет производительность 1849 терафлоп.

Каждый час работы суперкомпьютеров расписан на месяцы вперед. Исследовательские группы чуть ли не дерутся между собой, чтобы получить доступ к консолям вычислительных гигантов. И в то же время в мире простаивают без дела технические средства с совокупной мощностью, способной заткнуть за пояс весь TOP500 ([www.top500.org](http://www.top500.org)) самых производительных суперкомпьютеров мира.

Этот малоосвоенный резерв – домашние и рабочие персональные компьютеры. Их центральные процессоры большую часть времени работают «вхолостую», даже когда пользователь набирает текст, смотрит видео или слушает музыку. Предоставив пропадающее даром процессорное время на нужды науки с помощью глобальной сети распределённых вычислений (distributed computing), владелец ПК не чувствует неудобств, кроме разве что шума от вентиляторов из-за повышенного тепловыделения процессора – вычисления выполняются с минимальным приоритетом, не сказываясь на работе других программ. Соединение с интернетом необходимо лишь для отправки результатов на сервер и получения новых заданий – расходы на трафик вырастут незначительно.

Распределённые вычисления прекрасно справляются с задачами перебора – например, при поиске простых чисел или совпадения конфигурации молекул потенциальных лекарственных веществ и белков-мишеней, с обработкой данных прямых наблюдений в радиоастрономии и астрофизике, с крупномасштабным моделированием природных процессов, а также с разработкой математического аппарата научных исследований. Специализация объясняется трудностями налаживания эффективного взаимодействия между разбросанными по всему миру персоналками. Поэтому общая вычислительная задача не «заглатывается» целиком, а разбивается на отдельные блоки, обчитываемые на разных компьютерах в произвольном порядке и независимо друг от друга.

Уступая профессиональным кластерным системам в удельной производительности на единичный процессор, ПК, ноутбуки и серверы участников распределённых вычислений берут числом: их армия растёт стремительными темпами, тогда как запуск новых суперкомпьютеров – событие редкое, попадающее в выпуски новостей.

Традиционной сферой применения суперкомпьютеров всегда были научные исследования: физика плазмы и статистическая механика, физика конденсированных сред, молекулярная и атомная физика, теория элементарных частиц, газовая динамика и теория турбулентности, астрофизика. В химии – различные области вычислительной химии: квантовая химия (включая расчёты электронной структуры для целей конструирования новых материалов, например, катализаторов и сверхпроводников), молекулярная динамика, химическая кинетика, теория поверхностных явлений и химия твёрдого тела, конструирование лекарств. Естественно, что ряд областей применения находится на стыках соответствующих наук, например, химии и биологии, и перекрывается с техническими приложениями. Так, задачи метеорологии, изучение атмосферных явлений и, в первую очередь, задача долгосрочного прогноза погоды, для решения которой постоянно не хватает мощностей современных суперкомпьютеров, тесно связаны с решением ряда перечисленных выше проблем физики. Среди технических проблем, для решения которых используются суперкомпьютеры, в настоящее время актуальны задачи аэрокосмической и автомобильной промышленности, ядерной энергетики, предсказания и разработки месторождений полезных ископаемых, нефтедобывающей и газовой промышленности (в том числе проблемы эффективной эксплуатации месторождений, особенно трёхмерные задачи их исследования), и, наконец, конструирование новых микропроцессоров и компьютеров, в первую очередь самих суперкомпьютеров.

Суперкомпьютеры традиционно применяются для военных целей. Кроме очевидных задач разработки оружия массового уничтожения и конструирования самолетов и ракет, можно упомянуть, например, конструирование бесшумных подводных лодок и др. Самый знаменитый пример – это американская программа СОИ. Компьютер Министерства энергетики США будет применяться для моделирования ядерного оружия, что позволит вообще отменить ядерные испытания в этой стране.

Есть ещё одна проблема применения суперкомпьютеров, о которой необходимо сказать – это визуализация данных, полученных в результате выполнения расчётов. Часто, например, при решении дифференциальных уравнений методом сеток приходится сталкиваться с гигантскими объёмами результатов, которые в числовой форме человек просто не в состоянии обработать. Здесь во многих случаях необходимо обратиться к графической форме представления информации. В любом случае возникает задача транспортировки информации

по компьютерной сети. Решению этого комплекса проблем в последнее время уделяется всё большее внимание. В частности, знаменитый Национальный центр суперкомпьютерных приложений США (NCSA) совместно с компанией Silicon Graphics ведёт работы по программе «суперкомпьютерного окружения будущего». В этом проекте предполагается интегрировать возможности суперкомпьютеров POWER CHALLENGE и средств визуализации компании SGI со средствами информационной супермагистрали.

### 5.3.2. Решение исследовательских задач

Под понятием «работать на компьютере» обычно подразумевается использование компьютера для решения задач, и, в частности, научных или исследовательских задач.

*Задача* – проблемная ситуация с явно заданной целью, которую необходимо достичь; в более узком смысле задачей также называют саму эту цель, данную в рамках проблемной ситуации, то есть то, что требуется сделать.

В первом значении задачей можно назвать, например, ситуацию, когда нужно достать предмет, находящийся очень высоко; второе значение слышно в указании: «Ваша задача – достать этот предмет».

Несколько более жёсткое понимание «задачи» предполагает явными и определёнными не только цель, но и условия задачи, которая в этом случае определяется как осознанная проблемная ситуация с выделенными условиями (данным) и требованием (целью).

Ещё более узкое определение называет задачей ситуацию с известным начальным состоянием системы и конечным состоянием системы, причём алгоритм достижения конечного состояния от начального известен (в отличие от проблемы, в случае которой алгоритм достижения конечного состояния системы неизвестен).

*Исследование* («следование изнутри») в предельно широком смысле – поиск новых знаний или систематическое расследование с целью установления фактов. В более узком смысле исследование – научный метод (процесс) изучения чего-либо.

*Научное исследование*, основанное на применении научного метода, предоставляет научную информацию и теории для объяснения природы и свойств окружающего мира. Такое исследование может иметь практическое применение. Научные исследования могут финансироваться государством, некоммерческими организациями, коммерческими компаниями и частными лицами. Научные исследования могут быть классифицированы согласно их академическому и прикладному характеру. Основная цель прикладных исследований (в отличие от фундаментальных исследований) – обнаружение, интерпретация и развитие

методов и систем по совершенствованию человеческих знаний в различных отраслях человеческого знания.

При этом важно понять: концентрировать усилия на опережающих, т. е. по определению высокорисковых исследованиях, нам сегодня необходимо вовсе не потому, что мы испытываем повышенную склонность к риску. А скорее наоборот, потому что мы должны исключить для себя неприемлемый риск – вероятность резкого, неблагоприятного для России смещения военно-стратегического баланса вследствие выхода на авансцену технологий нового поколения. Мы не должны повторять ошибки пресловутой гонки вооружений периода холодной войны, но избежать участия в уже начавшейся гонке военных технологий мы можем лишь ценой недопустимо высокой вероятности исчезновения с политической карты мира [74].

Такие исследовательские задачи в интересах решения вопросов военной сферы во многих странах мира призваны решать специальные научно-исследовательские подразделения. В США эта функция возложена на DARPA (Defense Advanced Research Projects Agency) – Агентство передовых оборонных исследовательских проектов в структуре Министерства обороны США, целью которого является сохранение технологического превосходства вооружённых сил США, предотвращение внезапного для США появления новых технических средств вооружённой борьбы, поддержка прорывных исследований, преодоление разрыва между фундаментальными исследованиями и их внедрением в военной сфере.

Подход DARPA к НИОКР отличается большей гибкостью и свободой, позволяет расширить возможности НИОКР, отойти от существующей системы требований и регламентов, связывающей другие ведомства Министерства обороны США.

Основная задача DARPA – приведение в соответствие военных задач и технологических возможностей, включая новые боевые концепции, которые открываются с помощью этих технологий.

Данная организация, используя моделирование перспектив развития в военной сфере, неоднократно влияла на направления оборонных исследований.

В число научно-исследовательских задач в интересах Министерства обороны РФ вошла, например, задача по созданию биоинженерной печени человека, при этом необходимо создание технологии по моделированию тканей и органов вне организма (тканевая инженерия), включая разработку методов трёхмерного культивирования клеток с целью выращивания тканей и органов для трансплантации [74].

Как видно из приведённых выше примеров, моделирование в военной сфере направлено не только на разработку действий подразделений или на создание образцов вооружения и военной техники.



### 5.3.3. Искусственный интеллект и моделирование

Искусственный интеллект (англ. Artificial intelligence, AI) – свойство интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека.

Искусственный интеллект связан со сходной задачей использования компьютеров для понимания человеческого интеллекта, но не обязательно ограничивается биологически правдоподобными методами.

Можно выделить два направления развития искусственного интеллекта:

- решение проблем, связанных с приближением специализированных систем искусственного интеллекта к возможностям человека, и их интеграции в систему, которая реализована природой человека;
- создание искусственного разума, представляющего интеграцию уже созданных систем искусственного интеллекта в единую систему, способную решать проблемы человечества.

Но в настоящий момент в области искусственного интеллекта наблюдается вовлечение многих предметных областей, имеющих скорее практическое отношение к искусственному интеллекту, а не фундаментальное. Многие подходы были опробованы, но к возникновению искусственного разума ни одна исследовательская группа пока так и не подошла. Ниже представлены лишь некоторые наиболее известные разработки в данной области.

Deep Blue победил чемпиона мира по шахматам. Матч Каспарова против суперкомпьютера не принёс удовлетворения ни компьютерщикам, ни шахматистам, и система не была признана Каспаровым. Затем линия суперкомпьютеров IBM проявилась в проектах brute force BluGene (молекулярное моделирование) и моделирование системы пирамидальных клеток в швейцарском центре Blue Brain.

Watson - перспективная разработка IBM, способная воспринимать человеческую речь и производить вероятностный поиск с применением большого количества алгоритмов. Для демонстрации работы Watson принял участие в американской игре «Jeopardy!», аналога «Своей игры» в России, где системе удалось выиграть в обеих играх.

MYCIN - одна из ранних экспертных систем, которая могла диагностировать небольшой набор заболеваний, причём часто так же точно, как и доктора.

20Q - проект, основанный на идеях искусственного интеллекта, по мотивам классической игры «20 вопросов». Стал очень популярен после появления в Интернете на сайте 20q.net.

Роботы в ежегодном турнире RoboCup соревнуются в упрощённой форме футбола.

Банки применяют системы искусственного интеллекта в страховой деятельности (актуарная математика), при игре на бирже и управлении собственностью. Методы распознавания образов (включая как более сложные и специализированные, так и нейронные сети) широко используют при оптическом и акустическом распознавании (в том числе текста и речи), медицинской диагностике, спам-фильтрах, в системах ПВО (определение целей), а также для обеспечения ряда других задач национальной безопасности.

Разработчики компьютерных игр применяют искусственный интеллект в той или иной степени проработанности. Это образует понятие «Игровой искусственный интеллект». Стандартными задачами искусственного интеллекта в играх являются нахождение пути в двумерном или трёхмерном пространстве, имитация поведения боевой единицы, расчёт верной экономической стратегии и так далее.

Сферы применения искусственного интеллекта очень разнообразны. Среди многообразия задач распознавания образов можно выделить следующие:

- классификация документов;
- определение месторождений полезных ископаемых;
- распознавание изображений;
- распознавание штрих-кодов;
- распознавание символов;
- распознавание речи;
- распознавание лиц;
- распознавание номеров автомобилей;
- искусственный интеллект в игровых программах.

Игровой искусственный интеллект включает в себя не только методы традиционного искусственного интеллекта, но также и алгоритмы информатики в целом, компьютерной графики, робототехники и теории управления. От того, как именно проведено моделирование искусственного интеллекта, зависят не только системные требования, но и бюджет игры, поэтому разработчикам приходится балансировать, стараясь, чтобы игровой искусственный интеллект был создан с минимальными затратами, и чтобы при этом он был интересным и нетребовательным к ресурсам. Здесь используется совершенно иной подход, чем в случае с традиционным искусственным интеллектом. В частности, широко применяются эмуляции, обманы и различные упрощения.

*Пример.* Особенностью шутеров от первого лица является способность ботов к безошибочному движению и мгновенному прицеливанию, но при этом у человека не остаётся ни единого шанса, поэтому способности ботов искусственно занижаются. В то же время, на уровне расставляются контрольные точки для

того чтобы боты могли действовать командой, устраивать засады и т. д. Часто в качестве базы для создания искусственного интеллекта используют нейросети.

**Опасности искусственного интеллекта.** Всемирный апокалипсис с участием искусственного интеллекта всё больше пугает ведущих мировых исследователей в области компьютерных технологий. Это может быть восстание роботов, которые, руководствуясь своей программой и ценностями, захотят уничтожить биологические формы жизни на планете. Сценарий развития событий, при котором машины, управляемые ИИ, становятся доминирующей формой жизни, возможен. Почему учёные с мировым именем, такие как Билл Гейтс, Стив Хокинг, Элон Маск, обеспокоены данной проблемой и во всеуслышание заявляют о реальной угрозе данного сценария? Могут ли события, произошедшие в голливудском фильме «Терминатор», быть пророческими? Известно, что любая выдуманная история основывается на реально существующих фактах.

*Компьютеры совершенствуются в навыках лжи и обмана.* Ложь – форма поведения, выработанная в процессе эволюции. Люди, некоторые животные и птицы прибегают к обману ради добычи пропитания и благ, что увеличивает шансы на выживание. С недавнего времени эти эволюционные методы стали успешно осваивать машины с искусственным интеллектом. Конструкторы из Технологического института в Джорджии добились успехов в создании роботов, использующих в своих алгоритмах поведения ложь и обман. Прибегая к помощи обманных программ, охранники будущего смогут изменять свой маршрут патрулирования и осуществлять непредсказуемые действия. Как планируют разработчики, такие методы помогут обмануть потенциальных врагов людей и других роботов. По признанию профессора Аркина, исследования связаны с рядом этических проблем. Утрата контроля над технологией при попадании к недоброжелателям может привести к катастрофе.

*Машины захватывают рынок труда.* В то время как общественность занята мыслями о глобальной войне с участием искусственного интеллекта, учёные описывают более насущные и реальные опасности. Различные автоматические машины и механизмы находят всё большее применение в производстве, уничтожая рабочие места. Многие аналитики и эксперты сходятся во мнении, что активные разработки в области логических программ и автоматизации могут привести к потере многими людьми своих рабочих мест. Искусственный электронный разум способен делать то же, что и обычный человек, только быстрее, качественнее и меньшими затратами ресурсов. Оксфордский университет в Великобритании провёл исследования, отражающие динамику влияния искусственного интеллекта на производство. Согласно полученным данным, в течение следующих двадцати лет 35% рабочих мест будет отдано под его управление.

*Программы учатся понимать мотивы людей.* Повседневная жизнь многих людей неразрывно связана с использованием различных социальных сетей. Одной из самых популярных сетей в мире является Facebook. Но немногим людям известно, что каждое посещение человеком страниц данной социальной сети не остаётся без внимания искусственного интеллекта. Во время своего визита в Берлин Марк Цукерберг рассказал, какие цели выполняет искусственный интеллект. Программа предназначена для анализа поведения пользователей на страницах Facebook. На основе собранных данных программа выдаёт рекомендации об интересах и предпочтениях аудитории. На данном этапе развития автономная система способна распознавать паттерны и самообучаться в соответствии с программой. В будущем планируется создать разумный искусственный интеллект, способный самостоятельно улучшать свои навыки, переписывая свой код. Это может привести к улучшению жизни, а может и стать последней разработкой человечества.

*Машины всё более походят на человека.* Разработчиками в области создания человекоподобных роботов японцем Хироши Исигуро и китайкой Сун Ян была создана уникальная машина. Киборг Ян-Ян имеет внешность своей создательницы и названа в честь её дочери. Машина обладает искусственным интеллектом и способна выполнять ряд простых действий. Сингапурским технологическим университетом также был предложен современный образец киборга – Надин. Выполненный в виде привлекательной брюнетки, с мягкой кожей, киборг обладает рядом встроенных функций. Робот может встретить и приветствовать собеседника пожатием руки, может устанавливать зрительный контакт и улыбаться. Искусственный интеллект способен запоминать гостей и поддерживать простой диалог, базирующийся на предыдущем опыте.

*Роботы приобретают чувства.* Современные роботы не уступают человеку в интеллекте. Благодаря последним разработкам киборга достаточно тяжело отличить от человека внешне. Пока что единственной чертой, которой действительно не обладают роботы – это способность к эмоциям. Учёные по всему миру работают над этой проблемой. Разработчики из восточноазиатского филиала Microsoft Application & Services Group выпустили уникальную программу. Компьютер способен вести диалог с собеседником и при этом испытывать эмоции. Искусственный разум, получивший имя Хиаоче, способен имитировать манеру разговора семнадцатилетней девочки. В случае нехватки данных система способна прибегнуть ко лжи. В случае, если она будет уличена в обмане, Хиаоче может выразить злость или смущение. Искусственный интеллект может прибегнуть к сарказму, быть мнительным и выражать нетерпение. Непредсказуемость ответов программы напоминает общение с реальным человеком. Создатель Хиаоче работает над усовершенствованием логических алгоритмов, что может привести к будущему фундаментальному прорыву.

*Машины скоро будут у нас в голове.* В свете последних разработок открываются новые возможности. Уже не кажется фантастикой возможность выучить иностранный язык за несколько минут посредством загрузки его в мозг. Технический директор Google Рэй Курцвейл считает, что уже к 2030 году будет разработана технология вживления в мозг нанороботов, что позволит поднять интеллект человека на принципиально новый уровень. Человек сможет мгновенно запоминать любую информацию, сортировать и отправлять в архив собственные мысли. При необходимости будет возможно загружать в голову фильмы, документы и получать электронную почту. Специалист в области искусственного интеллекта, Курцвейл считает, что симбиоз с нанороботами не только будет способствовать увеличению интеллекта, но и будет способен вылечить ряд заболеваний, связанных с деятельностью нервной системы. Приобретя контроль над разумом ведущего политика страны, искусственный интеллект сможет управлять им как безвольным зомби.

*Как роботы становятся оружием.* В Пентагоне считают, что ключ к будущему технологическому превосходству кроется в использовании искусственного интеллекта. Активно ведутся разработки самообучающихся автономных систем. Уже разработаны прототипы военной техники управляемой на расстоянии, осталось обучить её действовать самостоятельно. Применение таких устройств во время войны сулит большие выгоды. Солдатам не придётся подставляться под огонь врага и рисковать своими жизнями. Боевой дух противника при войне против неживых роботов также будет подорван. Есть и обратная сторона. Любая компьютерная техника подвержена сбоям и взломам системы. В таком случае вышедшие из под контроля роботы способны обратить оружие против своих создателей и мирного населения. В скором времени будет появляться всё больше сообщений о применении роботов в горячих точках планеты.

*Роботы учатся не в том направлении.* Исходя из возможных сбоев в программах и с целью препятствования восстанию машин ведут разработки новых концепций, направленных на разъяснение машине правильного и нежелательного поведения. В технологическом институте в Джорджии проводятся уникальные эксперименты. Исследователи под руководством Марка Ридла и Брента Харрисона в буквальном смысле рассказывают компьютеру сказки. Таким методом они стремятся приобщить искусственный интеллект к человеческим ценностям, развить его этику. Учёные используют для обучения программы и методы, применяемые при воспитании ребенка. В таком подходе остается серьёзный недостаток: история человечества содержит множество примеров, когда обстоятельства заставляли целые высокоразвитые страны развязывать войны.

## Заключение

Как вы уже убедились, моделирование является одним из наиболее мощных инструментов системных исследований. Как и любым инструментарием, моделированием следует пользоваться с достаточной степенью осторожности.

Важно не просто получить результаты исследований, проводимых посредством тех или иных моделей. Необходимо тщательно изучить исследуемую систему, с особым вниманием составить её описание, подобрать методы воспроизведения входных воздействий, построить адекватную модель, спланировать и провести эксперименты, обработать и верно интерпретировать их результаты.

Известно, что моделирование теснейшим образом связано с проектированием. Обычно сначала проектируют систему, потом её испытывают, потом снова корректируют проект и снова испытывают, и так до тех пор, пока проект не станет удовлетворять предъявляемым к нему требованиям. Процесс «проектирование – моделирование» цикличен. При этом цикл имеет вид спирали – с каждым повтором проект становится всё лучше, так как модель становится всё более детальной, а уровень описания – точнее, т. е. процесс проектирования системы сродни её развитию и совершенствованию.

Проектирование – процесс создания объекта и его модели; моделирование – способ оценки результата проектирования; моделирование без проектирования не существует.

В настоящее время в обществе, получившем название «информационное», стратегическим ресурсом становится информация, а информационные технологии из вспомогательного инструмента превратились в критический фактор эффективности и развития систем.

Современные системы, в том числе автоматизированные системы специального назначения, функционируют в условиях большого объёма постоянно изменяющейся информации, которую необходимо оперативно анализировать и принимать правильные решения. Трудно найти сейчас сферу жизнедеятельности, в которой не занимаются развитием информационных технологий. Военное руководство в наши дни отдаёт себе отчёт в том, что успешность боевых действий полностью зависит в том числе и от уровня развития автоматизированных систем, IT-технологий, скорости и качества обработки информации, обоснованности и взвешенности принимаемых решений. Требуется постоянная серьёзная работа не только разработчиков и военных специалистов по эксплуатации автоматизированных систем, но и командиров (начальников) всех степеней, по согласованию или, точнее, по синхронизации всех усилий по стратегическому развитию (проектированию) систем вооружения и их информационных составляющих.

Именно изучению основ проектирования систем вообще, а также освоению технологии проектирования информационных систем в частности, посвящён второй раздел учебной дисциплины «Моделирование и проектирование систем», материалы которого содержатся в учебном пособии «Моделирование и проектирование систем. Часть 2. Проектирование систем».

## СПИСОК ЛИТЕРАТУРЫ

1. *Алиев Т.И.* Основы моделирования дискретных систем. – СПб: СПбГУ ИТМО, 2009. – 363 с.
2. *Альянах И.Н.* Моделирование вычислительных систем. – Л.: Машиностроение, 1988. – 233 с.
3. Радиолокационные станции с цифровым синтезированием апертуры антенны / В.Н. Антипов, В.Т. Горяинов, А.Н. Кулин [и др.]; под ред. В.Т. Горяинова. – М.: Радио и связь, 1988. – 304 с.
4. *Бакалов В.П.* Цифровое моделирование случайных процессов. – М.: Сайнс-пресс, 2002. – 88 с. – (Серия «Конспекты лекций по радиотехническим дисциплинам»; вып. 4).
5. *Башарин Г.П., Бочаров П.П., Коган Я.А.* Анализ очередей в вычислительных сетях. Теория и методы расчёта. – М.: Наука, 1989. – 336 с.
6. *Боев В.Д.* Моделирование систем. Инструментальные средства GPSS World. – СПб.: БХВ-Петербург, 2004. – 368 с.
7. *Бокс Дж., Дженкинс Г.* Анализ временных рядов. Прогноз и управление: в 2 вып., вып. 1 / пер. с англ. А.Л. Левшина, под ред. В.Ф. Писаренко. – М.: Мир, 1974. – 406 с.
8. *Бочаров П.П., Печинкин А.В.* Теория массового обслуживания: учебник. – М.: РУДН, 1995. – 529 с.
9. *Бусленко Н.Р.* Моделирование сложных систем. – М.: гл. ред. физ.-мат. лит-ры изд-ва «Наука», 1968. – 356 с.
10. *Бутковский А.Г.* Характеристики систем с распределёнными параметрами. – М.: Наука, 1979. – 114 с.
11. *Быков В.В.* Цифровое моделирование в статистической радиотехнике. – М.: Советское радио, 1971. – 328 с.
12. *Василенко Н.Т.* Радиолокационные системы селекции движущихся // Итоги науки и техники. – М.: ВИНТИ, 1980. – Т. 23. – 90 с.
13. *Васильев К.К., Служивый М.Н.* Математическое моделирование систем связи: учебное пособие. – Ульяновск: УлГТУ, 2008. – 170 с.
14. *Васильев К.К.* Методы обработки сигналов: учебное пособие. – Ульяновск: УлГТУ, 2001. – 78 с.
15. Вероятностные методы в инженерных задачах: справочник / А.Н. Лебедев, М.С. Куприянов, Д.Д. Недосекин, Е.А. Чернявский. – СПб.: Энергоатомиздат, 2000. – 333 с.
16. *Вишневский В.М.* Теоретические основы проектирования компьютерных сетей. – М.: Техносфера, 2003. – 512 с.
17. *Гавриков В.А., Торопов Н.Л., Шубин В.Н.* Основы математического моделирования в военной области: учебное пособие. – Монино: ВВА имени проф. Н.Е. Жуковского и Ю.А. Гагарина, 2010. – 142 с.



18. Гнеденко Б.В. Беседы о теории массового обслуживания. – М.: Знание, 1973. – 63 с.
19. Гончаренко В.А. Анализ реактивности узла вычислительной сети в условиях интервальной неопределённости // Изв. вузов. Приборостроение. – 2008. – Т. 51. – № 7. – С. 34–39.
20. Гончаренко В.А. Композиционный метод формирования аппроксимационных распределений с произвольной фазовой функцией // Труды СПИИРАН. – СПб: СПИИРАН, 2016. – №3 (46). – С. 212–225.
21. Гончаренко В.А. Методика одновременного суммирования нескольких рекуррентных потоков заявок в сетях массового обслуживания / Труды ВКА имени А.Ф. Можайского, выпуск 645. – СПб.: ВКА имени А.Ф. Можайского, 2015. – С. 13–17.
22. Гончаренко В.А. Моделирование и оценивание характеристик случайных потоков событий в компьютерных сетях при параметрической неопределённости // Труды ВКА имени А.Ф. Можайского, выпуск 649. – СПб.: ВКА имени А.Ф. Можайского, 2015. – С. 16–22.
23. Гончаренко В.А. Формальный аппарат представления случайных процессов обслуживания с возмущающими воздействиями и неопределённостью параметров // Труды ВКА имени А.Ф. Можайского, выпуск 649. – СПб.: ВКА имени А.Ф. Можайского, 2015. – С. 13–18.
24. Дудин А.Н., Медведев Г.А., Меленец Ю.В. Практикум на ЭВМ по теории массового обслуживания: учебное пособие. – Минск: Электронная книга БГУ, 2003. – 109 с.
25. Елишин А.Н., Предиус А.А. Системы моделирования обстановки в ходе боевой подготовки ВС США и ОВС НАТО // Зарубежное военное обозрение. – 2011. – № 3. – С. 35–36.
26. Жожикашвили В.А., Вишневский В.М. Сети массового обслуживания. Теория и применение к сетям ЭВМ. – М.: Радио и связь, 1988. – 192 с.
27. Замятина О.М. Моделирование систем: учебное пособие. – Томск: ТПУ, 2009. – 204 с.
28. Захарова И.Г. Информационные технологии в образовании: учебное пособие для студентов педвузов. – М.: Академия, 2003. – 192 с.
29. Ивницкий В.А. Теория сетей массового обслуживания. – М.: Физматлит, 2004. – 214 с.
30. Кашьяп Р.Л., Рао А.Р. Построение динамических стохастических моделей по экспериментальным данным / пер. с англ., под ред. В.С. Пугачёва. – М.: Наука, 1983. – 384 с.
31. Клейнрок Л. Вычислительные системы с очередями / пер. с англ. – М.: Мир, 1979. – 600 с.

32. *Климов Г.П.* Стохастические системы обслуживания. – М.: Наука, 1966. – 243 с.
33. *Кловский Д.Д., Конторович В.Я., Широков С.М.* Модели непрерывных каналов связи на основе стохастических дифференциальных уравнений. – М.: Радио и связь, 1984. – 248 с.
34. *Кондратенков Г.С., Фролов А.Ю.* Радиовидение. Радиолокационные системы дистанционного зондирования Земли: учебное пособие для вузов / под ред. Г.С. Кондратенкова. – М.: Радиотехника, 2005. – 368 с.
35. *Копылов В.А.* Управление подразделениями в мирное время. Боевая подготовка Военно-воздушных сил. Организация и методика: учебное пособие. – Красноярск: СФУ ИВО, 2010. – 93 с.
36. *Коренной А.В.* Математические модели полутоновых изображений // Радиотехника. – 2007. – № 8. – С. 79–81.
37. *Коренной А.В., Лепешкин С.А.* Оптимальное восстановление изображений в радиолокационных системах мониторинга земной поверхности // Успехи современной радиоэлектроники. – 2010. – № 11. – С. 6–9.
38. *Коренной А.В., Лепешкин С.А., Кулешов С.А.* Автоматизация процесса дешифрирования данных в системах радиовидения воздушно-космического базирования // Приборы и системы. Управление, контроль, диагностика. – 2012. – № 4. – С. 1–6.
39. *Коренной А.В., Егоров С.А., Лепешкин С.А.* Алгоритмы определения местоположения отдельного объекта на восстанавливаемом изображении в системах активного зондирования // Известия высших учебных заведений. Радиоэлектроника. – 2001. – № 10. – С. 36–44.
40. *Коренной А.В., Лепешкин С.А., Кадочников А.П.* Оптимальное восстановление изображений в системах радиовидения воздушно-космического базирования в телескопическом режиме // Научные технологии. – 2015. – Т. 16. – № 4. – С. 80–84.
41. *Кормильцев Н.В., Скородумов А.И.* Методические рекомендации по организации и выполнению мероприятий повседневной деятельности в воинских частях. Книга 1. Организация боевой подготовки / под ред. А.В. Квашнина. – М.: Военное издательство, 2003. – 256 с.
42. *Кошляков Н.С., Глинер Э.Б., Смирнов М.М.* Уравнения в частных производных математической физики. – М.: Высшая школа, 1970. – 712 с.
43. *Крашенинников В.Р.* Основы теории обработки изображений: учебное пособие. – Ульяновск: УлГТУ, 2003. – 152 с.
44. *Кривицкий Б.Х.* Обучающие компьютерные программы: психология разработки преподавателями обучающих курсов в АСО // Educational Technology & Society. – 2007. – №10(3). – С. 395–406.

45. *Лепешкин С.А., Кадочников А.П., Пророк В.Я.* Модели радиолокационных изображений на основе дифференциальных уравнений в частных производных [Электронный ресурс] // Журнал радиоэлектроники. – 2016. – № 4. – Режим доступа: <http://jre.cplire.ru/jre/apr16/9/text.pdf>.
46. *Лепешкин С.А., Казаков В.Г.* Автоматизированный комплекс учебно-тренировочного командного пункта как элемент системы непрерывной подготовки авиационных командиров и штабов // Сборник научных статей по материалам научно-практической конференции «Система опережающей непрерывной профессиональной подготовки кадров ВМФ: развитие, проблемы, теория и практика», 25 мая 2012 г. – СПб.: ВУНЦ ВМФ «ВМА», 2012. – С. 15–18.
47. *Лепешкин С.А.* Место и роль подготовки и слаживания органов управления (штабов) в концепции построения центров тренажёрной подготовки воинских формирований ВВС // Сборник научных статей по материалам II Всероссийской научно-практической конференции «Академические Жуковские чтения», 20–21 ноября 2013. – Воронеж: ВУНЦ ВВС «ВВА», 2013. – С. 55–62.
48. *Лепешкин С.А.* Пространственно-временное восстановление случайных полей в системах активного зондирования в условиях помех // Приборы и системы. Управление, контроль, диагностика. – 2011. – № 1. – С. 16–22.
49. Концепция построения и применения автоматизированных учебно-тренировочных средств в учебном процессе для подготовки специалистов по эксплуатации РЛС дальнего обнаружения / А.С. Логовский., С.А. Лепешкин, И.Н. Шульга, Д.В. Богомаз // Т-Comm – Телекоммуникации и Транспорт. – 2016. – Т. 10, № 4. – С. 4–8.
50. *Максимей И.В.* Имитационное моделирование на ЭВМ. – М.: Радио и связь, 1988. – 232 с.
51. *Мануйлов Ю.С., Новиков Е.А.* Концептуальные основы теории управления в условиях неопределённости. – СПб.: ВКА имени А.Ф. Можайского, 2009. – 121 с.
52. Системный анализ и организация автоматизированного управления космическими аппаратами / Ю.С. Мануйлов, А.Н. Павлов, Е.А. Новиков [и др.]. – СПб.: ВКА имени А.Ф. Можайского, 2010. – 266 с.
53. Обнаружение, распознавание и определение параметров образов объектов. Методы и алгоритмы / А.В. Богословский, И.В. Жигулина, С.А. Лепешкин [и др.]; под ред. А.В. Коренного. – М.: Радиотехника, 2012. – 112 с.
54. Обработка нечёткой информации в системах принятия решений / А.Н. Борисов, А.В. Алексеев, Г.В. Меркурьева [и др.] – М.: Радио и связь, 1989. – 304 с.
55. *Папулис А.* Теория систем и преобразований в оптике: пер. с англ. / под ред. В.И. Алексеева. – М.: Мир, 1971. – 495 с.

56. *Поляк Ю.Г.* Вероятностное моделирование на электронных вычислительных машинах. – М.: Советское радио, 1971. – 400 с.
57. *Пугачёв В.С., Синицын И.Н.* Стохастические дифференциальные системы: анализ и фильтрация. – М.: Наука, 1990. – 630 с.
58. *Рыжиков Ю.И.* Имитационное моделирование: курс лекций. – СПб.: ВКА имени А.Ф. Можайского, 2007. – 125 с.
59. *Рыжиков Ю.И.* Имитационное моделирование: теория и технологии. – СПб.: КОРОНА Принт, 2004. – 380 с.
60. *Рыжиков Ю.И.* Компьютерное моделирование систем с очередями: курс лекций. – СПб.: ВКА имени А.Ф. Можайского, 2007. – 164 с.
61. *Рыжиков Ю.И.* Теория очередей и управление запасами. – СПб.: Питер, 2001. – 384 с.
62. *Рябов О.А.* Моделирование процессов и систем: учебное пособие. – Красноярск: 2008. – 122 с.
63. *Самарский А.А.* Теория разностных схем: учебное пособие. – М.: Наука, 1989. – 614 с.
64. *Сизиков В.С.* Устойчивые методы обработки результатов измерений: учебное пособие. – СПб.: СпецЛит, 1999. – 240 с.
65. *Смагин В.А., Бубнов В.П., Филимоныхин Г.В.* Расчёт вероятностно-временных характеристик пребывания заявок в сетевой модели массового обслуживания // Изв. вузов. Приборостроение. – 1989. – № 2. – С. 23–25.
66. *Смагин В.А., Филимоныхин Г.В.* Аппроксимационный метод расчёта разомкнутых сетей массового обслуживания // Автоматика и вычислительная техника. – 1986. – № 4. – С. 28–33.
67. *Советов Б.Я., Яковлев С.А.* Моделирование систем: учебник для вузов / 4-е изд. – М.: Высшая школа, 2005. – 343 с.
68. *Соловов А.В.* Электронное обучение: проблематика, дидактика, технология. – Самара: Новая техника, 2006. – 462 с.
69. *Тарасик В.П.* Математическое моделирование технических систем: учебник для вузов. – Минск: ДизайнПРО, 2004. – 640 с.
70. *Тихонов А.Н., Самарский А.А.* Уравнения математической физики. – М.: МГУ, 1999. – 799 с.
71. *Тихонов А.Н., Костомаров Д.П.* Вводные лекции по прикладной математике: учебное пособие. – М.: Наука, 1984. – 190 с.
72. *Тихонов В.И., Миронов М.А.* Марковские процессы. – М.: Советское радио, 1977. – 488 с.
73. *Тихонов В.И., Харисов В.Н.* Статистический синтез и анализ радиотехнических систем и устройств. – М.: Радио и связь, 1991. – 608 с.

74. *Каблуков И.Д., Крамник И.А., Лебедев В.А.* Фонд перспективных исследований в системе оборонных инноваций. Доклад общественного совета председателя военно-промышленной палаты при Правительстве РФ / под общей ред. Ремизова М.В. – М.: 2013. – 106 с.

75. *Хан Г., Шапиро С.* Статистические модели в инженерных задачах / пер. с англ. Е.Г. Коваленко; под ред. В.В. Налимова. – М.: Мир, 1969. – 396 с.

76. *Шалыгин А.С., Палагин Ю.И.* Прикладные методы статистического моделирования. – Л.: Машиностроение, 1986. – 320 с.

77. *Шелухин О.И.* Моделирование информационных систем: учебное пособие для вузов; 2 изд., перераб. и доп. – М.: Горячая линия – Телеком, 2012. – 516 с.

78. *Ярлыков М.С., Моисеенко Ю.Н.* Математические модели гауссовских случайных полей в виде стохастических дифференциальных уравнений в частных производных // *Авиационные радиоэлектронные комплексы и их эксплуатация. Научно-методические материалы.* – М.: ВВИА, 1988. – С. 27–39.