

МИНИСТЕРСТВО ОБРАЗОВАНИЯ УКРАИНЫ  
Национальный аэрокосмический университет  
им. Н.Е. Жуковского  
"Харьковский авиационный институт "

М.Ф. Бабаков, А.В. Попов

Методы машинного моделирования  
в проектировании электронной аппаратуры

Учебное пособие

**AS IS**

Харьков «ХАИ» 2001

УДК 621.396.6

Методы машинного моделирования в проектировании электронной аппаратуры /М.Ф. Бабаков, А.В. Попов. -Учеб. пособие - Харьков: Нац. аэрокосмический ун-т "Харьк. авиац. ин-т", 2001. – 90 с.

Приведены теоретические основы, методы и алгоритмы машинного моделирования, используемые при проектировании электронной аппаратуры, которые входят в программу подготовки бакалавров по направлению "Электронные аппараты". Рассмотрены общие принципы моделирования, методы моделирования сигналов, воздействий и систем на аналоговых и цифровых вычислительных машинах.

Приведены необходимые справочные данные и примеры программной реализации основных алгоритмов цифрового моделирования.

Для студентов факультета радиотехнических систем летательных аппаратов.

Ил. 36. Табл. 2 Библиогр.: 9 назв.

Рецензенты: канд. техн. наук, доц. К.В. Скульский,  
канд. техн. наук С.И. Хоменко

© Национальный аэрокосмический университет  
им. Н.Е.Жуковского  
"Харьковский авиационный институт ", 2001 г.

## Введение

Моделирование является важным этапом проектирования электронной аппаратуры. Оно позволяет провести такие исследования сложных систем, которые не могут быть выполнены традиционными методами. Моделирование существенно снижает сроки и стоимость проектирования и за счет анализа большого количества вариантов повышает эффективность разрабатываемой системы.

Согласно системному подходу к проектированию разработчику электронной системы необходимо выявить все внешние связи и факторы, влияющие на функционирование системы, и в процессе моделирования проектируемой системы разработать адекватные модели не только системы, но и сигналов и внешних факторов, которые могут быть не только детерминированными, но и случайными. Моделированию подлежат не только принципы функционирования электронной аппаратуры и используемые схемотехнические решения, но и конструкция аппаратуры и ее элементы, технологические процессы изготовления, процессы испытаний, эксплуатации и ремонта.

Ключевые подходы к построению моделей основываются на понятиях и методах теории подобия и моделирования физических явлений. Разработка моделей основывается на строгих математических соотношениях теории сигналов, случайных процессов и теории систем. Реализация моделей требует знания основ теории алгоритмов, программирования, вычислительной техники и вычислительной математики. Обработка результатов моделирования с целью повышения адекватности и достоверности требует применения методов и алгоритмов математической статистики.

В данном учебном пособии приведены основные сведения из теории подобия и моделирования, рассмотрены принципы и методы моделирования сигналов, воздействий и систем с использованием как аналоговых вычислительных комплексов, так и современных цифровых вычислительных машин. Особое внимание уделено практическим вопросам реализации алгоритмов цифрового моделирования.

## **1. МЕТОДЫ ТЕОРИИ ПОДОБИЯ И МОДЕЛИРОВАНИЯ ПРИ ПРОЕКТИРОВАНИИ ЭА**

Одним из начальных этапов процесса проектирования электронной аппаратуры является моделирование проектируемого изделия, системы либо технологического процесса.

Ключевые подходы к построению моделей основываются на понятиях и методах теории подобия и моделирования физических явлений.

### **1.1. Понятия подобия и моделирования явлений**

Модель – это естественный или искусственный объект, находящийся в соответствии с изучаемым объектом или с какой-либо из его сторон. Моделирование есть метод опосредованного познания, при котором изучаемый объект (оригинал) находится в некотором соответствии (соотношении подобия) с другим объектом (моделью), причем объект-модель способен в том или ином отношении замещать оригинал на некоторых стадиях познавательного процесса.

Выделяют два характерных вида моделирования:

1. Мысленное моделирование, при котором на основе информации об исследуемых явлениях в сознании формируются образы, имеющие определенное сходство с соответствующими объектами. Математическая запись, составленная на основе этих образов и содержащая описание физических или других (например, экономических) закономерностей, и есть модель.
2. Материальное моделирование, при котором создается система - модель, имеющая определенное сходство с системой-оригиналом. Две эти материально реализованные системы, из которых одна рассматривается как отображение другой, связаны соотношением подобия.

Подобие явлений означает, что данные о протекании процессов, полученные при изучении одного явления, можно распространить на все явления, подобные данному. Модель не дает и не должна давать

подобие абсолютно всех процессов, содержащихся в явлении, а обеспечивает подобие только тех процессов, которые удовлетворяют критериям подобия, найденным на основе теории подобия.

Таким образом, понятие модели всегда требует введения понятия подобия. Каждому виду подобия соответствует свой вид моделирования.

## 1.2. Классификация методов моделирования

Классификация методов моделирования и подобия представлена рис. 1.1.

При полном моделировании и подобии обеспечивается подобие движения материи в основных формах ее существования, т.е. во времени и пространстве, и имеет место прямо-пропорциональное соотношение параметров модели  $X$  и оригинала  $Y$

$$x_i = m_i \cdot y_i.$$

где  $m_i$  – масштабный коэффициент;  $y_i$  – параметры системы или ее режима.

При неполном моделировании и подобии объект и модель, подобны только частично, например, или только в пространстве.

При приближенном моделировании некоторые факторы, имеющие незначительное влияние, моделируются приближенно или совсем не моделируются. Между некоторыми параметрами систем или их режимов при этом может не существовать соотношений подобия и  $x_i \neq m_i \cdot y_i$  или же  $x_i \approx m_i \cdot y_i$ . Это заведомо вызывает погрешность, которая должна быть оценена. Приближенное моделирование может быть как полным, так и неполным.

Каждый из рассмотренных трех способов подобия и моделирования может быть разделен на три вида:

- 1) мысленное, идеально-теоретическое;
- 2) мысленное аналитическое, использующее ту или иную аппаратуру для подтверждения своих отвлеченных представлений (расчеты на ЭВМ, аналоги, иллюстрирующие мысленно созданные положения);



Рис. 1.1

3) материальное, реально-практическое или вещественно-агрегатное.

Как мысленное, так и материальное моделирование может быть либо детерминированным, отражающим процессы с однозначно определенными причинами и их следствиями, либо стохастическим, отражающим вероятностные события.

Основные виды моделирования распадаются на ряд групп и подгрупп (см. рис. 1.1).

К наглядному моделированию относят различные мысленные представления (гипотезы) в форме тех или иных воображаемых моделей, например, планетарные модели атомов или молекул, причем для них могут создаваться модели в виде наглядных аналогов. К ним относятся и макеты, являющиеся геометрическими копиями изучаемого или проектируемого объекта.

Символическое (знаковое) моделирование предусматривает, прежде всего, упорядоченную запись, к которой относится использование условных знаков для описания явления. Сюда относятся знаковые построения и записи различных операций, основанные на топологии и теории графов.

Математическое мысленное моделирование включает модели, являющиеся средством для проверки теории. Это схемы замещения различных электрических схем и элементов электроники, алгоритмы и программы для ЭВМ.

При натурном моделировании в объект, подлежащий исследованию, не вносят специальных изменений. Но обязательным для этого вида моделирования, как и для любого другого, является требование обработки результатов экспериментов с помощью теории подобия.

Физическое моделирование отличается прежде всего тем, что исследования проводятся на установках, обладающих физическим подобием, т.е. сохраняющих полностью (или хотя бы в основном) природу явлений.

При аналого-цифровом моделировании (его называют математическим) физика исследуемого процесса не сохраняется. Моделирова-

ние основывается на изоморфизме уравнений, т.е. их способности описывать различные по своей природе явления и выявлять различные функциональные связи. Аналоговое прямое моделирование использует прямую непосредственную аналогию между величинами одного явления, величинами, присущими другому явлению. Например, движение маятника или изменения электрического тока в колебательном контуре. При структурном аналоговом моделировании воспроизводится не весь процесс в целом, а отдельные математические операции, которые выполняют элементы модели.

Цифровые модели имеют своей базой цифровые элементы, в частности, обычные цифровые ЭВМ. Разработанный алгоритм, программа, вводимая в ЭВМ, и критериальная запись полученных результатов - есть модель явления.

Гибридные модели являются сочетанием аналоговых моделей и специализированных цифровых ЭВМ. В такой комбинации удачно используются положительные свойства аналоговых моделей (наглядность, простота набора схемы, быстроедействие) и цифровых ЭВМ (точность хранения в памяти нужных данных и т.д.).

Сравнительно недавно появившиеся имитационные модели являются разновидностью цифровых. Они позволяют "проигрывать" множество вариантов и становятся эффективным средством анализа и прогноза явлений.

При функциональном или кибернетическом моделировании развитие системы рассматривается без анализа внутренних причинных связей, а только в плане уравнивания ее со средой с учетом механизма обратной связи. В таких моделях объекты отражаются главным образом на основании информационных процессов и процессов управления.

### **1.3. Теоремы теории подобия и применение их выводов при моделировании и экспериментах**

В теории подобия имеются три основные теоремы, которые формируют свойства подобных явлений, устанавливают необходимые и



достаточные условия их подобия и возможность преобразования функциональной зависимости между физическими параметрами в критериальное уравнение.

Первая теорема. Сравнимые явления (натурный объект и модель) подобны, если они имеют некоторое одинаковое сочетание параметров, называемых критериями подобия. Под критериями  $\pi_i$ , или инвариантами подобия понимаются безразмерные комплексы физических величин, определяющих то или иное физическое явление. Для подобных явлений  $\pi_i = idem$ , т.е. они одинаковы для всех рассматриваемых явлений.

Критерии подобия определяются из анализа размерностей физических параметров или из условия тождественности уравнений, описывающих исследуемую систему. Критерии подобия любого явления могут преобразовываться в критерии другой формы за счет перемножения, деления, возведения в степень или умножения на любой коэффициент.

Вторая теорема ( $\pi$  – теорема). Функциональная зависимость между физическими параметрами системы может быть представлена в виде зависимости между составленными из них критериев подобия (критериальное уравнение).

Использование этой теоремы позволяет получить в экспериментальных работах значительные преимущества:

- переход к безразмерным соотношениям позволяет распространять результаты аналитического или экспериментального исследования, проведенного применительно к конкретному явлению, на целый ряд подобных явлений;
- можно находить критериальные соотношения, не имея математического описания процесса в виде уравнения, а зная только все участвующие в процессе величины и их размерности;
- применение безразмерных величин уменьшает число параметров, входящих в исследуемое уравнение, что упрощает математические операции с ним;
- при проведении экспериментальных исследований не нужно изу-

чать влияние каждого фактора в отдельности.

$\pi$  – теорема касается только полных уравнений, к которым относятся физические уравнения, обладающие следующими свойствами:

1. Они размерно-однородны, т.е. размерности всех слагаемых уравнения одинаковы.
2. Для них выполняется правило Фурье: физические уравнения не меняют своего вида при изменении величины единицы измерения. Действительно, поскольку все слагаемые уравнения одной размерности и в общем случае представляют собой произведение нескольких физических величин (трансцендентные функции на размерность слагаемых не влияют), то размерность каждого слагаемого будет иметь вид:  $[A]^{\alpha_1} \cdot [B]^{\beta_1} \cdot [D]^{d_1}$  где  $[A]$ ,  $[B]$ ,  $[D]$  – размерности первичных единиц измерения. Если в таком уравнении изменить значение первичных единиц измерения в какое-то число раз, например, единицу размерности  $[A]$  - в  $\frac{1}{a}$  раз, единицу размерности  $[B]$  - в  $\frac{1}{b}$  раз, то значение каждого слагаемого изменится в  $a^{\alpha_1} \cdot b^{\beta_1}$  раз. Поскольку этот сомножитель является общим для всех слагаемых, его можно сократить в левой и правой частях уравнения. Таким образом, уравнение сохранит свой первоначальный вид и для новых единиц измерения.

Выводы приведенных теорем распространяются на системы любой сложности и любого вида, если выполняется ряд дополнительных положений о подобии. Согласно этим положениям:

1. Подобие сложных систем, состоящих из подобных по отдельности подсистем, обеспечивается подобием всех сходственных элементов, являющихся общими для подсистем.
2. Для подобия нелинейных систем, систем с переменными параметрами, а также анизотропных требуется совпадение относительных характеристик сходственных параметров, являющихся нелинейными или переменными, и одинаковой относительной анизотропии.
3. Для подобия явлений статистического характера необходимо сов-

падение плотностей вероятностей сходных параметров, представленных в виде относительных характеристик, и физической реализуемости сходной корреляции между статистически заданными параметрами, входящими в условие однозначности.

Третья теорема.  $\pi$  – теорема исходит из того, что сравниваемые явления заведомо подобны: она характеризует свойства подобных явлений, т.е. необходимые условия подобия. Практически важным является вопрос, как же установить и создать подобие явлений, т.е. осуществить моделирование. На этот вопрос и отвечает третья теорема, формулирующая необходимые и достаточные условия подобия и утверждающая, что для подобия явлений должны быть соответственно одинаковыми определяющие критерии подобия и подобны (в простейшем случае пропорциональны) сходные параметры, входящие в условия однозначности.

Например, уравнение  $U = iR + L \frac{di}{dt}$ , описывающее изменение тока в RL – цепи, дает множество подобных процессов  $i(t)$ , в зависимости от граничных ( $u$ ) или начальных условий  $i = i_0$  при  $t = t_0$ , а также значений параметров R, L, . Задание некоторых указанных факторов или условий однозначности выделяет из бесконечного множества процессов, которым соответствует данное уравнение, те конкретные процессы, подобие которых необходимо обеспечить.

#### **1.4. Способы определения критериев подобия явлений**

Практическое применение рассмотренных теорем увидим, проанализировав способы определения критериев подобия.

Критерии подобия определяются в основном двумя способами:

1. Приведением уравнений физического процесса к безразмерному виду (необходимо иметь уравнение исследуемого процесса).
2. Применением  $\pi$  – теоремы. Им можно пользоваться и в случаях, когда известны только параметры процесса, а уравнение процесса неизвестно.

1 способ применяется не только к алгебраическим, но и к дифференциальным и интегральным уравнениям, поскольку операции дифференцирования и интегрирования не влияют на однородность уравнения в целом, а по отношению к размерным членам уравнений неоднородные функции представляют собой безразмерные коэффициенты.

Для определения критериев подобия этим способом необходимо произвести деление членов уравнения на какой-то из них, отбросив затем символы дифференцирования и интегрирования, а также неоднородные функций. К полученным таким путем основным критериям необходимо присовокупить дополнительные критерии - аргументы, входящие в неоднородные члены уравнения.

Пример. Колебание груза массой  $m$  на пружине жесткости  $C$  с силой сопротивления вязкой среды, пропорциональной скорости:  $F_C = -kl$ . На груз действует вынуждающая сила  $F_B = F \sin \omega t$ . Уравнение движения имеет вид:

$$m \cdot l'' + k \cdot l' + c \cdot l = F \sin \omega t.$$

Отбросив операции дифференцирования и неоднородную функцию  $\sin \omega t$ , будем иметь:

$$\frac{m \cdot l}{t^2} + \frac{k \cdot l}{t} + c \cdot l = F.$$

Разделим полученное уравнение на первый член и получим основные критерии подобия:

$$\pi_1 = \frac{F \cdot t^2}{m \cdot l}; \quad \pi_2 = \frac{k \cdot t}{m}; \quad \pi_3 = \frac{c \cdot t^2}{m}.$$

Дополнительный критерий  $\pi_4 = \omega t$  имеет смысл, если воспроизводится изменение силы  $F_B$  именно по синусоидальному закону. Критериальное уравнение имеет вид  $\pi_1 = (\pi_2 + \pi_3 + 1)$ .

Из теории подобия следует, что подобие модели и оригинала не

нарушится, если коэффициенты  $k$  и  $C$  будут нелинейны при одинаковых относительных характеристиках модели и оригинала.

2 способ. Для его реализации в общем случае необходимо выполнить следующее:

1. Выявить  $n$  параметров  $x_1, x_2, \dots, x_n$ , которые характеризуют изучаемый процесс;
2. Составить матрицу размерностей  $n$  параметров;
3. Установить число  $k$  независимых между собой параметров;
4. Представить описание изучаемого явления в критериальной форме;
5. Составить выражения  $n - k$  критериев подобия во всех формах записи.

Рассмотрим реализацию этого способа на примере, приведенном выше.

1. Параметры, определяющие процесс движения груза:

$$x_1 = m; x_2 = \omega; x_3 = F; x_4 = l; x_5 = k; x_6 = C; x_7 = t.$$

2. Выберем три независимых единицы измерений применительно к системе измерений LMT, тогда система уравнений размерностей и соответствующая ей матрица размерностей примут вид:

	$[M]$	$[L]$	$[T]$
$[m] = [M]^1 [L]^0 [T]^0$	$x_1$	1	0
$[\omega] = [M]^0 [L]^0 [T]^{-1}$	$x_2$	0	-1
$[F] = [M]^1 [L]^1 [T]^{-2}$	$x_3$	1	-2
$[l] = [M]^0 [L]^1 [T]^0$	$x_4$	0	1
$[k] = [M]^1 [L]^0 [T]^{-1}$	$x_5$	1	-1
$[C] = [M]^1 [L]^0 [T]^{-2}$	$x_6$	1	-2
$[t] = [M]^0 [L]^0 [T]^1$	$x_7$	0	1

3. Устанавливаем  $k$  независимых параметров. Для этого находим параметры, определитель матрицы размерностей которых отличен

от нуля. В примере такими параметрами являются  $x_1, x_2, x_3$ . Все остальные параметры определяются путем деления и умножения степеней этих параметров. Это выполняется, когда соответствующие строки матрицы размерностей являются линейными комбинациями первых трех строк.

4. Представляем описания в критериальной форме:

$$\pi_j = \frac{x_{k+j}}{x_1^{\alpha_{k+j}} \cdot x_2^{\beta_{k+j}} \dots x_k^{\xi_{k+j}}}.$$

Для нахождения показателей  $\alpha_{k+j}, \beta_{k+j}, \dots, \xi_{k+j}$  составляем систему линейных уравнений между строками матрицы размерностей

$$\alpha_{k+j} \cdot x_1 + \beta_{k+j} \cdot x_2 + \dots + \xi_{k+j} \cdot x_3 = x_{k+j}.$$

Для критерия  $\pi_1$  из рассматриваемого примера в соответствии со сказанным будем иметь:

$$\begin{cases} \alpha_4 \cdot 1 + \beta_4 \cdot 0 + \xi_4 \cdot 1 = 0; \\ \alpha_4 \cdot 0 + \beta_4 \cdot 0 + \xi_4 \cdot 1 = 0; \\ \alpha_4 \cdot 0 + \beta_4 \cdot (-1) + \xi_4 \cdot (-2) = 0. \end{cases}$$

Решая полученную систему, находим:  $\alpha_4 = -1, \beta_4 = -2, \xi_4 = 1$ .

Тогда для критерия

$$\pi_1 = \frac{x_4 \cdot x_1 \cdot x_2^2}{x_3} = \frac{l \cdot m \cdot \omega^2}{F}.$$

Аналогично для других критериев:

$$\pi_2 = \frac{k}{m \cdot \omega}, \quad \pi_3 = \frac{C}{m \cdot \omega^2}, \quad \pi_4 = \omega \cdot t.$$

Эти критерии методом преобразования можно привести к найденным ранее:

$$(\pi_1^{-1}) \cdot \pi_4^2 = \frac{F \cdot t^2}{m \cdot l}, \quad \pi_2 \cdot \pi_4 = \frac{k \cdot t}{m}, \quad \pi_3 \cdot \pi_4 = \frac{C \cdot t^2}{m}.$$

### 1.5. Понятие об автомодельности явлений

При постановке и обработке экспериментов могут встречаться явления, называемые автомодельными, что означает автоматическое подобие какого-либо явления оригиналу независимо от абсолютных величин параметров элементов той системы, в которой данное явление протекает.

Поясним это на примере уравнения  $U = iR + L \frac{di}{dt}$ . Критерии подобия этого уравнения:  $\pi_1 = \frac{i}{(U \cdot R^{-1})}$ ,  $\pi_2 = \frac{t}{(L \cdot R^{-1})}$ , причем  $\pi_1 = f(\pi_2)$ . Для подобия какого-либо второго процесса первому необходимо равенство в сходственные моменты времени критерия  $\pi_2$ , т.е.

$$\frac{t_1}{L_1 \cdot R_1^{-1}} = \frac{t_2}{L_2 \cdot R_2^{-1}}.$$

Очевидно, что при любых наперед заданных значениях  $L_2, R_2$  это можно обеспечить выбором величины  $t_2$ , следовательно,  $m_t = \frac{t_1}{t_2}$ . Таким образом, при соответствующем выборе масштаба времени рассматриваемые процессы будут подобны при любых значениях параметров  $R_2, L_2$  второй системы. Одним из формальных признаков автомодельности является условие  $m_{\text{эс}} \leq k$ , где  $m_{\text{эс}}$  – число параметров элементов системы.

С учетом этого можно сделать вывод, что двучленные уравнения всегда автомодельны. Таким образом, для автомодельных процессов критерии подобия служат не для расчета величин параметров элементов модели, а лишь для определения масштабов при любых значениях параметров элементов модели.

## **1.6. Принципы применения методов теории подобия и моделирования**

Методы моделирования основываются на теории подобия, которая базируется на трех основных теоремах и ряде дополнительных положений. Прежде чем приступить к созданию какой-либо модели, необходимо найти критерии подобия для процессов, протекающих в оригинале, и выделить из их числа определяющие критерии. При этом, если с точки зрения математики безразлично, какой из критериев подобия считать определяющим, то при создании физических моделей необходимо учитывать и возможности практической реализации на модели.

Критерии подобия определяют масштабы, связывающие параметры модели и оригинала и выявляют наиболее характерные свойства моделируемого явления.

Модель и отображаемый при ее помощи объект находятся в отношении сходства, а не тождества. Только при этом имеет смысл моделирование, позволяющее выделять наиболее существенные связи и отношения легче и доступнее, чем при действиях с оригиналом. Чрезмерные различия между моделью и оригиналом могут лишать модель ее познавательного значения. Таким образом, из возможных полного, неполного или приближенного моделирования следует выбрать метод, наилучшим образом отвечающий задачам и возможностям исследователя.

При получении на основе моделирования характеристик явлений необходимо учитывать факторы, обуславливающие расхождение результатов, получаемых в модели и оригинале:

- а) погрешности определения отдельных параметров, входящих в критерии подобия;
- б) неточности исходного математического описания явления;
- в) погрешности в получении критериев подобия за счет заведомо упрощенного представления явления при его изучении;
- г) случайные отклонения параметров оригинала и модели от приня-



- тых (расчетных);
- д) погрешности проведения опытов, отклонение фактических параметров режима от расчетных и т.д.;
  - е) погрешности обработки результатов моделирования.

## 2. МЕТОДЫ АНАЛОГОВОГО МАШИННОГО МОДЕЛИРОВАНИЯ

Моделирование является важным этапом проектирования электронной аппаратуры. Оно позволяет провести исследования сложных систем, которые не могут быть выполнены традиционными методами. Моделирование существенно снижает сроки и стоимость проектирования и за счет анализа большого количества вариантов повышает эффективность разрабатываемой системы. Различают физическое, математическое и полунатурное моделирование на ЭВМ.

Физическое моделирование основано на аналогии физических процессов, протекающих в системе, электрическим процессам в машинной модели системы.

При математическом моделировании в ЭВМ реализуются математические зависимости, описывающие процессы, протекающие в моделируемой системе.

При полунатурном моделировании в состав моделирующей установки включаются блоки или устройства реальной системы, что позволяет наиболее полно учесть влияние их характеристик на свойства системы в целом.

При изучении процессов в реальных системах методами моделирования различают три типа времени:

1. **Реальное время**, т.е. время, в течение которого процессы протекают в реальных системах.

2. **Системное время** - время, в течение которого процессы протекают в модели системы. При математическом моделировании это время представляется в виде отдельной переменной  $t$ . Использование системного времени позволяет анализировать процессы в моделях в ускоренном или замедленном темпе, а также останавливать

время для изучения процессов в какой-либо момент времени  $t = t_1$ .

3. **Машинное время** - время, в течение которого происходит моделирование в ЭВМ. Как правило, это время, регистрируемое таймером ЭВМ.

Для моделирования применяют цифровые, аналоговые и гибридные (аналого-цифровые) ЭВМ.

## 2.1. Принципы построения аналоговых вычислительных машин

Аналоговые вычислительные машины (АВМ) обладают рядом характерных свойств, отличающих их от цифровых и позволяющих реализовать методы моделирования, программирование которых на ЦВМ является проблематичным. Основными особенностями АВМ являются:

- высокое быстродействие;
- работа в реальном масштабе времени;
- простота программирования;
- возможность подключения реальной аппаратуры;
- возможность наблюдения протекающих процессов;
- низкая точность решения (погрешность моделирования составляет от 0.1 до 10 %).

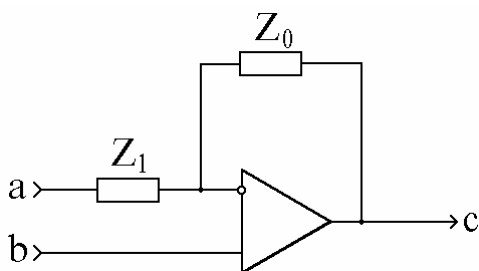


Рис. 2.1. Операционный усилитель

Основным элементом АВМ является операционный усилитель (ОУ) с обратной связью (рис. 2.1), имеющий два входа, называемых инвертирующим (а) и неинвертирующим (b). Выходной сигнал ОУ определяется разностью входных сигналов  $c = k(b - a)$ , где  $k$  - коэффициент усиления ОУ. Если сопротивления  $Z_1$  и  $Z_0$  зависят от частоты, то передаточная функция ОУ имеет вид

Если сопротивления  $Z_1$  и  $Z_0$  зависят от частоты, то передаточная функция ОУ имеет вид

$$W(p) = -\frac{Z_0(p)}{Z_1(p)}.$$

Типовыми элементами (блоками) АВМ являются:

1. Инвертирующий усилитель (рис. 2.2) с передаточной функцией

$$W(p) = -\frac{R_0}{R_1} = k.$$

Данный элемент АВМ выполняет математическую операцию умножения сигнала  $x$  на константу  $k$  (т.н. безинерционное звено):

$$y(t) = k \cdot x(t).$$

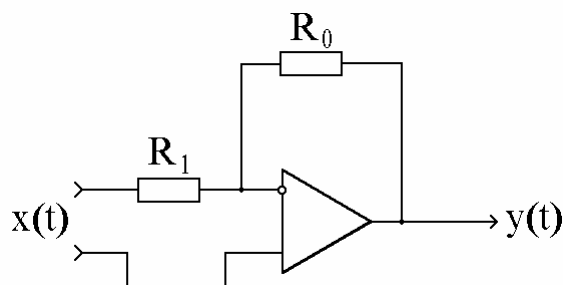


Рис. 2.2. Инвертирующий усилитель

2. Неинвертирующий усилитель (рис. 2.3) имеет передаточную функцию

$$W(p) = \frac{R_0}{R_1 + 1} = k$$

В математическом смысле аналогичен инвертирующему при  $k > 0$ .

3. Сумматор (рис. 2.4) одновременно реализует операции сложения нескольких сигналов  $U_i$  с умножением их на весовые коэффициенты  $k_i$ :

$$U_{\text{ВЫХ}} = \sum k_i \cdot U_i, \quad k_i = -\frac{R_0}{R_i}.$$

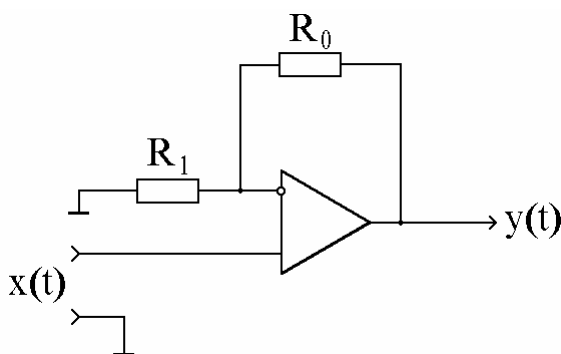


Рис. 2.3. Неинвертирующий усилитель

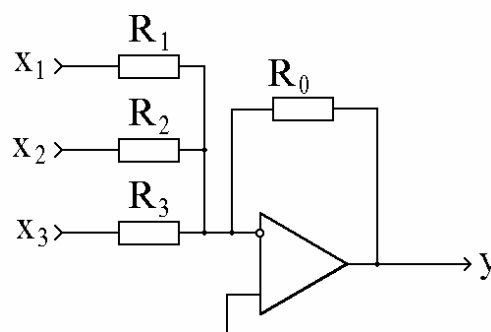


Рис. 2.4. Инвертирующий сумматор

Аналогично может быть построен неинвертирующий сумматор (при подаче входных напряжений на неинвертирующий вход) и сумматор общего вида (рис. 2.5.), реализующий как сложение, так и вычитание сигналов.

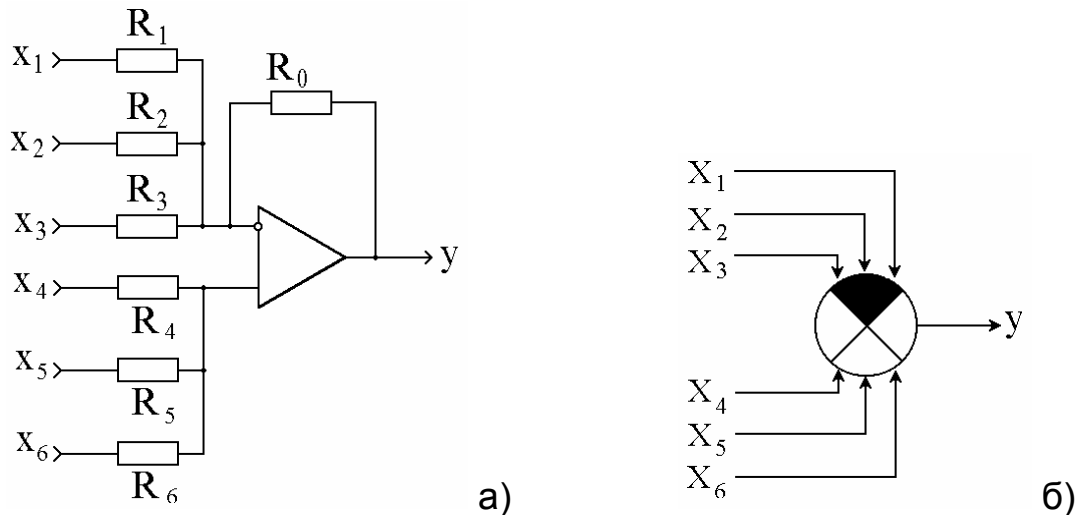


Рис. 2.5. Сумматор

а) схема электрическая; б) обозначение на структурных схемах

4. Интегратор на основе операционного усилителя (рис. 2.6) реализует передаточную функцию

$$W(p) = -\frac{k}{p},$$

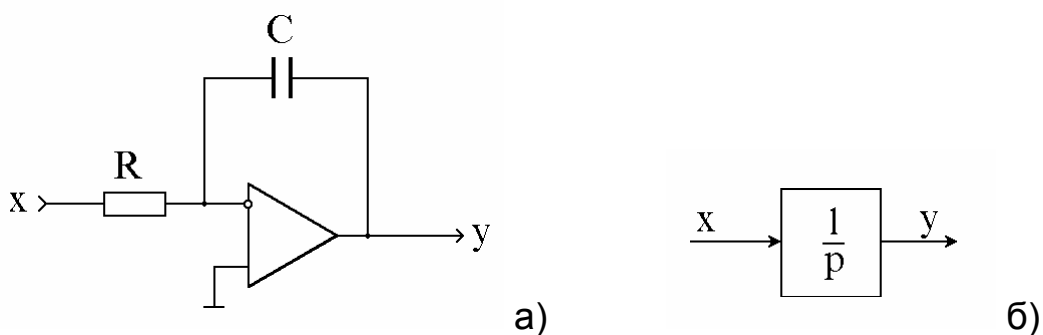


Рис. 2.6. Интегратор

а) схема электрическая б) обозначение на структурных схемах

где  $k = \frac{1}{T} = \frac{1}{R \cdot C}$  ( $T$  - т.н. постоянная времени интегратора). Данная передаточная функция во временной области соответствует интегри-

рованию сигнала

$$y(t) = \int_0^t x(\tau) d\tau + y(0),$$

где  $y(0)$  – т.н. начальные условия интегрирующего звена (рис. 2.7).

5. Пропорционально - интегрирующее звено (инерционное звено, апериодическое) (рис. 2.8) имеет передаточную функцию вида

$$W(p) = -\frac{k}{1 + p \cdot T},$$

где  $k = \frac{R}{R_1}$ ,  $T = R \cdot C$ .

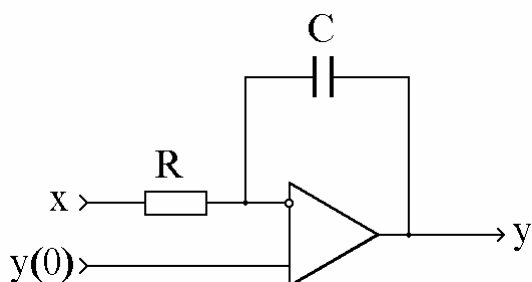


Рис. 2.7. Интегратор с начальными условиями

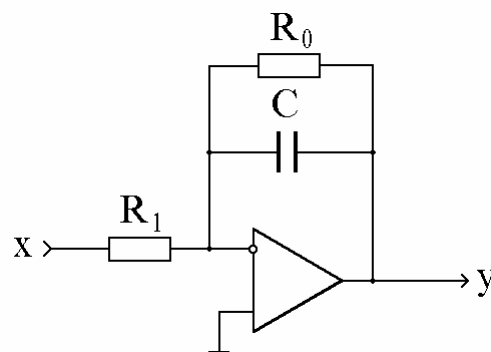


Рис. 2.8. Пропорционально - интегрирующее звено

6. Колебательное звено по своим частотным свойствам эквивалентно резонансному усилителю с колебательным контуром, однако в АВМ данный блок реализуется в виде активного фильтра на ОУ без использования индуктивных элементов (рис. 2.9).

Передаточная функция при этом имеет вид

$$W(p) = \frac{-k}{T^2 \cdot p^2 + 2 \cdot \xi \cdot T \cdot p + 1},$$

где  $k = \frac{R_0}{R_1}$ ,  $T^2 = R_1 \cdot R_0 \cdot C^2$ ,  $\xi = \sqrt{\frac{R_1}{R_0}}$ .

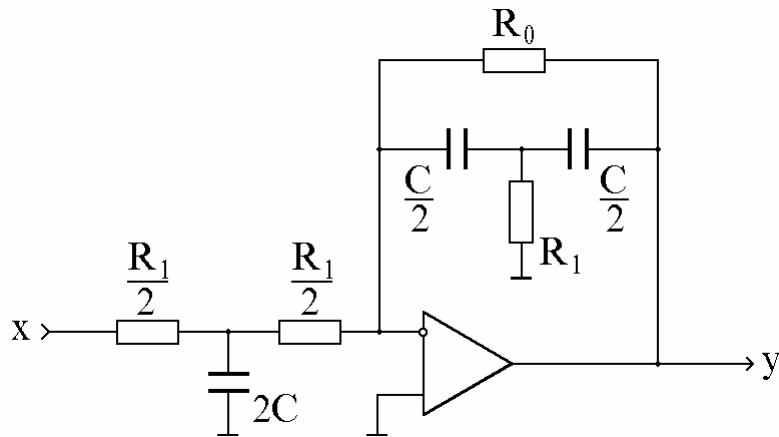


Рис. 2.9. Колебательное звено

7. Логарифмический усилитель входит в структуру многих серийно выпускаемых аналоговых вычислительных машин и реализует функцию преобразования сигнала  $U_{\text{ВЫХ}} = k \cdot \lg(U_{\text{ВХ}})$ , т.е. является нелинейным устройством. Для построения схем с логарифмической амплитудной характеристикой используется зависимость падения напряжения  $U$  на р-п-переходе полупроводникового прибора от протекающего через него тока  $I$ , которая в области малых токов имеет вид  $U \approx N \cdot \lg(I/I_{\text{ОБР}})$ .

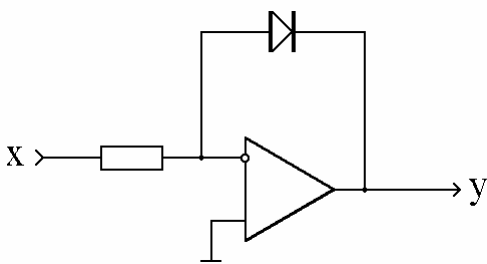


Рис. 2.10.

При использовании полупроводникового диода в цепи обратной связи операционного усилителя (рис. 2.10) характеристика усилителя за счет нелинейной обратной связи будет иметь логарифмический характер. Естественно, что такой способ имеет значительные погрешности, вызванные неидеальностью харак-

теристики р-п-перехода, нестабильностью обратного тока  $I_{\text{ОБР}}$  и т.д. Практические схемы отличаются от представленной использованием в цепи обратной связи дифференциальных пар подобранных транзисторов, наличием цепей термокомпенсации и расширения динамического диапазона.

8. Антилогарифмический усилитель (экспоненциатор) реализует нелинейное преобразование сигнала вида  $U_{ВЫХ} = k \cdot 10^{U_{ВХ}}$  и аналогичен по принципу построения логарифмическому, только нелинейный элемент используется во входной цепи операционного усилителя (рис. 2.11).

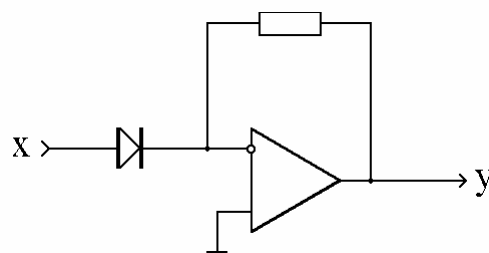


Рис. 2.11. Экспоненциатор

9. Нелинейный преобразователь общего вида используется в АВМ для формирования различных нелинейных функций и с помощью ручек регулировки позволяет осуществлять ступенчатую аппроксимацию однозначной нелинейности. Использование обратных связей, встроенных диодов, а также различных точек съема выходного сигнала позволяет моделировать широкий класс однозначных, двузначных и петлевых нелинейностей.

10. Перемножитель сигналов выполняет математическую операцию умножения аналоговых сигналов

$$U_{ВЫХ}(t) = U_1(t) \cdot U_2(t) \cdot k$$

(рис. 2.12-а) и реализуется, как правило, на основе интегральных микросхем – перемножителей, например, 525ПС2, 174ПС1, AD534 и др. Коэффициент  $k$  для указанных микросхем равен  $1/10$ .

11. На основе интегральных перемножителей в цепи обратной связи операционных усилителей строятся операционные блоки, выполняющие операцию аппаратного деления аналоговых сигналов (рис. 2.12-б)

$$U_{ВЫХ}(t) = k \cdot \frac{U_1(t)}{U_2(t)}$$

Моделирование на АВМ выполняется

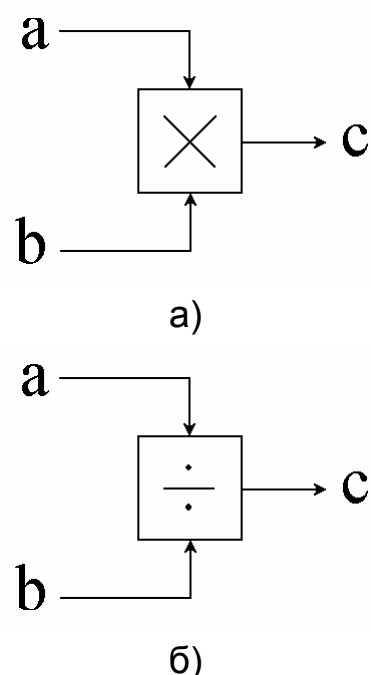


Рис. 2.12. Условные графические обозначения операционных блоков:

а) перемножения сигналов  $c = a \times b$ ;  
б) деления сигналов  $c = a / b$ .

путем соединения имеющихся в наличии блоков (1-11) согласно структурной схеме или дифференциальному уравнению, приведенных к машинному виду путем введения так называемых машинных переменных.

## 2.2. Машинные переменные.

При моделировании реальные переменные моделируемой системы, соответствующие сигналам и воздействиям (например, частота, температура, скорость, давление и т.д.) отображаются в АВМ напряжениями, которые называют машинными переменными.

Соответствие между реальными и машинными переменными устанавливается с помощью масштабных коэффициентов, выбираемых из условия:

$$m_x \leq \frac{u_m}{|x_m|},$$

где  $u_m$  - максимальное напряжение на выходе усилителей АВМ (как правило,  $\pm 10 \dots 15$  В);  $|x_m|$  - максимальное значение моделируемой переменной.

Масштабные коэффициенты - размерные величины. Так, если моделируемый процесс  $x(t)$  - температура в системе, то размерность масштабного коэффициента - В/град.

Если требуется замедлить или ускорить процессы в модели по сравнению с процессами в реальной системе, то вводят масштабный коэффициент времени  $m_t$ :

$$m_t = \frac{t_M}{t},$$

где  $t_m$  - системное время,  $t$  - реальное время. При этом все постоянные времени  $T$  и константы передачи в дифференцирующих и интегрирующих звеньях  $k$  пересчитывают по формулам:

$$T_M = m_t \cdot T;$$



для дифференцирующих звеньев

$$k_M = m_t \cdot k;$$

для интегрирующих звеньев

$$k_M = \frac{k}{m_t}.$$

Необходимо также учитывать собственные масштабные коэффициенты блоков АВМ. Например, для блоков умножения и деления сигналов (рис. 2.12), реализованных на ИМС 174ПС1:

$$c = \frac{a \cdot b}{10}, \quad \text{и} \quad c = \frac{a}{10 \cdot b}.$$

### 2.3. Методика моделирования на АВМ

Характеристики реальных устройств, строго говоря, практически всегда нелинейны. На практике реальные устройства зачастую идеализируют, представляя их в виде линейных динамических систем с целью упрощения анализа. Однако в ряде случаев приходится учитывать нелинейный характер реальных устройств. Путем эквивалентных преобразований структурной схемы нелинейную систему можно разделить на нелинейную безинерционную часть и линейную динамическую подсистему. Характеристику нелинейной части системы можно аппроксимировать с помощью нелинейных блоков АВМ путем их соответствующей настройки, последовательного или (и) параллельного соединения.

Линейную динамическую часть системы описывают либо дифференциальным уравнением

$$\sum_{i=0}^n a_i \cdot y^{(i)}(t) = \sum_{j=0}^m b_j \cdot x^{(j)}(t), \quad (2.1)$$

либо соответствующей ему передаточной функцией

$$W(p) = \frac{\sum_{j=0}^m b_j \cdot p^j}{\sum_{i=0}^n a_i \cdot p^i} = \frac{Y(p)}{X(p)}. \quad (2.2)$$

Для физически осуществимых объектов  $m \leq n$  [1].

Уравнению (2.1) моделируемой системы соответствует по структуре машинное уравнение

$$\sum_{i=0}^n a_{Mi} \cdot \frac{\partial^i U_y}{\partial t_M^i} = \sum_{j=0}^m b_{Mj} \cdot \frac{\partial^j U_x}{\partial t_M^j}, \quad (2.3)$$

где  $a_{Mi}$ ,  $b_{Mj}$  – машинные коэффициенты;  $U_y$ ,  $U_x$  – машинные переменные соответствующие переменным  $x$ ,  $y$ .

Значения коэффициентов  $a_{Mi}$ ,  $b_{Mj}$  определяют из условия подобия уравнений, для чего в выражение (2.1) подставляют значения

$$y = \frac{U_y}{m_y}, \quad x = \frac{U_x}{m_x}, \quad t = \frac{t_M}{m_t};$$

$$\sum_{i=0}^n a_i \frac{d^i U_y \cdot m_t^i}{m_y \cdot dt_M^i} = \sum_{j=0}^m b_j \frac{d^j U_x \cdot m_t^j}{m_x \cdot dt_M^j}, \quad (2.4)$$

Сравнив уравнения (2.1) и (2.4), получим

$$a_{Mi} = a_i \frac{m_t^i}{m_y}, \quad b_{Mj} = b_j \frac{m_t^j}{m_x}. \quad (2.5)$$

С учетом масштабирования процедура моделирования системы включает в себя следующие этапы:

- 1) составление уравнения системы;
- 2) введение масштабов переменных;
- 3) определение машинных коэффициентов  $a_{Mi}$ ,  $b_{Mj}$ ;

- 4) составление схемы набора с машинными коэффициентами;
- 5) исследование модели и получение  $U_y(t_M)$ ;
- 6) переход к действительной переменной

$$y = \frac{U_y}{m_y}, \quad t = \frac{t_M}{m_t}.$$

## 2.4. Пример решения дифференциальных уравнений на АВМ

Пусть задано дифференциальное уравнение системы

$$a \cdot y''(t) + b \cdot y'(t) + c \cdot y(t) = d \cdot x'(t) + e \cdot x(t).$$

Перейдем к изображению по Лапласу [1, 2]:

$$a \cdot p^2 \cdot Y(p) + b \cdot p \cdot Y(p) + c \cdot Y(p) = d \cdot p \cdot X(p) + e \cdot X(p).$$

Вынесем  $X(p)$  и  $Y(p)$ :

$$Y(p) \cdot (a \cdot p^2 + b \cdot p + c) = X(p) \cdot (d \cdot p + e).$$

Передаточная функция модели системы:

$$W(p) = \frac{Y(p)}{X(p)} = \frac{d \cdot p + e}{a \cdot p^2 + b \cdot p + c}.$$

Введем переменную состояния  $Z$

$$\frac{Y(p)}{d \cdot p + e} = \frac{X(p)}{a \cdot p^2 + b \cdot p + c} = Z(p)$$

и представим дифференциальное уравнение в виде системы уравнений:

$$\begin{cases} Y(p) = Z(p) \cdot (d \cdot p + e) \\ X(p) = Z(p) \cdot (a \cdot p^2 + b \cdot p + c) \end{cases}.$$

Перейдем к системе дифференциальных уравнений в переменных состояния:

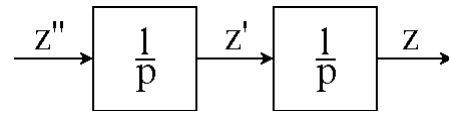
$$\begin{cases} y(t) = d \cdot z' + z \cdot e \\ x(t) = a \cdot z'' + b \cdot z' + c \cdot z \end{cases}.$$

Выделим старшую производную  $z''$ :

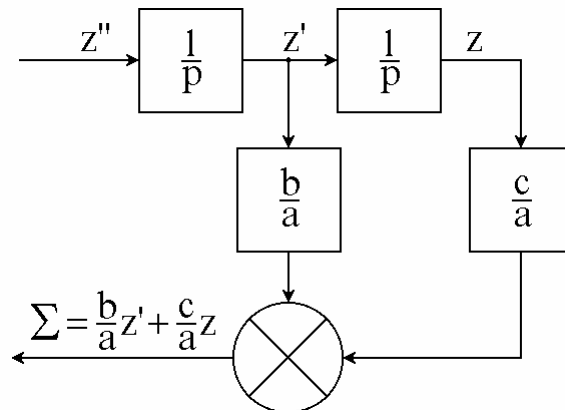
$$z'' = \frac{x}{a} - \left( \frac{b}{a} z' + \frac{c}{a} z \right).$$

Разрабатываем структурную схему моделирующей установки:

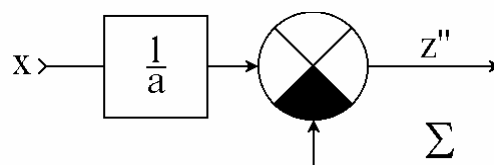
1) Считаем, что  $z''$  известно. Интегрированием получим  $z'$  и  $z$ :



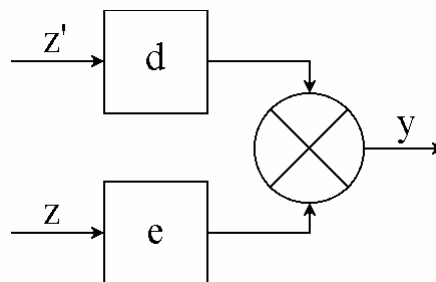
2) Находим сумму  $\frac{b}{a} z' + \frac{c}{a} z$ :



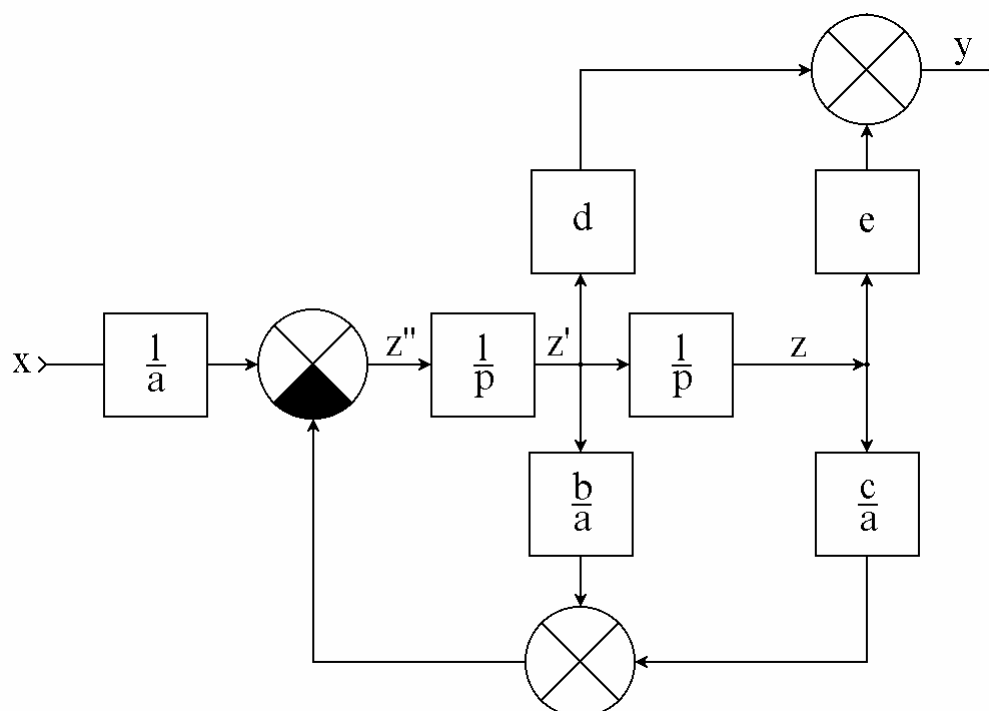
3) Поскольку  $Z'' = \frac{x}{a} - \left( \frac{b}{a} Z' + \frac{c}{a} Z \right)$ , то можем найти  $Z''$ :



4) По уравнению  $y = Z' \cdot d + Z \cdot e$  формируем  $y$ :



Общая схема моделирования:



Для реализации моделирующей установки используются сумматоры, интеграторы и усилители.

## 2.5. Моделирование детерминированных сигналов на АВМ

При моделировании детерминированных сигналов используются источники эталонных напряжений и встроенные генераторы сигналов специальной формы (синусоидальных, треугольных, прямоугольных и т.д.).

Широко используемая в математических выражениях единичная функция  $1(t)$  моделируется как постоянное напряжение, появляющееся в момент включения АВМ, либо с помощью коммутатора напряжений.

Если требующийся сигнал или воздействие может быть представлено в виде полинома  $x(t) = a_0 \cdot 1(t) + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 + \dots$ , то используется цепочка интеграторов и усилителей, как показано на рис. 2.13.

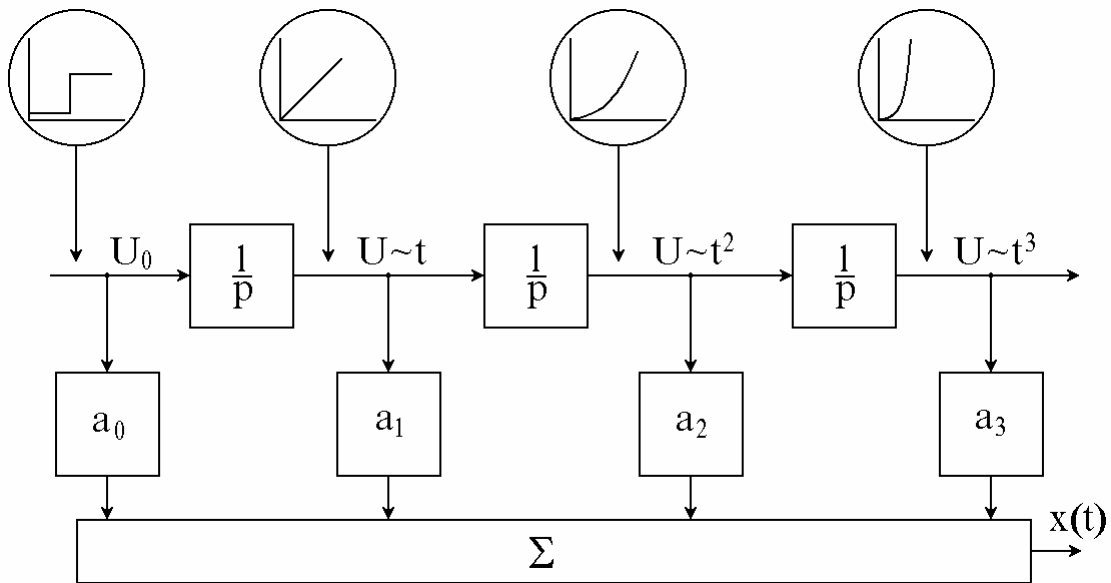


Рис. 2.13. Генератор сигнала, заданного в виде полинома

Если моделируемый сигнал описывается аналитической зависимостью вида  $x = f(t)$ , то его можно рассматривать как результат решения дифференциального уравнения. Например, для экспоненциального сигнала вида

$$x(t) = k \cdot e^{-\alpha \cdot t}$$

производная

$$\frac{\partial x}{\partial t} = -k \cdot \alpha \cdot \exp(-\alpha \cdot t),$$

отсюда искомое дифференциальное уравнение имеет вид

$$\frac{\partial x}{\partial t} = -\alpha \cdot x(t).$$

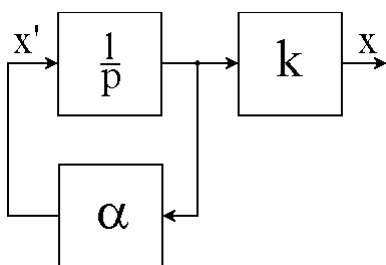


Рис. 2.14. Схема генерирования экспоненциального сигнала

Схема моделирования этого уравнения на АВМ показана на рис. 2.14.

В ряде случаев для получения алгоритма моделирования требуется многократное дифференцирование. Например, для аналитического сигнала вида

$$x(t) = \sin(\omega \cdot t)$$

после первого дифференцирования по  $t$  получим

$$\frac{\partial x}{\partial t} = \omega \cdot \cos(\omega \cdot t).$$

Второе дифференцирование дает искомый результат:

$$\frac{\partial^2 x}{\partial t^2} = -\omega^2 \cdot \sin(\omega \cdot t),$$

откуда следует, что

$$x(t) + \omega^2 \cdot x''(t) = 0.$$

Решая дифференциальное уравнение, получаем

$$x'' = -\omega^2 \cdot x(t).$$

Схема моделирования данного уравнения на АВМ приведена на рис. 2.15. Она позволяет генерировать гармонический сигнал с частотой, регулируемой за счет изменения коэффициента усиления  $k = -\omega^2$ .

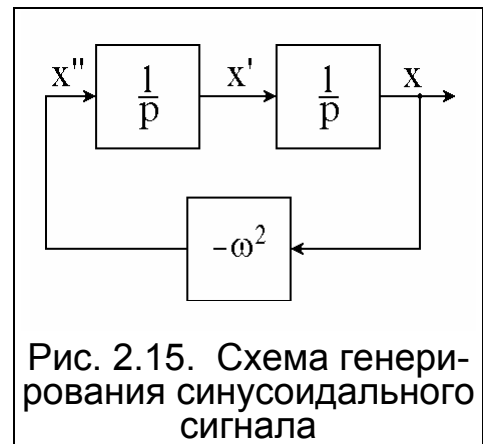


Рис. 2.15. Схема генерирования синусоидального сигнала

## 2.6. Моделирование случайных процессов на АВМ

Случайный процесс  $x(t)$  характеризуется плотностью распределения  $f(x)$  (в общем случае многомерной) и спектральной плотностью мощности  $S_X^2(j\omega)$  (или корреляционной функцией  $R_X(\tau)$ ) [5].

Если сигнал на входе линейной динамической системы является некоррелированным случайным процессом  $R_X(\tau) = \delta(\tau)$ , то его спектральная плотность является равномерной в диапазоне частот  $0 \dots \infty$ , [1,5] т.е.

$$S_X^2(\omega) = N_0,$$

где  $N_0$  - энергия шума. Такой случайный процесс называют "белым

шумом".

Поскольку спектр выходного процесса линейной динамической системы определяется как [1,3]

$$S_Y(j\omega) = K(j\omega) \cdot S_X(j\omega),$$

то спектральная плотность мощности случайного процесса на выходе линейной динамической системы

$$\begin{aligned} S_Y^2(\omega) &= S_Y(j\omega) \cdot S_Y^*(j\omega) = \\ &= K(j\omega) \cdot S_X(j\omega) \cdot K^*(j\omega) \cdot S_X^*(j\omega) =, \\ &= K^2(\omega) \cdot S_X^2(\omega) \end{aligned}$$

т.е. спектральная плотность выходного случайного процесса определяется как произведение спектральной плотности входного случайного процесса на квадрат амплитудно-частотной характеристики линейной динамической системы

$$S_Y^2(\omega) = N_0 \cdot A^2(\omega), \quad (2.6)$$

а корреляционная функция

$$\begin{aligned} R(\tau) &= F^{-1}\{S^2(\omega)\}, \\ S^2(\omega) &= F\{R(\tau)\}, \end{aligned} \quad (2.7)$$

где  $F(\bullet)$  – прямое, а  $F^{-1}(\bullet)$  – обратное преобразование Фурье.

Данные соотношения позволяют определить метод получения случайных процессов с заданной корреляционной функцией  $R_Y(\tau)$  или спектральной плотностью  $S_Y^2(\omega) = F(R_Y(\tau))$  методом формирующего фильтра.

В качестве источника "белого шума" в аналоговых вычислительных машинах используются либо шумовые диоды, либо стабилитроны. Простейший генератор шума на операционном усилителе и двух стабилитронах показан на рис. 2.16. Белый шум пропускается через формирующий фильтр с передаточной функцией

$$W(p) = S_Y(p),$$



которая определяется из соотношения (2.6), если задана спектральная плотность случайного процесса, либо из (2.7), если задана требуемая корреляционная функция.

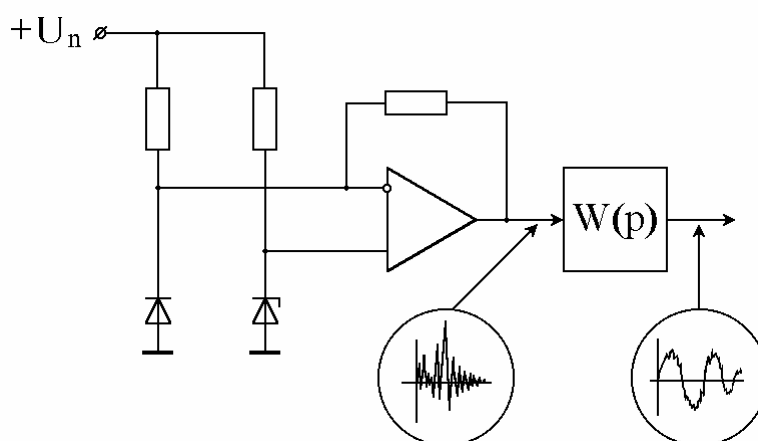


Рис. 2.16. Схема моделирования случайного процесса

### 3. МЕТОДЫ ЦИФРОВОГО МОДЕЛИРОВАНИЯ НА ЭВМ

#### 3.1. Основные принципы цифрового моделирования

Одним из важнейших инструментов автоматизированного проектирования является цифровое моделирование на ЭВМ проектируемых объектов, представляемых в виде математических моделей. Математическая модель - система математических конструкций (чисел, переменных, матриц, множеств и т.д.) и отношений между ними, которые адекватно отражают существенные с инженерных позиций свойства моделируемого объекта или явления.

Особенностью цифрового моделирования на ЭВМ является то, что все непрерывные функции (времени, частоты, температуры и т.д.)  $x(t)$  представляют вектором (массивом) их дискретных отсчетов  $x_i$  (рис. 3.1-а), взятых через равные промежутки времени  $\Delta t$  (либо другого параметра) и определяющих (согласно теореме Котельникова [1,3]) граничную частоту спектра сигнала, однозначно восстанавливаемую по отсчетам  $x_i$  (рис. 3.1-б):

$$F_{MAX} = \frac{1}{2 \cdot \Delta t}. \quad (3.1)$$

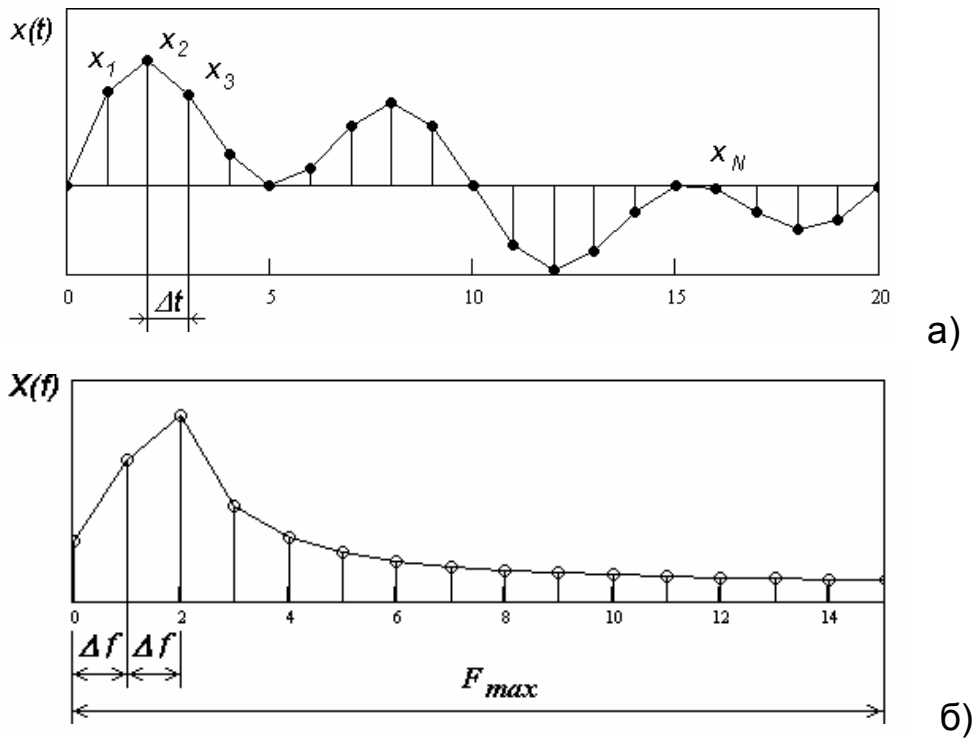


Рис. 3.1. Дискретное представление непрерывного процесса (а) и его спектра (б)

Частота дискретизации при этом

$$f_D = 2 \cdot F_{MAX}. \quad (3.2)$$

Количество отсчетов  $N$  (объем выборки) связано с необходимым временем наблюдения  $T$  за процессом

$$N = \frac{T}{\Delta t}. \quad (3.3)$$

Для анализа спектрального состава процессов используют, как правило, процедуру быстрого преобразования Фурье (БПФ) [3], реализующую преобразование Фурье для дискретных отсчетов процесса  $x_i$

$$X(f) = \int_0^T x(t) \cdot \exp(-j \cdot 2 \cdot \pi \cdot f \cdot t) dt.$$

В результате применения процедуры БПФ получается массив отсчетов  $X_k$ , являющийся дискретным представлением спектра сигнала с

равномерным дискретным шагом по частоте  $\Delta f$  (рис. 3.1-б), который однозначно связан со временем наблюдения за процессом  $T$  как

$$\Delta f = \frac{1}{T}. \quad (3.4)$$

Особенностью процедуры БПФ является требование к объему выборки (3.3)  $N = 2^M$ , где  $M$  – целое число (см. приложение 1).

### 3.2. Моделирование детерминированных сигналов и воздействий на ЦВМ

Все наблюдаемые процессы, сигналы и воздействия, влияющие на электронную аппаратуру, в самом общем виде можно разделить на детерминированные и случайные. К детерминированным относят процессы, которые могут быть описаны математическими соотношениями либо заданы таблично. Процессы, описывающие детерминированные явления, могут быть периодическими и непериодическими [6].

Для моделирования периодических и почти периодических сигналов используется разложение процесса  $x(t)$  в ряд Фурье

$$x(t) = \sum_{k=-\infty}^{\infty} C_k \cdot e^{jk\omega_1 t}, \quad (3.5)$$

$$C_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \cdot e^{-jk\omega t} dt. \quad (3.6)$$

Т.е. между временным представлением периодических сигналов и их частотными спектрами существует взаимно однозначное соответствие. Поэтому, при решении задач моделирования часто заменяют операции над процессами операциями над их частотными спектрами.

Важное отличие переходных процессов от периодических и почти периодических состоит в том, что их невозможно представить с помощью дискретного спектра. Такие сигналы, как и случайные процес-

сы, описываются непрерывным спектром или спектральной плотностью [1, 3]:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j\omega t} dt, \quad (3.7)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) \cdot e^{j\omega t} d\omega. \quad (3.8)$$

Данные формулы являются прямым (3.7) и обратным (3.8) преобразованиями Фурье и описывают непрерывный сигнал  $x(t)$ , заданный на интервале  $t \in [-\infty, \infty]$ , соответственно, в частотной и временной форме. Интегральное преобразование Фурье существует не для всех сигналов, а только для так называемых абсолютно интегрируемых, для которых

$$\int_{-\infty}^{\infty} |x(t)| dt < \infty.$$

При моделировании детерминированных сигналов на цифровых ЭВМ возможно два подхода:

- моделирование во временной области, когда сигнал задается в виде функции времени  $x(t) = \varphi(t)$ ;
- моделирование в частотной области, когда сигнал формируется как детерминированная функция частоты  $X(f) = \varphi(f)$ .

При моделировании сигнала  $x(t) = \varphi(t)$  принципиально важно определить время моделирования  $T$ , поскольку разрешающая способность по частоте процедуры БПФ (рис. 3.1) зависит от времени наблюдения сигнала:

$$\Delta f = 1/T.$$

При этом необходимо иметь ввиду, что процедура БПФ ориентирована на обработку периодических сигналов, т.е. вычисленный таким способом спектр сигнала может отличаться от истинного вследствие периодического продолжения сигнала при вычислении спектра. Этот

эффект показан на рис. 3.2.

Максимально-возможная частота в спектре сигнала  $F$  определяет минимальную частоту дискретизации

$$f'_D = 2 \cdot F.$$

Однако на практике следует принимать частоту дискретизации в 5...10 раз выше, чтобы получить 10...20 отсчетов сигнала на период. Это не только облегчает визуализацию, но и снижает погрешность моделирования [6].

Интервал дискретизации (интервал времени между отсчетами – см. рис. 3.1) однозначно связан с частотой дискретизации:

$$\Delta t' = \frac{1}{f'_D}.$$

Тогда необходимое количество отсчетов (объем выборки)

$$N' = T / \Delta t'.$$

При использовании в дальнейшем процедуры БПФ необходимо полученное значение  $N'$  “округлить вверх” до ближайшего значения, кратного  $2^m$ :

$$N = \text{Round}(N') \cdot 2^m.$$

Например:  $N' = 500 \Rightarrow N = 512$ ,  $N' = 900 \Rightarrow N = 1024$ .

Далее уточняем интервал дискретизации

$$\Delta t = T / N,$$

частоту дискретизации

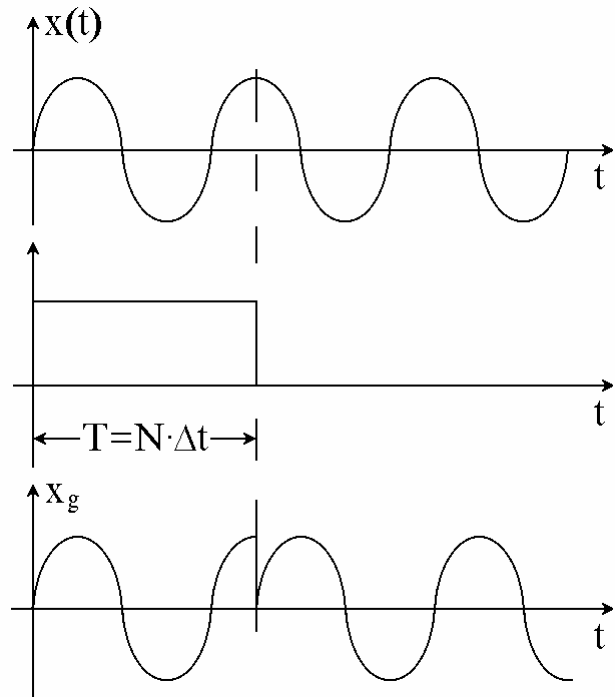


Рис. 3.2. Периодическое продолжение  $x_g$  сигнала  $x(t)$

$$f_D = 1/\Delta t$$

и формируем массив дискретного времени

$$t_i = i \cdot \Delta t,$$

где  $i = 0 \dots N - 1$  или  $i = 1 \dots N$  в зависимости от используемых программных средств.

Теперь можно выполнять собственно моделирование сигнала:

$$x_i = \varphi(t_i).$$

При спектральном методе формирования сигнала после определения времени наблюдения сигнала  $T$  и объема выборки  $N$  необходимо определить количество отсчетов в дискретном спектре сигнала

$$M = N/2,$$

поскольку при использовании БПФ только половина отсчетов спектра содержит полезную информацию. Вторую половину массива из  $N$  отсчетов некоторые процедуры БПФ не возвращают вовсе, а некоторые возвращают комплексно-сопряженные зеркально расположенные значения спектра сигнала (см. рис. 3.3 и приложение 2).

Если в качестве исходных данных задана разрешающая способ-

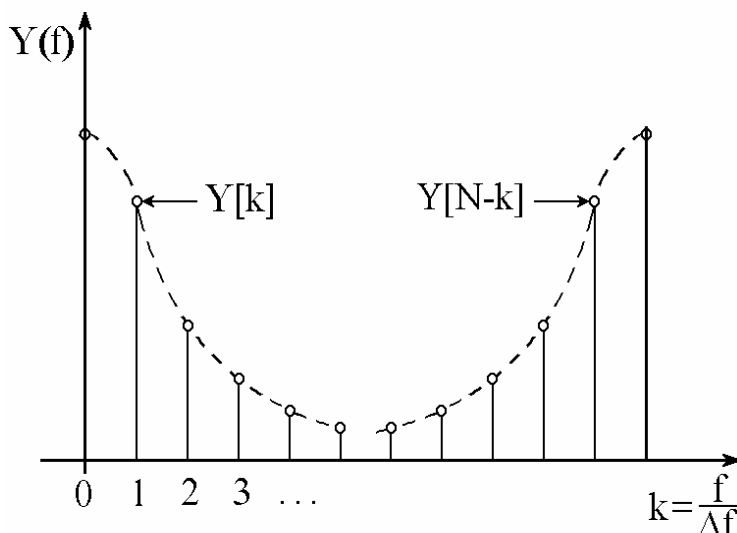


Рис. 3.3. Зеркальное расположение копии спектра в массиве

ность по частоте  $\Delta f$  (например, при моделировании сигналов со сложными видами модуляции), то необходимое время моделирования можно определить как

$$T = 1/\Delta f,$$

а объем выборки будет зависеть от ширины спектра сигнала  $F_{MAX}$ :

$$M = F_{MAX} / \Delta f.$$

Сетку частот определяют по времени наблюдения процесса как

$$f_j = j \cdot \Delta f = j / T,$$

где  $j = 0 \dots M - 1$ , либо  $j = 1 \dots M$  в зависимости от используемого программного обеспечения. Например, при использовании процедуры  $fft(\bullet)$  программы MathCAD переменную цикла по частоте следует принимать  $j = 0 \dots M$ .

Теперь может быть выполнено формирование спектра сигнала

$$X_j = \varphi(f_j)$$

и после обратного преобразования Фурье

$$x(t) = F^{-1}\{X(f)\}$$

будет получен требующийся сигнал  $x(t)$ .

Моменты времени  $t_i$ , для которых получены отсчеты сигнала  $x_i = x(t_i)$ , определяются объемом выборки и временем наблюдения как:

$$N = 2 \cdot M,$$

$$\Delta t = T / N,$$

$$t_i = i \cdot \Delta t,$$

где  $i = 0 \dots N - 1$  или  $i = 1 \dots N$  в зависимости от используемых программных средств.

### 3.3. Моделирование случайных сигналов и воздействий

При разработке и проектировании электронных средств и технологических процессов их производства необходимо учитывать случайный характер параметров комплектующих электрорадиоэлементов (ЭРЭ), конструктивных размеров, входных сигналов, внешних воздействий и т. д. Выходные параметры электронной аппаратуры  $y_i$  при этом представляют в виде

$$\begin{aligned}
 y_1 &= \phi_1(x_1, \dots, x_n), \\
 y_i &= \phi_i(x_1, \dots, x_n), \\
 y_m &= \phi_m(x_1, \dots, x_n),
 \end{aligned}
 \tag{3.9}$$

где  $x_i$  – случайные и детерминированные первичные параметры (параметры ЭРЭ, размеры, параметры входных сигналов и внешних воздействий и т.п.),  $\phi(\bullet)$  – функция взаимосвязи выходного параметра с первичными (входными).

Анализ соотношений (3.9) показывает, что выходные параметры  $y_i$  также случайны и в общем случае статистически зависимы (как параметры, зависящие от одних и тех же случайных переменных). Нахождение их характеристик чисто аналитическим путем по известным законам распределений первичных параметров является крайне сложной задачей.

Решение задачи можно упростить путем проведения статистического имитационного эксперимента, при котором машинным путем имитируются реализации  $x_{ik}$  случайных первичных параметров  $x_i$ , по ним вычисляют в соответствии с соотношениями (3.9) реализации  $y_{ik}$  выходных параметров  $y_i$ . Статистическая обработка полученных массивов  $y_{ik}$  позволяет определить требуемые статистические характеристики, при этом на вид функций (3.9) не накладывают никаких ограничений. Таким образом, основная трудность при статистическом моделировании состоит в формировании реализаций случайных значений первичных параметров.

### **3.3.1. Имитация значений стандартной равномерно распределенной величины**

Большинство алгоритмов моделирования случайных величин основано на функциональном преобразовании реализаций случайной величины  $x$ , равномерно распределенной в интервале  $[0, 1]$ . Такую величину называют стандартной, она подчиняется закону распреде-



ления с плотностью распределения [4]

$$f(x) = 1. \quad (3.10)$$

Известно, что закону распределения (3.10) подчиняется ошибка округления при приближенных вычислениях. Например,  $5.7 \times 6.2 = 35.34$ . Если результат округлить до 35, т.е. выделить целую часть результата, то ошибка округления 0,34 будет лежать в интервале от 0 до 1. На основе этой идеи строятся алгоритмы генерации случайных чисел с равномерным законом распределения [9].

Для получения реализаций равномерно распределенных в интервале  $[0, 1]$  случайных чисел можно применить мультипликативный алгоритм, состоящий в следующем:

- 1) задают исходное случайное число  $x_0$ , нечетное, принадлежащее интервалу  $[0, 1]$ , не кратное 5, например,  $x_0 = 0.673783$ ;
- 2) умножают  $x_0$  на коэффициент  $k$ , значение которого может быть произвольным, однако, как показали исследования, среди множества двузначных чисел только два из них – 37 и 73 – обеспечивают устойчивую работу алгоритма;
- 3) в полученном значении выделяют дробную часть, которая и есть новое случайное число:

$$x_i = x_{i-1} \cdot k - \text{int}(x_{i-1} \cdot k),$$

где  $\text{int}(\bullet)$  – соответствующая функция выделения целой части числа ( $\text{Trunc}(\bullet)$  в Паскале,  $\text{floor}(\bullet)$  – в системе MathCAD);

- 4) действия, изложенные в пп. 2 и 3 повторяют  $N$  раз, где  $N$  – требуемое количество реализаций равномерно распределенного числа.

В различных системах программирования имеются стандартные процедуры (подпрограммы) генерации реализаций таких чисел. В языке Паскаль это функция  $\text{random}(n)$ , в системе MathCAD такой подпрограммой является функция  $\text{rnd}(n)$ .

### 3.3.2. Случайные события и дискретные случайные величины

#### Моделирование простых случайных событий

Допустим, случайное событие  $A$  может произойти с вероятностью  $P(A) = P$ . Необходимо построить алгоритм формирования события  $A$ .

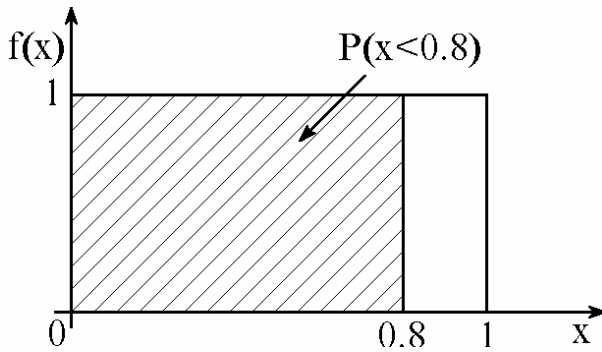


Рис. 3.4. Вероятность события (3.4)

Сформируем равномерно распределенную случайную величину  $x_i$ ,  $i = 1 \dots N$ , где  $N$  — интервал ( $N \cdot \Delta t = T$  — время) ожидания события  $A$ . Например,  $P = 0.8$ . Тогда с учетом (3.10) вероятность того, что  $x_i$  будет меньше значения 0,8 (рис.

$$P(x_i < 0.8) = \int_0^{0.8} f(x) dx = 0.8,$$

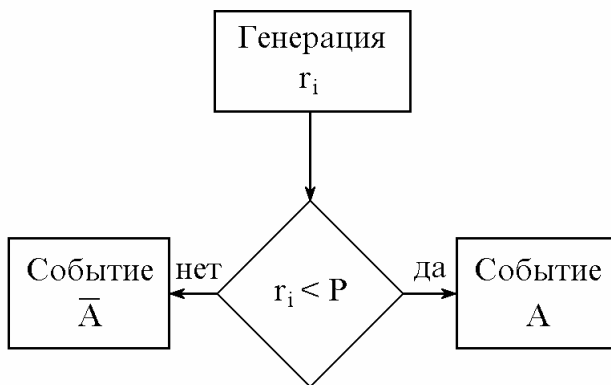


Рис. 3.5. Алгоритм моделирования случайного события

откуда следует алгоритм моделирования случайных событий, представленный на рис. 3.5, где  $r_i$  — случайная величина с равномерным законом распределения (3.10),  $P$  — вероятность события  $A$ .

При  $N$  обращениях к генератору случайных чисел ( $i = 1 \dots N$ ) будет имитировано  $M$  событий.

При этом

$$\lim_{N \rightarrow \infty} \frac{M}{N} = P(A).$$

### Моделирование сложных событий

Допустим, событие  $C$  – сложное событие, заключающееся в том, что одновременно должны произойти два простых независимых события:  $A$  с вероятностью  $P(A)$  и  $B$  с вероятностью  $P(B)$ . При этом возможны четыре события:  $C = A \cap B$ ,  $\bar{A} \cap B$ ,  $A \cap \bar{B}$ ,  $\bar{A} \cap \bar{B}$ . Для моделирования таких событий необходимо последовательно получить два равновероятно распределенных числа  $r_1$  и  $r_2$ , по которым будет определяться требующееся событие согласно алгоритму, приведенному на рис. 3.6.

Если же события  $A$  и  $B$  зависимые, то для моделирования необходимо знать условную вероятность  $P(B|A)$ , которая используется вместо  $P(B)$  при проверке условия  $r_2 < P(B|A)$  [4].

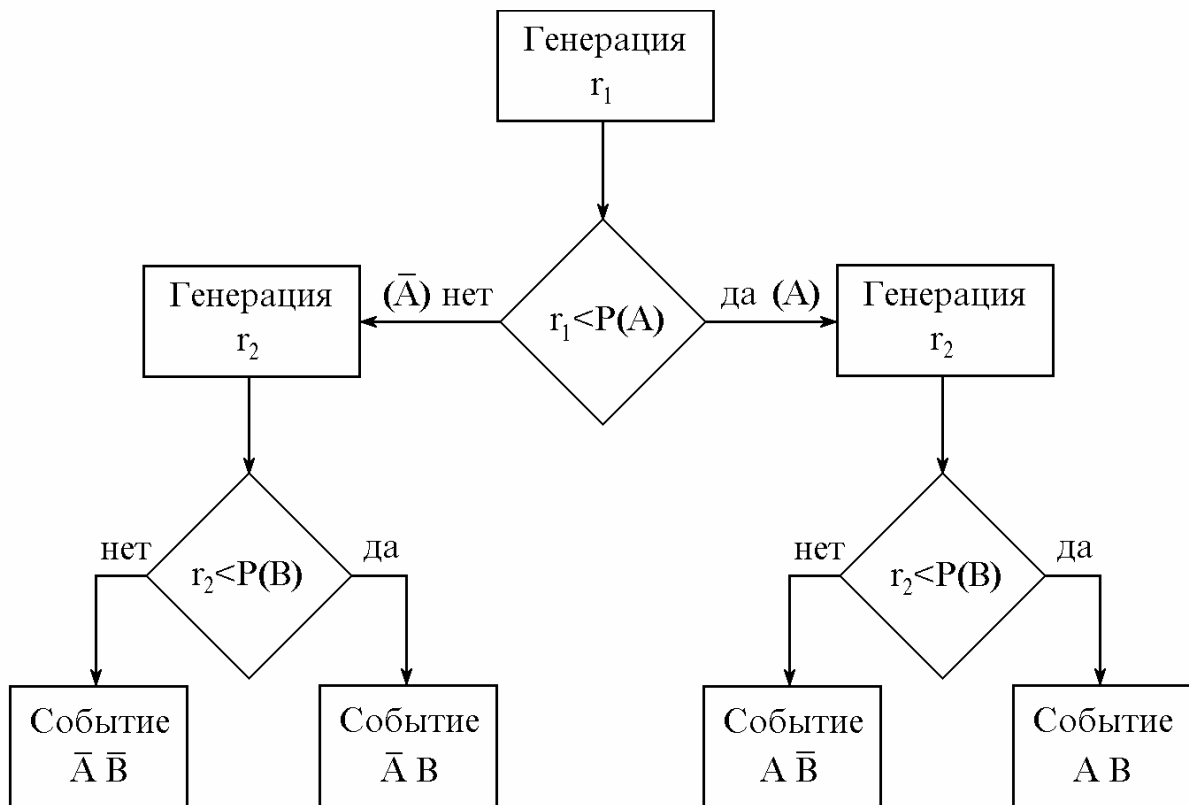


Рис. 3.6. Алгоритм моделирования сложных событий.

### Моделирование реализаций дискретных случайных величин

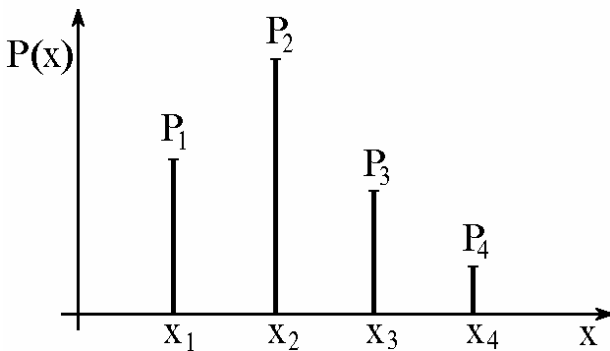


Рис. 3.7. Закон распределения дискретной случайной величины.

Дискретная случайная величина [4] принимает счетное множество дискретных значений  $x_k, k = 1 \dots K$  с заданными вероятностями  $p_k$  (рис. 3.7), причем

$$\sum_{k=1}^K p_k = 1. \quad (3.11)$$

Алгоритм формирования реализаций такой величины строят по

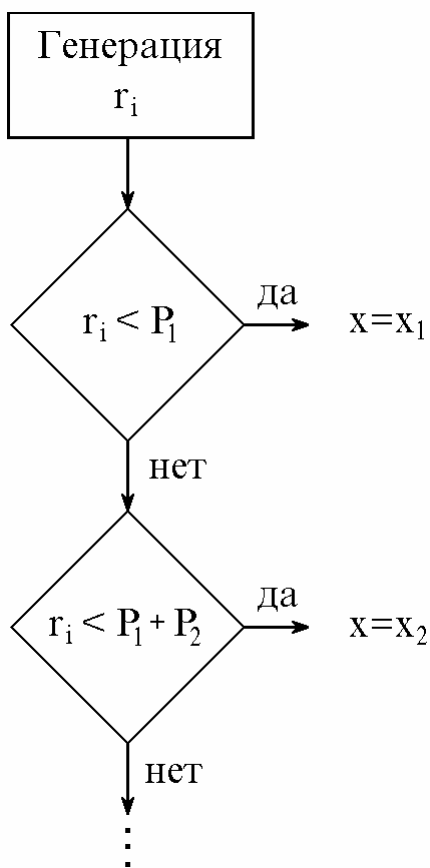


Рис. 3.8. Алгоритм моделирования дискретных случайных величин

принципу обратной функции распределения (рис. 3.7): формируют реализацию равномерно распределенных чисел  $r_i$  и циклически ( $k = 1 \dots K - 1$ ) проверяют, если  $r_i < \sum_{j=1}^k p_j$ , то  $y_i = x_k$ . Алгоритм моделирования дискретной случайной величины представлен на рис. 3.8. Очевидно, что проверка неравенства при  $k = K$  не нужна в силу (3.11).

### 3.3.3. Моделирование Марковских процессов

При моделировании дискретных и цифровых систем возникает необходимость имитировать на ЭВМ случайные процессы, изменяющие свое значение в дискретные моменты времени  $t_i$  и при-

нимающие дискретные значения  $x_k$  ( $k = 1 \dots K$ ). Если такой случайный процесс не изменяет во времени своего пространства состояний  $\{x_k\}$ , а распределение вероятностей  $P(x_k)$  на  $i$ -том шаге зависит от исхода процесса на предыдущем ( $i - 1$ ) шаге, то такой процесс называют Марковским и описывают начальным распределением  $P(0) = P(x_k | t = 0)$  и матрицей переходов  $[P]$  размерностью  $K \times K$  [5]:

$$[P] = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1K} \\ P_{21} & P_{22} & \dots & P_{2K} \\ & & \dots & \\ P_{K1} & P_{K2} & \dots & P_{KK} \end{bmatrix}, \quad (3.12)$$

причем сумма элементов матрицы (3.12) по строке

$$\sum_{j=1}^K P_{ij} = 1,$$

поскольку строка матрицы (3.12) содержит закон распределения дискретной случайной величины  $x_k$ , а номер строки соответствует состоянию (исходу) случайной величины  $x_k$  на предыдущем шаге процесса.

Физический смысл матрицы переходов (3.12) можно понять, составив граф переходов случайного процесса из одного состояния в другое. На рис. 3.9 представлен граф переходов Марковского процесса, имеющего три состояния (1), (2) и (3), т.е.  $k = 1 \dots 3$ .

Вероятность  $P_{11}$  говорит о том, что процесс находится на 1-м уровне, т.е.  $x(t_i) = x_1$ , и на следующем шаге останется в том же состоянии с вероятностью  $P_{11}$ , перейдет в состояние (2) с вероятностью  $P_{12}$ , либо в состояние (3) с вероятностью  $P_{13}$ , т.е. вероятность перехода процесса из состояния (2) в состояние (3) –  $P_{23}$ , из состояния (3) в состояние (1) –  $P_{31}$  и т.д.

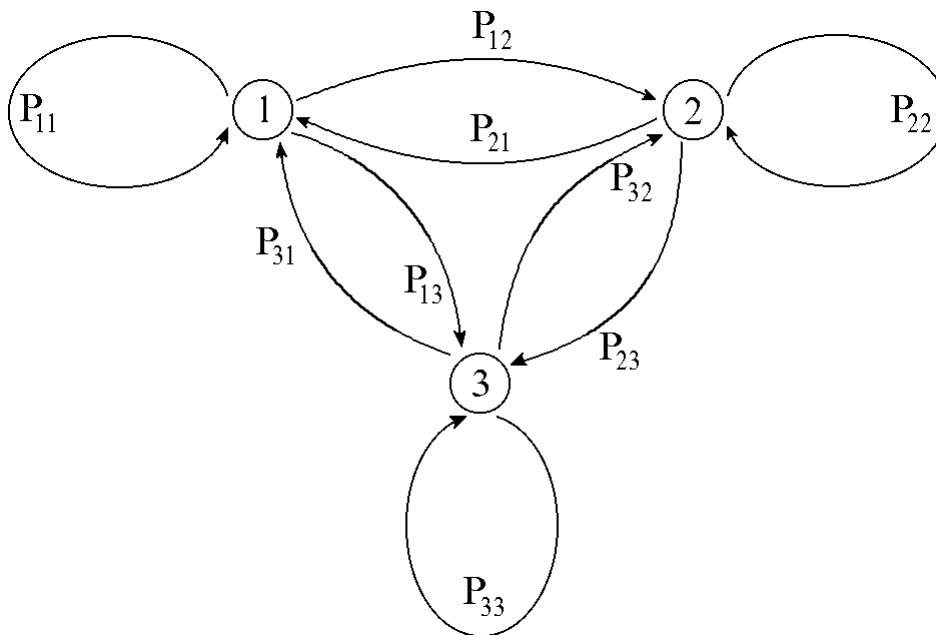


Рис. 3.9. Граф переходов Марковского процесса

Моделирование таких процессов выполняется в два этапа (см. рис. 3.10). Поскольку начальное состояние процесса (значение  $x_k = x[t = 0]$ ) априорно не определено, а задано лишь распределение  $P(0) = P(x_k | t = 0)$ , то реализацию начального состояния случайного процесса  $x^{(0)}$  получают как реализацию случайной величины с дискретным законом распределения  $p_k = P(0)$  по методике п. 3.3.2.

По полученному значению  $x^{(0)}$  получают первый отсчет Марковского процесса  $x^{(1)}$  как реализацию случайной величины с дискретным законом распределения  $p_k = P_{jk}$ ,  $k = 1 \dots K$ , причем используемый номер строки  $j$  матрицы переходов (3.12) определяется как порядковый номер значения  $x^{(0)}$ . Например, если  $x^{(0)} = x_2$ , то  $j = 2$ , и в качестве закона распределения для получения  $x^{(1)}$  используется вторая строка матрицы (3.12).

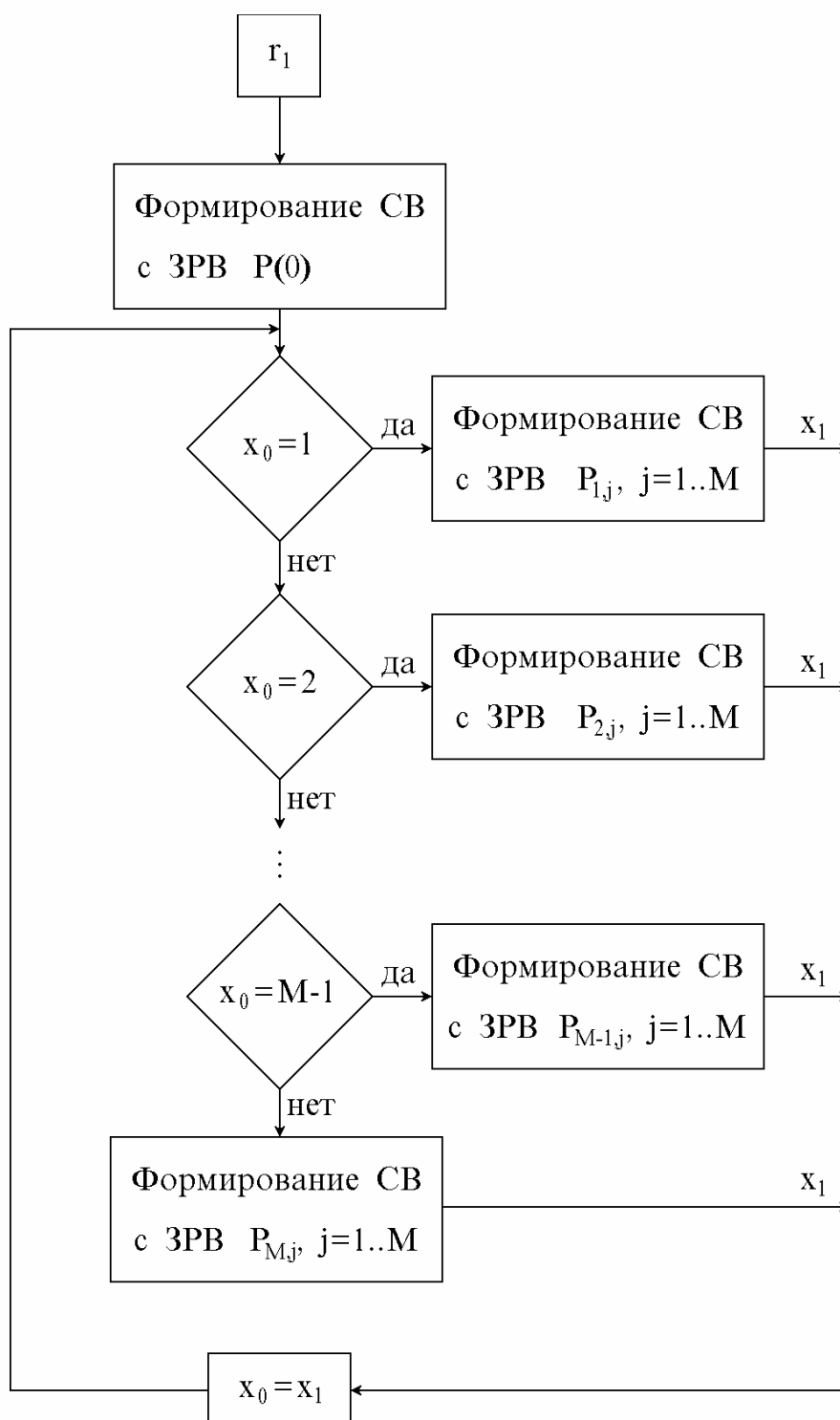


Рис. 3.10. Алгоритм моделирования Марковского процесса

Аналогично все последующие отсчеты процесса  $x^{(i)}$ ,  $i = 2 \dots N$  ( $N$  – требующийся объем выборки) получаются по предыдущим  $x^{(i-1)}$  путем определения порядкового номера  $j$  реализации  $x^{(i-1)}$ .

Таким образом, моделирование Марковского случайного процесса сводится к моделированию дискретной случайной величины с изменяемым на каждом шаге моделирования законом распределения.

### 3.3.4. Моделирование непрерывных случайных величин

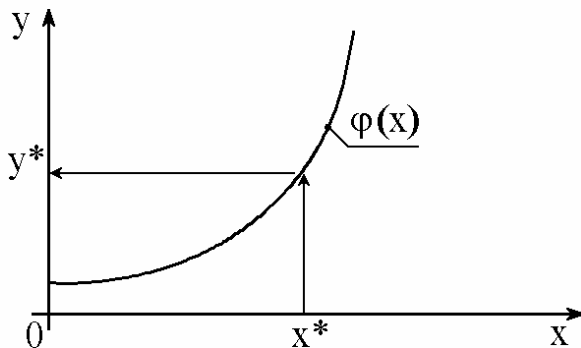


Рис. 3.11. Функциональное преобразование случайной величины

Предположим, что имеется случайная величина  $x$  с плотностью распределения  $f(x)$ . Случайная величина  $x^*$  может быть трансформирована в величину  $y^*$  посредством непрерывного преобразования  $y = \varphi(x)$ , как показано на рис. 3.11.

Случайная величина  $y$  характеризуется законом распределения  $f(y)$ . В силу непрерывности преобразования  $\varphi(x)$  соблюдается сохранение вероятности [4, 5]

$$P\{x_i < x\} = P\{y_i < y\} \text{ или } F(x) = F(y),$$

где  $x_i$ ,  $y_i$  – реализации величин  $x$  и  $y$ , при этом  $y_i = \varphi(x_i)$ , или

$$\int_{-\infty}^x f(x) dx = \int_{-\infty}^y f(y) dy. \quad (3.13)$$

Пусть  $x$  – стандартная величина  $r$ , равномерно распределенная на интервале  $[0 \dots 1]$ . Тогда из соотношений (3.13) и (3.10) получаем

$$r = \int_{-\infty}^y f(y) dy. \quad (3.14)$$



Решение уравнения (3.14) относительно  $y$  приводит к необходимой функции преобразования  $\varphi(r)$  для получения по реализации  $r_i$  стандартной случайной величины  $r$  реализации  $y_i$  случайной величины  $y$  с требуемым законом распределения  $f(y)$ . Уравнение (3.14) можно записать так же как

$$r = F(y), \quad (3.15)$$

а его решение – как

$$y = F^{-1}(r), \quad (3.16)$$

где  $F^{-1}$  – функция, обратная  $F$ .

Таким образом, алгоритм моделирования случайной величины  $y$  с заданной законом распределения (плотностью распределения  $f(y)$  или функцией распределения  $F(y)$ ) получается путем нахождения функции  $F^{-1}(\bullet)$ , обратной к функции распределения  $F(y)$ .

Если в качестве исходных данных задана плотность распределения  $f(y)$ , то необходимо аналитически получить выражения

$$F(y) = \int_{-\infty}^y f(y)dy, \text{ и } y = F^{-1}(r),$$

после чего по реализации стандартных случайных чисел  $r_i$ , равномерно распределенных на интервале  $[0..1]$ , может быть получена реализация случайной величины  $y_i = F^{-1}(r_i)$ , как показано на рис. 3.15.

Применение метода обратной функции (3.16) проиллюстрируем на примере экспоненциального распределения  $f(y) = \lambda \cdot e^{-\lambda \cdot y}$ .

Функция распределения

$$F(y) = \int_{-\infty}^y f(y)dy = \int_0^y \lambda \cdot e^{-\lambda \cdot y} dy = 1 - e^{-\lambda \cdot y} = r,$$

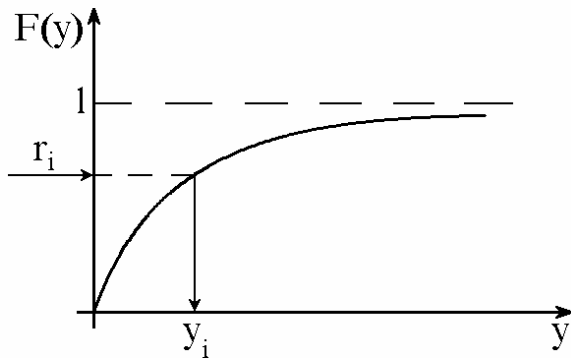


Рис. 3.12. Моделирование непрерывных случайных величин методом обратной функции

откуда обратная функция

$$y = -\frac{1}{\lambda} \cdot \ln(1 - r).$$

Тогда алгоритм получения реализации случайной величины  $y$  с экспоненциальным законом распределения с учетом симметричности равномерного закона распределения будет иметь вид

$$y_i = -\frac{1}{\lambda} \cdot \ln(r_i),$$

а его программная реализация представлена в приложении 1.

Методом обратной функции могут быть получены случайные величины с любым интегрируемым законом распределения.

К сожалению, не для всех законов распределения существует явная обратная функция  $F^{-1}(\bullet)$ . В частности, это наблюдается при широко применяемом нормальном законе распределения

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot \exp\left\{-\frac{(x-m)^2}{2 \cdot \sigma^2}\right\} = N(m, \sigma).$$

В таких случаях применяют другие методы формирования реализаций, основанные на частных свойствах распределения. Известно, что закон распределения суммы достаточно большого числа произвольно распределенных слагаемых ("центральная предельная теорема") стремится к нормальному [4]. С учетом этого свойства алгоритм формирования реализаций  $y_i$  с нормальным распределением из реализаций  $r_i$  стандартной величины строят следующим образом:

1) берут  $n$  независимых реализаций величины  $r$  равномерно распределенной в интервале  $[0...1]$ :

$$\{r_j\}, \quad j = 1 \dots n$$

2) формируют сумму

$$z = \sum_{j=1}^n r_j,$$

которая распределена нормально и имеет параметры

$$m_z = n \cdot 0,5; \quad \sigma_z = \sqrt{\frac{n}{12}}.$$

3) центрируют и нормируют величину  $z$ , сводя ее к стандартной нормальной величине  $z_0$  с распределением  $N(0,1)$ ,

$$z_0 = \frac{z - m_z}{\sigma_z}.$$

4) величину  $z_0$  преобразовывают в нормально распределенную случайную величину  $y$  с распределением  $N(m, \sigma)$

$$y = m + \sigma \cdot z_0.$$

5) окончательно записывают алгоритм формирования реализаций

$$y_i = m + \sigma \cdot \sqrt{\frac{12}{n}} \cdot \left\{ \sum_{j=1}^n r_j - \frac{n}{2} \right\}. \quad (3.17)$$

Этот алгоритм позволяет получить хорошее приближение к нормальному закону уже при  $n \geq 10$ . Наиболее удобна форма алгоритма (3.17) при  $n = 12$  и  $n = 48$ :

$$y_i = m + \sigma \cdot \left\{ \sum_{j=1}^{12} r_j - 6 \right\}, \quad y_i = m + \sigma \cdot 2 \cdot \left\{ \sum_{j=1}^{48} r_j - 24 \right\}.$$

В первом случае мы имеем быстрый алгоритм со средней точностью формирования случайных чисел с нормальным законом распределения, во втором – более медленный при более высокой точности. Следует отметить, что увеличение объема суммируемой выборки в (3.17)

свыше  $n = 50$  не приводит к заметному улучшению качества генератора случайных чисел. Пример программной реализации алгоритма (3.17) приведен в прил. 1, а пример его применения – в прил. 2.

На основе случайных величин с нормальным законом распределения могут быть получены случайные величины с другими законами распределения путем применения нелинейных преобразований. Например, случайные величины с Рэлеевским законом распределения

$$f(y) = \frac{y}{\sigma^2} \cdot e^{\frac{-y^2}{2\sigma^2}}, \quad y > 0, \quad \sigma = 1$$

могут быть получены из 2-х нормально распределенных чисел  $x_1$  и  $x_2$

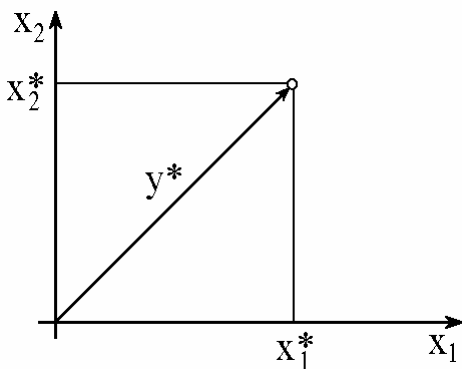


Рис. 3.13. Длина случайного вектора

путем преобразования  $y_i = x_{1i}^2 + x_{2i}^2$ , поскольку при преобразовании нормально распределенной двумерной случайной величины  $N(0,1)$  (точка на плоскости, см. рис. 3.13) из декартовой системы координат в полярную длина радиус-вектора подчиняется Рэлеевскому закону распределения [4].

### 3.3.5. Моделирование случайных векторов

Пусть необходимо получить векторную случайную величину

$$\vec{y} = \begin{bmatrix} y_1 \\ \dots \\ y_M \end{bmatrix} \text{ с заданным математическим ожиданием } \vec{M}_y = \begin{bmatrix} m_1 \\ m_2 \\ \dots \\ m_n \end{bmatrix} \text{ и задан-$$

ной корреляционной матрицей  $K_y = [K_{ij}]$ ,  $i, j = \overline{1, n}$  [4]

$K_{ij} = \sigma_i \cdot \sigma_j \cdot r_{ij}$ , где  $r_{ij}$  – коэффициент корреляции (нормированный). Известно [6, 7], что любой вектор  $\vec{y}$  с коррелированными компонентами  $y_i$  может быть декоррелирован путем линейных матричных опера-

ций с использованием собственных чисел и собственных векторов матрицы  $[K]$ . Соответственно, возможно и обратное преобразование некоррелированного вектора случайных величин  $\bar{x}$  с нулевым математическим ожиданием и единичной дисперсией, т.е. диагональной единичной корреляционной матрицей

$$K_x = I = \begin{bmatrix} 1 & \dots & 0 \\ \dots & 1 & \dots \\ 0 & \dots & 1 \end{bmatrix}.$$

Преобразование будет иметь вид

$$\bar{y} = [A] \cdot \bar{x}, \quad (3.18)$$

где матричное преобразование  $[A]$  определяется собственными числами  $[\lambda_i]$  и собственными векторами  $[\Phi]$  требуемой корреляционной матрицы  $[K]$ :

$$[A] = [\Phi] \cdot [\lambda]^{1/2}.$$

Алгоритм формирования случайного вектора с коррелированными компонентами имеет следующие этапы:

1. По заданной корреляционной матрице  $[K]$  требуемого случайного вектора  $\bar{y}$  размерностью  $n$  находят  $n$  собственных чисел и  $n$  собственных векторов матрицы  $[K]$ :

$$[\lambda] = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \quad [\Phi] = \begin{bmatrix} \Phi_{11} & \dots & \Phi_{1n} \\ \Phi_{12} & & \dots \\ \Phi_{1n} & & \Phi_{nn} \end{bmatrix}.$$

2. По методике п. 3.3.4 генерируется  $n$  реализаций  $x_i$  заданным законом распределения, из которых формируется  $n$ -мерный вектор  $\bar{x}$  некоррелированных случайных величин с нулевым математическим ожиданием и единичной дисперсией.

3. Выполняется перемножение матриц

$$\begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \dots & \Phi_{1n} \\ \dots & \dots & \dots \\ \Phi_{n1} & \dots & \Phi_{nn} \end{bmatrix} \cdot \begin{bmatrix} \sqrt{\lambda_1} & 0 & 0 \\ 0 & \sqrt{\lambda_2} & 0 \\ 0 & 0 & \sqrt{\lambda_n} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix}.$$

4. Полученный таким образом коррелированный случайный вектор  $\vec{y}$  с нулевым математическим ожиданием приводится к требуемому мат. ожиданию

$$\vec{y} = \vec{y} + \vec{M}.$$

Возможен и другой подход к моделированию коррелированных случайных векторов. Можно матрицу  $[A]$  в (3.18) выбрать треугольной, тогда

$$\begin{aligned} y_1 &= a_{11} \cdot x_1, \\ y_2 &= a_{21} \cdot x_1 + a_{22} \cdot x_2, \\ y_n &= a_{n1} \cdot x_1 + \dots + a_{nn} \cdot x_n. \end{aligned}$$

Элементы  $a_{ki}$  матрицы  $[A]$  находят из условий

$$M\{y_k y_i\} = K_{ki}, \quad M\{x_k x_l\} = \begin{cases} 1, k = l \\ 0, k \neq l \end{cases}$$

$$M\{y_1^2\} = a_{11}^2 = K_{11},$$

$$M\{y_1 y_2\} = a_{11} a_{12} = K_{12},$$

$$M\{y_2^2\} = a_{21}^2 + a_{22}^2 = K_{22} \quad \text{и т.д.};$$

откуда получают

$$a_{11} = \sqrt{k_{11}}, \quad a_{21} = \frac{k_{12}}{\sqrt{k_{11}}}, \quad a_{22} = \sqrt{k_{22} - \frac{k_{12}^2}{k_{11}}} \quad \text{и т.д.}$$

Записывают реализацию  $y_i$  вектора  $\vec{y}$  по реализации  $x_i$  вектора  $\vec{x}$  как

$$\vec{y} = \vec{m}_y + [A] \cdot \vec{x}.$$

### 3.3.6. Моделирование стационарных случайных процессов

#### Моделирование случайных процессов матричным методом

Рассмотренный выше подход к формированию случайных векторов может быть применен для формирования реализаций стационарного случайного процесса, отсчеты которого имеют статистическую взаимосвязь, т.е. являются коррелированными [5, 7].

Последовательность отсчетов  $x_i$  процесса  $x(t)$  может быть представлена в виде  $n$ -мерного вектора  $\vec{x} = [x_1; x_2; x_3; \dots; x_n]$  (см. рис. 3.14). При этом размерность этого вектора определяется радиусом корреляции процесса  $x(t)$ . Если радиус корреляции процесса равен  $T$ , а интервал дискретизации равен  $\Delta t$ , то размерность вектора  $n = T/\Delta t$ .

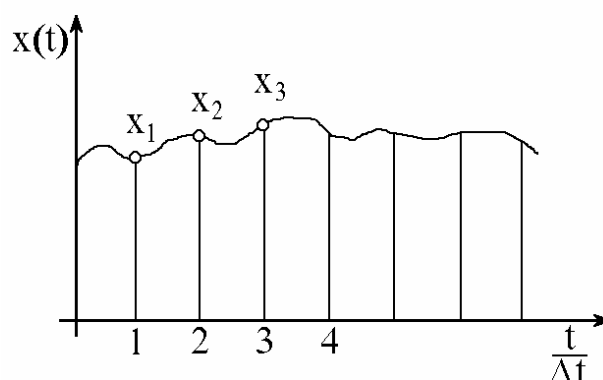


Рис. 3.14. Формирование вектора из отсчетов случайного процесса

Если радиус корреляции процесса равен  $T$ , а интервал дискретизации равен  $\Delta t$ , то размерность вектора  $n = T/\Delta t$ .

Например, задана автокорреляционная функция  $R(\tau)$ , приведенная на рис. 3.15. Радиус корреляции составляет  $3\Delta t$ ,  $\Rightarrow n = 3$ . Тогда  $R(\tau)$  может быть представлена в виде корреляционной матрицы:

$$K = \begin{bmatrix} 1 & 0.5 & 0.1 \\ 0.5 & 1 & 0.5 \\ 0.1 & 0.5 & 1 \end{bmatrix}.$$

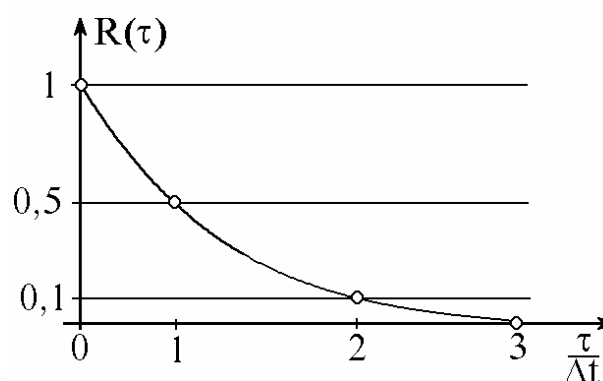


Рис. 3.15. Корреляционная функция процесса

Алгоритм моделирования случайного процесса  $x(t)$  с заданными законом распределения  $f(x)$  и корреляционной функцией  $R(\tau)$  состоит из следующих этапов:

- 1) генерация последовательности  $N + 1$  отсчетов некоррелированной случайной величины  $y_i$  с законом распределения  $f(x)$ ;
- 2) формирование корреляционной матрицы  $K_{ij} = R[(i - j)\Delta t]$  по заданной корреляционной функции  $R(\tau)$ ;
- 3) выборка первых  $n$  из  $N$  отсчетов и формирование из них вектора  $\bar{y}$ .
- 4) преобразование  $\bar{y}$  в  $\bar{x}$  согласно методике п. 3.3.5;
- 6) выборка из  $n$  компонент вектора  $\bar{x}$  первого из последовательности отсчетов процесса  $x(t_1) = x_1$ ;
- 7) сдвиг входной последовательности  $y_i$  на 1 отсчет;
- 8) повторение пп. 4-7  $N$  раз для получения  $x_i, i = 1 \dots N$ .

Преимуществом данного подхода является высокая точность формирования заданных корреляционных связей между отсчетами случайного процесса. Недостаток – значительные вычислительные затраты на матричные операции. Матричный метод применим при моделировании случайных процессов с "короткими" корреляционными функциями, для которых размерность вектора невысока.

### **Моделирование стационарных случайных процессов с заданной корреляционной функцией спектральным методом**

Более эффективный путь моделирования случайного процесса – применение преобразования Фурье. Поскольку корреляционная функция СП случайного процесса связана с энергетическим спектром сигнала преобразованием Фурье [1, 5]

$$S_x^2(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} R(\tau) \cdot e^{-j\omega\tau} d\tau,$$

то по  $R(\tau)$  может быть определен энергетический спектр. Тогда методика моделирования состоит из следующих этапов:

- 1) моделирование некоррелированной выборки  $y_i, i = 1 \dots N$  с за-



данным законом распределения;

2) нахождение по корреляционной функции формы требуемого  $S_x(\omega)$ ;

3) нахождение спектра процесса  $y(t)$  как  $Y(\omega) = F(y(t))$ ;

4) нахождение спектра процесса  $X(\omega) = S_x(\omega) \cdot Y(\omega)$ ;

5) вычисление реализации процесса  $x(t)$ :

$$x(t) = F^{-1}(X(\omega)).$$

Недостаток данного метода заключается в отсутствии информации о фазовом спектре, за счет чего имеет место неточное формирование заданной корреляционной функции. Кроме того, при моделировании происходит частичное искажение формы и параметров закона распределения.

### 3.3.7. Методы статистического анализа данных

Для статистической обработки данных имитационного эксперимента применяют типовые алгоритмы статистического анализа наблюдений, основными из которых являются следующие:

1. **Выборочное среднее значение** (оценка математического ожидания)

$$\bar{x} = \frac{1}{N} \cdot \sum_{i=1}^N x_i,$$

где  $N$  – число отсчетов, а  $x_i$  – значения отсчетов. Рассчитываемая таким образом величина является несмещенной оценкой истинного среднего значения (см. прил.1).

2. **Выборочное среднеквадратическое отклонение**

$$\hat{\sigma} = \sqrt{\sum_{i=1}^N \frac{(x_i - \bar{x})^2}{N - 1}}.$$

Величина  $\sigma^2$  является несмещенной оценкой дисперсии случайного процесса (см. прил.1).

### 3. Гистограмма плотности распределения

При объеме выборки  $N$  плотность распределения  $f(x)$  можно оценить как [6]

$$\hat{f}(x) = \frac{N_X}{\Delta x \cdot N},$$

где  $\Delta x$  – узкий интервал, симметричный относительно точки  $x$ ;  $N_X$  – количество попаданий в интервал  $\Delta x$ . При равномерном шаге  $\Delta x$  одним из наиболее быстрых является адресный алгоритм построения гистограммы, состоящий из следующих этапов:

- для заданной выборки  $x_i$ , где  $i = 1 \dots N$ , находят минимальное ( $MIN$ ) и максимальное ( $MAX$ ) значения;
- определяют шаг  $\Delta X$  для заданного количества разбиений  $M$  гистограммы  $\Delta X = \frac{(MAX - MIN)}{M}$ ;
- для каждого значения  $x_i$ , где  $i = 1 \dots N$ , определяют его положение в гистограмме  $H: K = \frac{(x_i - MIN)}{\Delta X}$  ( $K$  находится в интервале  $[0 \dots M]$ );
- выполняют накопление гистограммы  $H_K = H_K + 1$ ; массив гистограммы  $H$  должен быть предварительно обнулен;
- оценивают плотность распределения

$$\hat{f}_K = \frac{H_K}{N \cdot \Delta x}, \quad k = 0 \dots M.$$

Необходимо отметить, что в большинстве программных средств имеются встроенные процедуры построения гистограмм. Например, в программе MathCAD для этих целей служит функция  $hist(\bullet)$ .

Аналогичная подпрограмма на языке Паскаль приведена в приложении 1, а пример ее применения – в приложении 2.

#### 4. **Функция распределения**

$$F(x) = \int_{-\infty}^x f(z) dz,$$

может быть оценена по результатам оценки плотности распределения [6]:

$$\hat{F}_k = \sum_{i=0}^k \hat{f}_i, \quad k = 0..M.$$

5. **Корреляционная матрица случайного вектора**  $[K]$  и коэффициенты корреляции  $r_{kj}$  могут быть оценены непосредственно по отсчетам компонент вектора  $x_{ji}$ ,  $j = 1..n$ ,  $i = 1..N$ , как

$$\hat{K}_{kj} = \frac{1}{N} \sum_{i=1}^N (x_{ki} - \bar{x}_k) \cdot (x_{ji} - \bar{x}_j), \quad \hat{r}_{kj} = \frac{\hat{K}_{kj}}{\hat{\sigma}_k \cdot \hat{\sigma}_j}$$

где  $n$  – размерность вектора,  $N$  – объем выборки (количество отсчетов),  $k = 1..n$ .

6. **Корреляционная функция случайного процесса** может быть определена либо прямыми вычислениями по отсчетам процесса  $x_i$  (см. программу в прил.1 и пример ее применения в прил.2) как

$$\hat{R}(l) = \frac{1}{N-l} \sum_{i=1}^{N-l} (x_i - \bar{x}) \cdot (x_{i+l} - \bar{x}), \quad (3.19)$$

либо методом быстрого преобразования Фурье:

$$\tilde{R} = F^{-1} \{ X(\omega) \cdot X^*(\omega) \},$$

где  $X(\omega) = F\{x(t)\}$ ,  $*$  – знак комплексного сопряжения. Вторым способом более экономичен по вычислительным затратам, однако следует помнить, что вычисленная при помощи преобразования Фурье корреляционная функция является "циклической" [6], т.е.

$$\tilde{R}(l) = \frac{N-l}{N} \cdot \{ \hat{R}(l) + \hat{R}(N-1-l) \}, \quad (3.20)$$

где  $\hat{R}(\bullet)$  – корреляционная функция, вычисленная как (3.19). На практике для быстро затухающих корреляционных функций этот эффект несущественен, однако для повышения точности алгоритма рекомендуется дополнять выборку из  $N$  отсчетов таким же количеством нулей. При этом прямая и "зеркальная" составляющие корреляционной функции в (3.20) полностью разделяются, и их наложение друг на друга не происходит [6].

### 7. Оценка формы закона распределения.

Форму плотности вероятности характеризуют нормированные центральные моменты 3-го и 4-го порядка – коэффициенты асимметрии  $\alpha$  и эксцесса  $\varepsilon$  (см. прил. 1):

$$\hat{\alpha} = \frac{\sum_{i=1}^N (x_i - \bar{x})^3}{N \cdot \hat{\sigma}^3}, \quad \hat{\varepsilon} = \frac{\sum_{i=1}^N (x_i - \bar{x})^4}{N \cdot \hat{\sigma}^4} - 3. \quad (3.21)$$

Коэффициент асимметрии характеризует симметричность закона распределения (например, для нормального закона  $\alpha = 0$ , для экспоненциального –  $\alpha = 2$ ). Коэффициент эксцесса описывает "островершинность" закона распределения. За эталон принят нормальный закон –

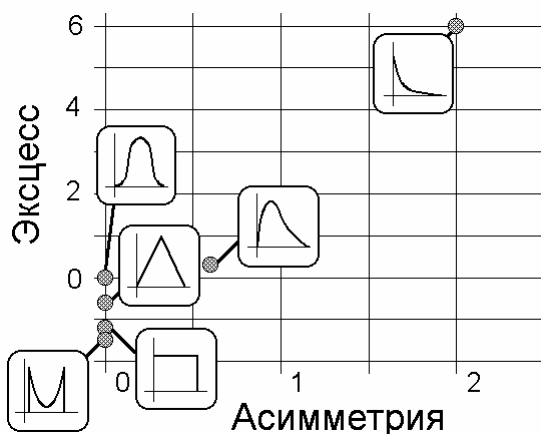


Рис. 3.16. Фрагмент плоскости моментов

$\varepsilon = 0$ . Для равновероятного закона  $\varepsilon = -1.2$ , треугольного (Симпсона)  $\varepsilon = -0.6$ , экспоненциального  $\varepsilon = 6$ .

Если теоретические значения асимметрии  $\alpha$  и эксцесса  $\varepsilon$  для различных законов распределения нанести на плоскость в системе координат  $\{\alpha, \varepsilon\}$ , как показано на рис.

3.16, то получится диаграмма, называемая плоскостью моментов.

8. **Выбор статистической модели** (подбор закона распределения по экспериментальным данным). Если значения оценок коэффициентов асимметрии  $\hat{\alpha}$  и эксцесса  $\hat{\varepsilon}$ , полученные согласно (3.21) нанести точкой на плоскость моментов (рис. 3.16), то в качестве стати-

стической модели исследуемой случайной величины или случайного процесса можно принять ближайший из известных законов распределения. Если на плоскости моментов рядом с точкой, соответствующей экспериментальным данным, оказалось несколько известных законов распределения, необходимо проверить все варианты статистической модели на адекватность.

9. **Проверка согласия теоретического и экспериментального распределений** может быть осуществлена по критерию  $\chi^2$ . Для его применения необходимо вычислить гистограмму экспериментальной выборки  $H(x)$  и теоретические значения плотности для предполагаемого закона распределения  $f(x)$ . После чего нужно найти теоретические значения вероятностей попадания в интервалы гистограммы

$$P_k = \int_{x_k}^{x_{k+1}} f(z) dz, \quad k = 1..M,$$

где  $M$  – количество интервалов в гистограмме. Отклонение  $\chi^2$  может быть определено по формуле

$$\chi^2 = \sum_{k=1}^M \frac{(N \cdot P_k - H_k)^2}{N \cdot P_k}, \quad (3.22)$$

где  $N$  – количество отсчетов в выборке  $\{x_i\}$ .

Вывод о непротиворечии эмпирического распределения (гистограммы) выбранному теоретическому распределению делают на основе сравнения рассчитанного значения  $\chi^2$  с квантилем этого распределения  $\chi_{\text{табл}}^2(\tau, P)$ , приведенным в таблицах [2]. Квантиль распределения определяется заданной вероятностью  $P$  принятия правильного решения (о непротиворечии распределений) и числом степеней свободы  $\tau$ , которое равно количеству разрядов гистограммы за вычетом количества независимых условий, наложенных на экспериментальные вероятности  $f_k(x)$ :

- 1)  $\sum f_k = 1$  (это условие налагают всегда);
- 2)  $\sum x_k \cdot f_k = m_x$  (если предъявляют требование соответствия среднего значения);
- 3)  $\sum x_k^2 \cdot f_k = \sigma_x^2$  (если предъявляют требование соответствия дисперсии);

Если  $\chi^2 < \chi_{\text{табл}}^2$ , то теоретическое и эмпирическое распределения с вероятностью  $P$  не противоречат друг другу.

### 3.4. Методы цифрового моделирования динамических систем

Электронная аппаратура, ее конструкция и элементы, технологические процессы изготовления в подавляющем большинстве случаев являются динамическими системами, которые могут быть описаны в общем случае нелинейными нестационарными дифференциальными уравнениями вида:

$$\Phi \{y^{(n)}, y^{(n-1)}, \dots, y^{(1)}, y, x^{(m)}, \dots, x^{(1)}, x, t\} = 0,$$

где  $y(t)$ ,  $x(t)$  – выходная реакция и входное воздействие на объект соответственно.

Согласно теореме Котельникова временные функции  $y(t)$ ,  $x(t)$  можно заменить их дискретными отсчетами  $y(t_i)$ ,  $x(t_i)$ , которые в памяти ЭВМ представляются массивами чисел  $y[i]$ ,  $x[i]$ . Сущностью цифрового моделирования объектов является установление взаимосвязи между  $y(t_i)$  и  $x(t_i)$ , т.е.  $y(t_i) = A\{x(t_i)\}$  и воспроизведение этой связи на ЭВМ.

В большинстве случаев удается осуществить декомпозицию объекта и представить его в виде структурной схемы. Тогда цифровое моделирование сводится к последовательному цифровому моделированию преобразования сигнала по структурной схеме, что не вызывает принципиальных трудностей, если известны алгоритмы моделирования отдельных блоков [8].

Наиболее сложным при этом является цифровое моделирование линейных динамических объектов [1, 3, 8], описываемых либо дифференциальным уравнением

$$\sum_{i=0}^n a_i \cdot y^{(i)}(t) = \sum_{j=0}^m b_j \cdot x^{(j)}(t), \quad (3.23)$$

либо эквивалентными ему передаточной функцией

$$W(p) = \frac{\sum_{j=0}^m b_j \cdot p^j}{\sum_{i=0}^n a_i \cdot p^i} = \frac{Y(p)}{X(p)}, \quad (3.24)$$

или импульсной переходной характеристикой (реакцией объекта на воздействие  $\delta$  - функции), определяемой как обратное преобразование Лапласа передаточной функции

$$\omega(t) = L^{-1}\{W(p)\}. \quad (3.25)$$

С использованием этих характеристик могут быть получены два основных алгоритма цифрового моделирования: прямой и рекуррентный.

### 3.4.1. Прямой алгоритм цифрового моделирования

Этот алгоритм основан на уравнении свертки, связывающем входной и выходной сигналы с импульсной переходной характеристикой

$$y(t) = \int_{-\infty}^{\infty} \omega(\tau) \cdot x(t - \tau) d\tau. \quad (3.26)$$

Для реальных физически осуществимых линейных систем [1,3] функция  $\omega(t)$  имеет двустороннее ограничение:

$\omega(t) \equiv 0$  при  $t < 0$  (это условие всегда справедливо для физически осуществимых систем);

$\omega(t) \equiv 0$  при  $t > T$  (функцию  $\omega(t)$ , неограниченную вправо, мож-

но приближенно заменить ограниченной, если  $\omega(t) \rightarrow 0$  при  $t \rightarrow \infty$ ).

С учетом этих ограничений уравнение свертки приобретает вид

$$y(t) = \int_0^T \omega(\tau) \cdot x(t - \tau) d\tau. \quad (3.27)$$

Пусть отсчеты входного и выходного сигналов взяты с шагом  $\Delta t$ , и пусть  $\Delta t$  кратен величине  $T$ , т.е.  $T/\Delta t = N$ . Тогда дискретные значения выходного сигнала в моменты  $t_i = i \cdot \Delta t$

$$y(i \cdot \Delta t) = \int_0^T \omega(\tau) \cdot x(i \cdot \Delta t - \tau) d\tau.$$

При достаточно малом  $\Delta t$  интеграл можно заменить суммой по способу прямоугольников (замена подынтегральной функции ступенчатой кривой):

$$y_i = \Delta t \cdot \sum_{j=0}^{N-1} \omega[j] \cdot x[i - j], \quad (3.28)$$

где  $\omega[j] = \omega(j \cdot \Delta t)$  – дискретные значения импульсной переходной характеристики.

Выражение (3.28) представляет собой алгоритм скользящего весового суммирования входного процесса  $x[i]$  с весовой функцией

$$a[i] = \Delta t \cdot \omega[i].$$

Если применить другие методы численного интегрирования в выражении (3.27) (например, трапеций или Симпсона), то весовой коэффициент изменится.

Алгоритм (3.28) соответствует вычислению т.н. круговой свертки

$$y[i] = \sum_{k=0}^{N-1} x[k] \cdot a[i - k] \quad \text{или} \quad y[i] = \sum_{k=0}^{N-1} a[k] \cdot x[i - k],$$

где  $i = 0, 1, 2, \dots, N - 1$ . Данный метод неявно предполагает, что рассматривается периодическая последовательность  $x[i]$  с периодом  $N$  отсчетов.



Последовательность  $y[i]$  также будет периодической с периодом  $N$  отсчетов, поэтому достаточно ее вычислить на одном периоде.

Реализация данного алгоритма требует  $N \cdot (N + 1)$  операций умножения, что ограничивает применяемость данного метода.

### 3.4.2. Спектральный метод цифрового моделирования

Этот метод является разновидностью прямого алгоритма цифрового моделирования и состоит в применении к выражению (3.26) преобразования Фурье:

$$F\{y(t)\} = F\{\omega(t)\} \cdot F\{x(t)\}. \quad (3.30)$$

Для существования преобразования Фурье функция  $\omega(t)$  должна удовлетворять условию

$$\lim_{t \rightarrow \infty} \omega(t) = 0.$$

Если это условие выполняется, то

$$F\{\omega(t)\} = W(p) |_{p=j\omega}.$$

Таким образом, если вместо интегрального преобразования Фурье использовать дискретное преобразование Фурье или его разновидность – быстрое преобразование Фурье (БПФ – прямое, ОБПФ – обратное), то это дает возможность уменьшить количество операций и, следовательно, сократить время моделирования.

Если известна передаточная функция системы  $W(p)$ , то она может быть представлена в дискретном виде  $W[k]$ . Если в качестве исходных данных используется импульсная характеристика, то дискретная передаточная функция может быть определена как дискретное преобразование Фурье от передаточной функции  $W[k] = F[\omega[i]]$ .

Алгоритм моделирования может быть представлен в следующем виде [3].

1. Исходный сигнал  $x(t)$  подвергается дискретизации с интерва-

лом времени  $\Delta t$  (см. п. 3.2 и рис. 3.17-а).

2. Для дискретной последовательности входного сигнала  $x[i]$  за счет применения БПФ определяется ее дискретный спектр  $X[k] = F\{x[i]\}$ , представляющий собой дискретизацию непрерывного спектра  $X(f) = F\{x(t)\}$  с шагом по частоте  $\Delta f$  (см. рис. 3.17-б).

3. Передаточная функция системы  $W(f) = W(p)|_{p=j \cdot 2\pi \cdot f}$  представляется в дискретном виде как массив комплексных коэффициентов передачи  $W[k] = W(k \cdot \Delta f)$  (рис. 3.17-в).

4. Тогда спектр выходного сигнала определяется как произведение элементов массивов  $Y[k] = X[k] \cdot W[k]$  (рис. 3.17-г).

5. Дискретная последовательность  $y[i]$  (сигнал на выходе системы) вычисляется как ОБПФ от спектра  $Y[k]$  (рис. 3.17-д):

$$y[i] = F^{-1}[Y[k]].$$

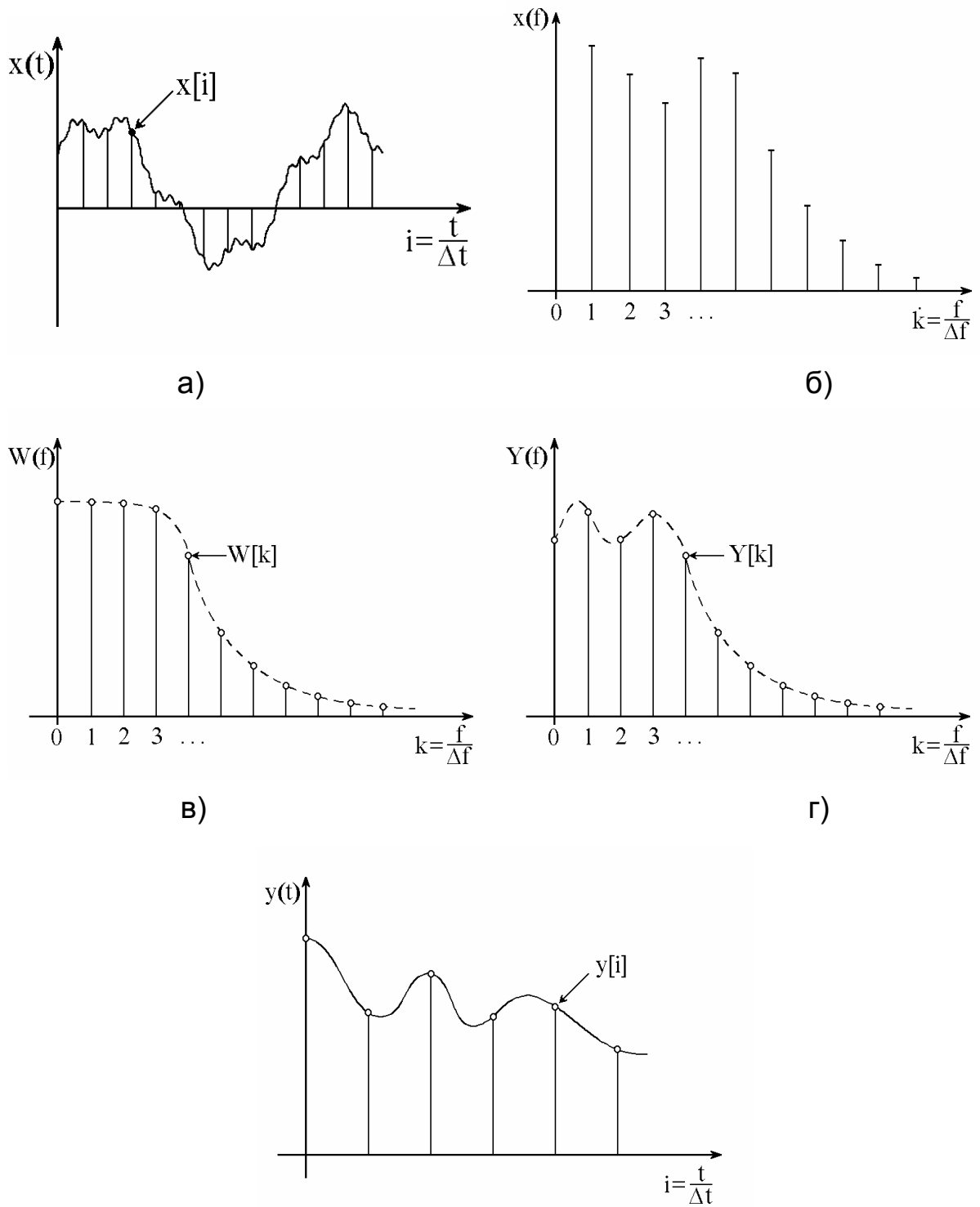
Однако за счет ограниченности длины последовательности  $y[i]$  реально восстановление происходит с погрешностью, величина которой обратно пропорциональна объему выборки как  $\sqrt{N}$ .

Очевидно, что способ вычисления выходного сигнала с помощью БПФ

$$y[i] = F^{-1}\{F\{x[i]\} \cdot F\{\omega[i]\}\} \quad (3.31)$$

предполагает, что функции  $x[i]$  и  $\omega[i]$  являются периодическими и (3.31) соответствует вычислению круговой свертки сигнала  $x[i]$  с импульсной характеристикой  $\omega[i]$ . Т.е. выходной сигнал  $y[i]$  также предполагается периодическим с длиной периода  $N \cdot \Delta t$ , где  $\Delta t$  - интервал дискретизации (см. рис. 3.3). Применение такого алгоритма к непериодическим сигналам дает значительную погрешность, если импульсная характеристика системы не является быстро спадающей функцией на интервале времени  $T = N \cdot \Delta t$ .

Линейная свертка не имеет предполагаемого периодического продолжения импульсной характеристики и также может быть вычислена



д)  
Рис. 3.17. Моделирование прохождения сигнала через систему  
спектральным методом

с помощью БПФ. Для этого импульсную характеристику длиной  $N_1$  и входной сигнал длиной  $N_2$  дополняют нулями до  $N_1 + N_2 - 1$  отсчетов.

Тогда круговая свертка, полученная с помощью БПФ, будет соответствовать линейной свертке непериодических последовательностей.

### 3.4.3. Рекуррентный алгоритм цифрового моделирования

Главный недостаток методов дискретной свертки и БПФ – необходимость иметь полную выборку входного сигнала для получения хотя бы одного отсчета выходного сигнала. В системах реального времени это неприемлемо, так как время обработки сигнала, как правило, ограничено. Применение метода Z-преобразования позволяет строить алгоритмы моделирования с минимальной задержкой на обработку входной информации. Недостатком таких алгоритмов является жесткая связь со структурой моделируемой системы, в отличие от метода БПФ, в котором структура алгоритма постоянна.

Метод Z-преобразований позволяет:

- разработать специализированный алгоритм для универсальной ЭВМ;
- разработать структуру специального вычислителя для обработки сигнала по заданной передаточной функции (осуществить так называемый синтез цифровых фильтров).

Основной характеристикой линейных динамических систем является передаточная функция системы

$$W(p) = \frac{Y(p)}{X(p)},$$

на основе которой строятся все алгоритмы моделирования.

При переходе к дискретным системам либо дискретным методам моделирования аналоговых систем, возникает задача построения дискретного аналога передаточной функции, в частности, при применении метода Z-преобразования – системной функции [3]

$$W(Z) = \frac{Y(Z)}{X(Z)},$$

которая будет служить основанием для построения дискретного алгоритма.

Этот алгоритм предполагает вычисление текущего отсчета выходной переменной  $y[i]$  по предыдущим отсчетам этой и входной переменной  $x[i]$

$$y[i] = A\{y[i-k], x[i-k]\}.$$

Временную функцию  $x(t)$ , представленную своими дискретными отсчетами  $x[i] = x(t_i)$ , взятыми через постоянные интервалы времени  $t_i = i \cdot \Delta t$ , записывают в виде гребенчатой функции

$$x^*(t) = \sum_{k=0}^{\infty} x(k \cdot \Delta t) \cdot \delta(t - k \cdot \Delta t), \quad (3.32)$$

где  $\delta(t - k \cdot \Delta t)$  –  $\delta$ -функция, смещенная в точку  $k \cdot \Delta t$ .

Преобразование Лапласа для функции (3.32) имеет вид

$$X(p) = \sum_{k=0}^{\infty} x(k \cdot \Delta t) \cdot e^{-p \cdot k \cdot \Delta t}. \quad (3.33)$$

После введения переменной  $Z = e^{-k \cdot \Delta t}$  формулу (3.33) представляют в виде

$$X(z) = \sum_{k=0}^{\infty} x(k \cdot \Delta t) \cdot Z^{-k}. \quad (3.34)$$

Выражение (3.34) является Z-преобразованием временной функции  $x(t)$ , которое обладает следующими важными свойствами [1, 3]:

– линейности

$$Z\{\sum x_i\} = \sum Z\{x_i\};$$

– сдвига:

если  $X(z) = Z\{x(t)\}$ , то для целых  $k$

$$Z\{x(t - k \cdot \Delta t)\} = z^{-k} \cdot X(z).$$

Таким образом, сдвиг влево на  $k$  тактов дискретизации эквивалентен умножению Z-преобразования на  $z^{-k}$ .

Методика применения Z-преобразования содержит следующие этапы.

1) Пусть задана передаточная функция системы  $W(p)$  в виде отношения полиномов по степеням  $p$ .

$$W(p) = \frac{\sum_{j=0}^m b_j \cdot p^j}{\sum_{i=0}^n a_i \cdot p^i}, \quad n \geq m \quad (3.35).$$

К такому виду  $W(p)$  может быть приведена либо путем преобразования Лапласа от дифференциального уравнения системы, либо путем аппроксимации экспериментально полученной частотной характеристики.

2) Разделим числитель и знаменатель (3.35) на старшую степень  $p^n$ .

$$W(p) = \frac{\sum_{j=0}^m \frac{b_j}{p^{n-j}}}{\sum_{i=0}^n \frac{a_i}{p^{n-i}}} \quad (3.36).$$

3) Представим Z-преобразование данного выражения путем замены в

(3.36)  $\frac{1}{p^k}$  на соответствующее выражение для Z, выбранное из

таблицы 3.1.

Таблица 3.1

Z-преобразования оператора  $\frac{1}{p^k}$

$k$	$\frac{1}{p^k}$	$Z\left\{\frac{1}{p^k}\right\}$
0	1	1
1	$\frac{1}{p}$	$\frac{z \cdot \Delta t}{z - 1}$
2	$\frac{1}{p^2}$	$\frac{z \cdot \Delta t^2}{(z - 1)^2}$
3	$\frac{1}{p^3}$	$\frac{1}{2} \cdot \frac{z \cdot (1 + z) \cdot \Delta t^3}{(z - 1)^3}$
4	$\frac{1}{p^4}$	$\frac{1}{6} \cdot \frac{z \cdot (z + 4z + z^2) \cdot \Delta t^4}{(z - 1)^4}$
5	$\frac{1}{p^5}$	$\frac{1}{24} \cdot \frac{z \cdot (1 + 11z + 11z^2 + z^3) \cdot \Delta t^5}{(z - 1)^5}$

4) Приведем полученное выражение к виду

$$W(Z) = \frac{\sum_{j=0}^m b_j^* \cdot Z^j}{\sum_{i=0}^n a_i^* \cdot Z^i} \quad (3.37).$$

5) Разделим числитель и знаменатель (3.37) на старшую степень  $Z^n$

$$W(Z) = \frac{\sum_{j=0}^m b_j^* \cdot Z^{-j}}{\sum_{i=0}^n a_i^* \cdot Z^{-i}} \quad (3.38).$$

6) Поскольку  $W(Z) = Y(Z)/X(Z)$ , представим (3.38) в виде

$$Y(Z) \cdot \sum_{i=1}^n a_i^* \cdot Z^{-i} = X(Z) \cdot \sum_{j=1}^n b_j^* \cdot Z^{-j} \quad (5)$$

7) Выделим из (5)  $Y(Z)$  при степени  $Z^0$

$$Y(Z) = \frac{1}{a_0^*} \left[ \sum_{j=0}^m b_j^* \cdot X(Z) \cdot Z^{-j} - \sum_{i=1}^n a_i^* \cdot Y(Z) \cdot Z^{-i} \right]. \quad (3.39)$$

8) Выполним обратное Z-преобразование, т.е. перейдем к дискретным оригиналам. Поскольку умножение на  $Z^{-k}$  означает сдвиг во времени назад на  $k$  тактов дискретизации, то

$$y(Z) \cdot Z^{-j} \div y[n-j], \quad X(Z) \cdot Z^{-i} \div x[n-i].$$

Тогда из (3.39) получим

$$y[n] = \frac{1}{a_0^*} \left[ \sum_{j=0}^m b_j^* \cdot x[n-j] - \sum_{i=1}^n a_i^* \cdot y[n-i] \right]. \quad (3.40)$$

Таким образом, получен рекуррентный алгоритм в виде линейного разностного уравнения для вычисления отклика системы  $y[n]$  на входное воздействие  $x[n]$  как функция отсчетов  $x[n-j]$ ,  $j = 0 \dots m$  и предыдущих значений  $y[n-i]$ ,  $i = 1 \dots n$ .

Необходимо отметить, что для перехода от (3.36) к (3.37) могут использоваться различные формы Z-преобразования в зависимости от способа численного интегрирования (см. табл. 3.2).



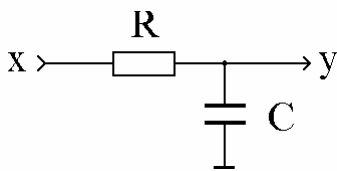
Таблица 3.2.

Зависимость вида Z – преобразования  
от метода интегрирования

Метод численного интегрирования	Z – преобразование
<p>Интегрирование "вперед":</p>	$\left[ \frac{1}{p} \right] \div \left[ \frac{z \cdot \Delta t}{z-1} \right]$
<p>Интегрирование "назад"</p>	$\left[ \frac{1}{p} \right] \div \left[ \frac{z \cdot \Delta t}{z+1} \right]$
<p>Интегрирование "методом трапеций"</p>	$\left[ \frac{1}{p} \right] \div \left[ \frac{z+1}{z-1} \cdot \frac{\Delta t}{2} \right]$

### Пример применения метода Z-преобразований

Частотная характеристика системы, представленной на рисунке 3.24, может быть представлена в виде



$$K(j\omega) = \frac{Z_2}{Z_1 + Z_2},$$

где  $Z_1 = R$ ,  $Z_2 = \frac{1}{j \cdot \omega \cdot C}$ ,  $j = \sqrt{-1}$ . Тогда

Рис. 3.18. Модель системы

$$K(j\omega) = \frac{\frac{1}{j\omega C}}{R + \frac{1}{j\omega C}} = \frac{1}{1 + j\omega RC} \quad (3.41).$$

Перейдем от (3.41) к преобразованию Лапласа при нулевых начальных условиях

$$\begin{cases} y(t) = 0, t \leq 0 \\ x(t) = 0, t < 0 \end{cases},$$

$$\frac{1}{1 + j\omega RC} = \frac{1}{1 + pRC} = W(p) \Big|_{p=j\omega}.$$

1) Разделим числитель и знаменатель на старшую степень  $p$ :

$$W(p) = \frac{\frac{1}{p}}{\frac{1}{p} + RC}.$$

2) Выполним Z-преобразование. Для интегрирования «вперед»

$$\frac{1}{p} \div \frac{Z \cdot \Delta t}{Z - 1} \Rightarrow W(Z) = \frac{\frac{Z \cdot \Delta t}{Z - 1}}{\frac{Z \cdot \Delta t}{Z - 1} + R \cdot C}. \quad (3.42)$$

3) Приведем (3.42) к виду дробно-рациональной функции:

$$W(Z) = \frac{Z \cdot \Delta t}{Z \cdot \Delta t + R \cdot C \cdot (Z - 1)} = \frac{Z \cdot \Delta t}{Z \cdot \Delta t + Z \cdot R \cdot C - R \cdot C}.$$

4) Разделим числитель и знаменатель на старшую степень  $Z$ :

$$W(Z) = \frac{\Delta t}{\Delta t + R \cdot C - R \cdot C \cdot Z^{-1}}.$$

5) Поскольку  $W(Z) = \frac{Y(Z)}{X(Z)}$ , то

$$\frac{\Delta t}{\Delta t + RC - RCZ^{-1}} = \frac{Y(Z)}{X(Z)},$$

$$Y(Z) \cdot (\Delta t + R \cdot C) - Y(Z) \cdot R \cdot C \cdot Z^{-1} = X(Z) \cdot \Delta t.$$

6) Выделим  $Y(Z)$  при нулевой степени  $Z$ :

$$Y(Z) = \frac{X(Z) \cdot \Delta t + Y(Z) \cdot Z^{-1} \cdot R \cdot C}{\Delta t + R \cdot C}.$$

7) Выполним обратное  $Z$ -преобразование:

$$y_i = \frac{X_i \cdot \Delta t + y_{i-1} \cdot R \cdot C}{\Delta t + R \cdot C}, \quad i = 1 \dots N.$$

Таким образом, получен дискретный рекуррентный алгоритм моделирования системы (рис. 3.24) с начальными условиями  $y_0 = 0$ .

Пример программной реализации полученного алгоритма и результаты его применения приведены в прил. 2.

## Библиотека процедур моделирования и статистической обработки данных

Исходные тексты процедур на языке Паскаль тестировались на компиляторах Borland Pascal 7.0 и TMT Pascal 3.0 и объединены в модуль **TpStat**. Все процедуры по назначению можно разделить на три основных группы.

1. Процедуры комплексных операций: сложения *Cadd*, вычитания *Csub*, умножения *Cmul*, деления *Cdiv*, комплексного сопряжения *Conjg*, вычисления квадратного корня *CSqrt*, модуля *Cabs*, аргумента *Carg*, реальной *RealC* и мнимой *ImagC* части комплексного числа, формируемого процедурой *Cmplx*, а также вычисления быстрого преобразования Фурье *FFT* от массива комплексных чисел. Процедуры *Init\_FFT\_table* и *Free\_FFT\_table* позволяют ускорить процесс вычислений БПФ при многократном вызове процедуры *FFT*.

2. Процедуры статистической обработки данных позволяют вычислить среднее значение и дисперсию (*mean\_disp*) для массива реальных чисел, построить гистограмму (*hist*) и определить третий и четвертый центральные моменты выборки (*statistics*). Процедура *abcf* позволяет определить как автокорреляционную (при значении параметра *bcf* = 0), так и взаимную (при *bcf* = 1) корреляционную функции для 2-х массивов реальных чисел.

3. Процедуры моделирования случайных чисел с экспоненциальным *ExpGen*, логарифмически нормальным *LnGen*, Рэлеевским *RelGen*, нормальным *NormGen* и Вейбулловским *WabGen* законами распределения.

Примеры применения некоторых процедур приведены в приложении 2.

```

UNIT TPSTAT;
INTERFACE
CONST
  Num_Max = 8192;           {Ограничение для Borland Pascal}
  fft_max  = Num_Max div 2;
  StError  : integer      = 0;           { Флаг ошибки }
  nFFT_old : integer      = 0;
  x0: single = 0.87623;           { Стартовое случайное число }
TYPE
  fft_logic = (forwd,inver);  cmplx_type = (creal,cimag);
  complex   = array [cmplx_type] of single;
  SingleArr = array [1..Num_Max] of single;
  SingleArrPtr = ^ SingleArr;
  ComplexArr = array [1..fft_max] of complex;
  ComplexArrPtr = ^ ComplexArr;
  LongintArr = array [1..Num_Max] of longint;
  LongintArrPtr = ^ LongintArr;
{ ----- Procedures ----- }
function      expr (a, y: single) : single;
procedure     Cadd (x,y:complex; var c:complex);
procedure     Csub (x,y:complex; var c:complex);
function      Cabs (x:complex):single;
procedure     Cmul (x,y:complex; var c:complex);
procedure     Cdiv (x,y:complex; var c:complex);
procedure     Cmplx(x,y : single; var c: complex);
function      RealC(c: complex): single;
function      ImagC(c: complex): single;
function      Carg (c:complex) : single;
Procedure     Conjg(ci: complex; var co: complex);
Procedure     CSqrt(X : complex; var X1 : complex);
Procedure     Csqr (cinp: complex; var cout : complex );
Function      Init_FFT_table (Ncomplex : integer) : boolean;
Procedure     Free_FFT_table;
procedure     FFT(p: pointer; nComplex:integer; inv:fft_logic);
procedure     HIST (h: longintArrPtr; Nhist : integer;
                  x: singleArrPtr; Npoint : integer);
procedure     mean_disp(x: singleArrPtr; Npoint : integer;
                      var mean, disp : single);
procedure     statistics(x: singleArrPtr; Npoint : integer;
                      var xmin, xmax, mean, disp, sko, Asymmetry, excess : single);
procedure     ExpGen (x: singleArrPtr; Npoint : integer; l:single);

```

```

procedure LnGen (x: singleArrPtr; Npoint : integer; m,s : single);
procedure WabGen (x: singleArrPtr; Npoint : integer; a,b : single);
procedure RelGen (x: singleArrPtr; Npoint : integer; l: single);
procedure NormGen(x: singleArrPtr; Npoint : integer; m,s : single);
procedure abcf(a,b,c : singleArrPtr; Npoint, bcf: integer);
IMPLEMENTATION
TYPE
  rec_cmplx = record
    case integer of
      0: (ARS : singleArr);
      1: (AC : complexArr);
    end;
  pnt_rec_cx = ^rec_cmplx;
VAR
  aw : ComplexArrPtr;
{ ----- Procedures ----- }
function expr (a, y: single) : single; {  $\text{expr} = a^y$  }
begin if a>0.0 then expr:=exp(y*ln(a)) else expr:=0; end;
procedure cadd (x,y:complex; var c:complex); begin
  c[creal]:=x[creal]+y[creal]; c[cimag]:=x[cimag]+y[cimag];
end;
procedure csub (x,y:complex; var c:complex); begin
  c[creal]:=x[creal]-y[creal]; c[cimag]:=x[cimag]-y[cimag];
end;
function cabs (x:complex):single;
begin cabs:=sqrt(sqrt(x[creal])2+sqrt(x[cimag])2); end;
procedure cmul (x,y:complex; var c:complex); begin
  c[creal]:=x[creal]*y[creal]-x[cimag]*y[cimag];
  c[cimag]:=x[creal]*y[cimag]+x[cimag]*y[creal];
end;
procedure cdiv (x,y:complex; var c:complex);
  var r : single; begin
  r:=sqrt(y[creal]2+sqrt(y[cimag])2); if r <> 0.0 then begin
    c[creal]:=(x[creal]*y[creal]+x[cimag]*y[cimag])/r;
    c[cimag]:=(-x[creal]*y[cimag]+x[cimag]*y[creal])/r;
  end else begin STERROR:=-1; c[creal]:=0.0; c[cimag]:=0.0; end;
end;
procedure Cmplx (x,y : single; var c: complex);
  begin c[creal]:=x; c[cimag]:=y; end;
function RealC (c: complex): single; begin RealC:=c[creal]; end;

```

```

function ImagC (c: complex): single; begin ImagC:=c[cimag]; end;
function Carg (c:complex) : single;
var r: single; begin
  if c[creal] <> 0 then r:= arctan(c[cimag]/c[creal]) else r:=Pi;
  if c[creal] < 0 then if c[cimag] > 0 then r:=pi+r;
  if c[cimag] < 0 then if c[creal] < 0 then r:=pi+r else r:=2*pi+r;
  Carg:=r;
end;
Procedure Conj (ci: complex; var co: complex);
begin co[creal]:=ci[creal]; co[cimag]:=-ci[cimag]; end;
Procedure cSqr ( cinp: complex; var cout : complex );
begin
  cout[creal]:=sqr(cinp[creal])-sqr(cinp[cimag]);
  cout[cimag]:=2*cinp[creal]*cinp[cimag];
end;
Procedure CSqrt (X : Complex ; Var X1 : Complex);
  Var Y,Z : Single;
Begin Y:=Cabs(X); Y:=Sqrt(Y); Z:=Carg(X);
  If (X[creal] < 0) and (X[cimag] <= 0) then Z:=Z+PI; Z:=0.5*Z;
  X1[creal]:=Y*Cos(Z); X1[cimag]:=Y*Sin(Z);
End;
function Init_FFT_table (Ncomplex : integer) : boolean;
var nxp2, m : integer; wpwr, arg : single; p : pointer;
begin
  Init_FFT_table:=false; nxp2:=Ncomplex div 2;
  if (nFFT_old <> Ncomplex) then begin
    if (nFFT_old > 0) then begin
      p:=AW; FreeMem(p,(nFFT_old div 2)*sizeof(complex));
nFFT_old:=0;
    end;
    if (maxavail <= Ncomplex*sizeof(complex)) then exit;
    if (fft_max < Ncomplex) then exit;
    GetMem(p,nxp2*sizeof(complex)); AW:=p; wpwr:=pi/nxp2;
    for m:=1 to nxp2 do begin
      arg:=wpwr*(m-1); AW^[m][creal]:=cos(arg);
AW^[m][cimag]:=sin(arg);
    end;
    nFFT_old:=ncomplex;
  end; Init_FFT_table:=true;
end;

```

**Procedure Free\_FFT\_table;**

```

var p: pointer; begin
  if (nFFT_old > 0) then begin p:=AW;
    FreeMem(p,(nFFT_old div 2)*sizeof(complex));
    nFFT_old:=0; end;
end;
procedure FFT (p:pointer ; Ncomplex:integer; inv:fft_logic);
var
  iter,irem,mxp,j1,j2,it,m,n1,n2,i,j,k,nxp,nxp2,ns,ans,n : integer;
  sign, wpwr,arg : single;      w,t : complex;
  AMA : pnt_rec_cx;      autoinit : boolean;
begin
  n:=nComplex; ama:=p; nxp2:=n div 2; autoinit:=false;
  if (nFFT_old <> n) then begin autoinit:=true;
    If not Init_FFT_table( Ncomplex) then begin
      StError:=-203; exit;
    end;
  end;
  end;
with AMA^ do begin
  iter:=0; irem:=n; irem:=irem div 2;
  while irem <> 0 do begin  iter:=iter+1; irem:=irem div 2  end;
  if inv = forwd then  sign:=-1 else  sign:=1; nxp2:=n;
  for it:=1 to iter do begin
    nxp:=nxp2; nxp2:=nxp div 2; ns:=(n div 2) div nxp2;
    for m:=1 to nxp2 do begin ans:=ns*(m-1)+1; mxp:=nxp;
      w[creal]:=AW^[ans][creal]; w[cimag]:=AW^[ans][cimag]*sign;
      while mxp <= n do begin
        j1:=mxp-nxp+m; j2:=j1+nxp2; csub(AC[j1],AC[j2],t);
        cadd(AC[j1],AC[j2],AC[j1]); cmul(t,w,AC[j2]); mxp:=mxp+nxp
      end
    end
  end;
  n2:=n div 2; n1:=n-1; j:=1;
  for i:=1 to n1 do begin
    if i < j then begin  t:=AC[j]; AC[j]:=AC[i]; AC[i]:=t  end;
    k:=n2; while k < j do begin  j:=j-k; k:=k div 2  end; j:=j+k
  end;
  if inv = forwd then  for i:=1 to n do begin
    AC[i][creal]:=AC[i][creal]/n; AC[i][cimag]:=AC[i][cimag]/n
  end; end; if autoinit then Free_FFT_table;
end;

```



```

procedure HIST (h: longintArrPtr; Nhist : integer;
                x: singleArrPtr; Npoint : integer);
var i,k : integer; dx,xn,XMIN,XMAX : single;
begin
  XMIN:=X^[1]; XMAX:=X^[1];
  for i:=1 to Npoint do begin
    IF (X^[i] > XMAX) then XMAX:=X^[i];
    IF (X^[i] < XMIN) then XMIN:=X^[i];
  end;
  xn:=Nhist;
  for i:=1 to Nhist do H^[i]:=0; DX:=(XMAX-XMIN)/xn;
  for i:=1 to Npoint do begin
    K:=Trunc((X^[i]-XMIN)/DX)+1;
    IF (K > Nhist) then K:=Nhist; H^[K]:=H^[K]+1;
  end;
end;

procedure mean_disp(x: singleArrPtr; Npoint : integer;
                    var mean, disp : single);

var i : integer;
begin
  mean:=x^[1];
  disp:=x^[1]*x^[1];
  for i:=2 to Npoint do begin
    mean:=mean+x^[i];
    disp:=disp+x^[i]*x^[i];
  end;
  mean:=mean/Npoint;
  disp:=disp/(Npoint-1.0);
  disp:=disp-mean*mean;
end;

procedure statistics(x: singleArrPtr; Npoint : integer;
                    var xmin, xmax, mean, disp, sko, Asymmetry, Excess : single);
var i: integer; r, rr: single;
begin
  xmin:=x^[1]; xmax:=x^[1]; mean:=0.0; disp:=0.0;
  Asymmetry:=0.0; Excess:=0.0;
  for i:=1 to Npoint do begin
    if x^[i] > xmax then xmax:=x^[i];
    if x^[i] < xmin then xmin:=x^[i];
    mean:=mean+x^[i];
  end;

```

```

mean:=mean/Npoint;
for i:=1 to Npoint do begin
  r:=x^[i]-mean; rr:=r*r;
  disp:=disp+rr;
  Asymmetry:=Asymmetry+rr*r;
  Excess:=Excess+rr*rr;
end;
disp:=disp/(Npoint-1.0);
sko:=sqrt(abs(disp));
Asymmetry:=Asymmetry/Npoint;
Asymmetry:=Asymmetry*Asymmetry/(disp*disp*disp);
Excess:=Excess/(Npoint*disp*disp);
end;
procedure ExpGen(x: singleArrPtr; Npoint : integer; l:single);
var i: integer;
begin
  for i:=1 to Npoint do begin
    x0:=x0*37; x0:=x0-Trunc(x0); x^[i]:=-ln(x0)/l;
  end;
end;
procedure LnGen(x: singleArrPtr; Npoint : integer; m,s : single);
var i,j : integer; r : array [1..2] of single; p : single;
begin p:=2*pi;
  for i:=1 to Npoint do begin
    for j:=1 to 2 do begin
      x0:=x0*37; x0:=x0-Trunc(x0); r[j]:=x0;
    end;
    x^[i]:=m*exp(s*sqrt(-p*ln(r[1]))*cos(p*r[2]));
  end;
end;
procedure WabGen(x: singleArrPtr; Npoint : integer; a,b : single);
var i: integer;
begin
  for i:=1 to Npoint do begin x0:=x0*37; x0:=x0-Trunc(x0);
    x^[i]:=expr(-ln(x0),1./a)/b;
  end;
end;
procedure RelGen(x: singleArrPtr; Npoint : integer; l:single);
var i: integer;
begin
  for i:=1 to Npoint do begin x0:=x0*37; x0:=x0-Trunc(x0);

```

```

    x^[i]:=sqrt(-2*ln(x0))*l;
end;
end;
procedure NormGen(x: singleArrPtr; Npoint : integer; m,s : single);
var i,j : integer;
begin
  for i:=1 to Npoint do begin X^[i]:=-6.0;
    for j:=1 to 12 do begin x0:=x0*37; x0:=x0-Trunc(x0);
      x^[i]:=x^[i]+x0;
    end;
    x^[i]:=m+s*x^[i];
  end;
end;
procedure abcf(a,b,c : singleArrPtr; Npoint, bcf: integer);
var i,j,n : integer; s0,s1,s2,m1,m2 : single;
begin
  n:=Npoint div 2;
  mean_disp(a,Npoint,m1,s1);
  m2:=m1;
  s2:=s1;
  if bcf<>0 then mean_disp(b,Npoint,m2,s2);
  s0:=s1*s2*n;
  if s0<>0 then begin
    for i:=1 to n do begin
      c^[i]:=0;
      for j:=1 to n do c^[i]:=c^[i]+(a^[j]-m1)*(b^[j+i-1]-m2);
    end;
    if bcf=0 then s0:=c^[1];
    for i:=1 to n do c^[i]:=c^[i]/s0;
    for i:=n+1 to Npoint do c^[i]:=c^[Npoint-i+1];
  end else Writeln(' ABCF: Error - Bad dispersion. ');
end;
END.

```

### Пример программы моделирования

```

uses tpStat, addGraph;
const N = 4096; Nh=64;
var
  M, i : integer;
  x,y,z,a,s : array [1..N] of single;
  hi : array [1..Nh] of longint;
  cs : array [1..N] of complex;
  xmin, xmax, mean, disp, sko, Asymmetry, excess, R, C, dt : single;
begin
  x0:=0.3665543;
  NormGen(@z,N,0,1); { Моделирование случайных чисел }
  HIST (@hi,Nh,@z,N); { Построение гистограммы }
  BoxDraw(@hi,Nh,'Гистограмма'){ См. рис. П2.1 }
  { Статистический анализ }
  statistics(@z,N, xmin, xmax, mean, disp, sko, Asymmetry, excess);
  writeln('Minimum=',xmin:0:3, ' Maximum=',xmax:0:3);
  writeln('Mean=',mean:0:3, ' Disp=',disp:0:3, ' С.К.О.=',sko:0:3);
  writeln('Asymmetry=',Asymmetry:0:3,' Excess=',excess:0:3);
  for i:=1 to N do x[i]:=sin(i*0.1)+z[i]; { Сигнал sin(*)+шум }
  LineDraw(@x,512,'Сигнал+шум'); { См. рис. П2.1 }
  for i:=1 to N do Cmplx(x[i],0.0,cs[i]); { Комплексный массив cs=x+j*0 }
  fft(@cs,N,forwd); { Прямое преобразование Фурье (БПФ) }
  for i:=1 to N do s[i]:=cabs(cs[i]); { Модуль спектра }
  StickDraw (@s,512,'Спектр сигнала'); { См. рис. П2.1 }
  abcf(@x,@x,@a,N,0); { Автокорреляционная функция сигнала }
  LineDraw(@a,512,'Автокорреляционная ф-я'); { См. рис. П2.1 }
  { Моделирование RC-фильтра }
  R:=1000.0; C:=1.0E-6; dt:=0.001; y[1]:=0.0;
  for i:=2 to N do y[i]:=(x[i]*dt+y[i-1]*R*C)/(dt+R*C);
  LineDraw(@y,512,'Выход RC-фильтра'); { См. рис. П2.1 }
end.

```

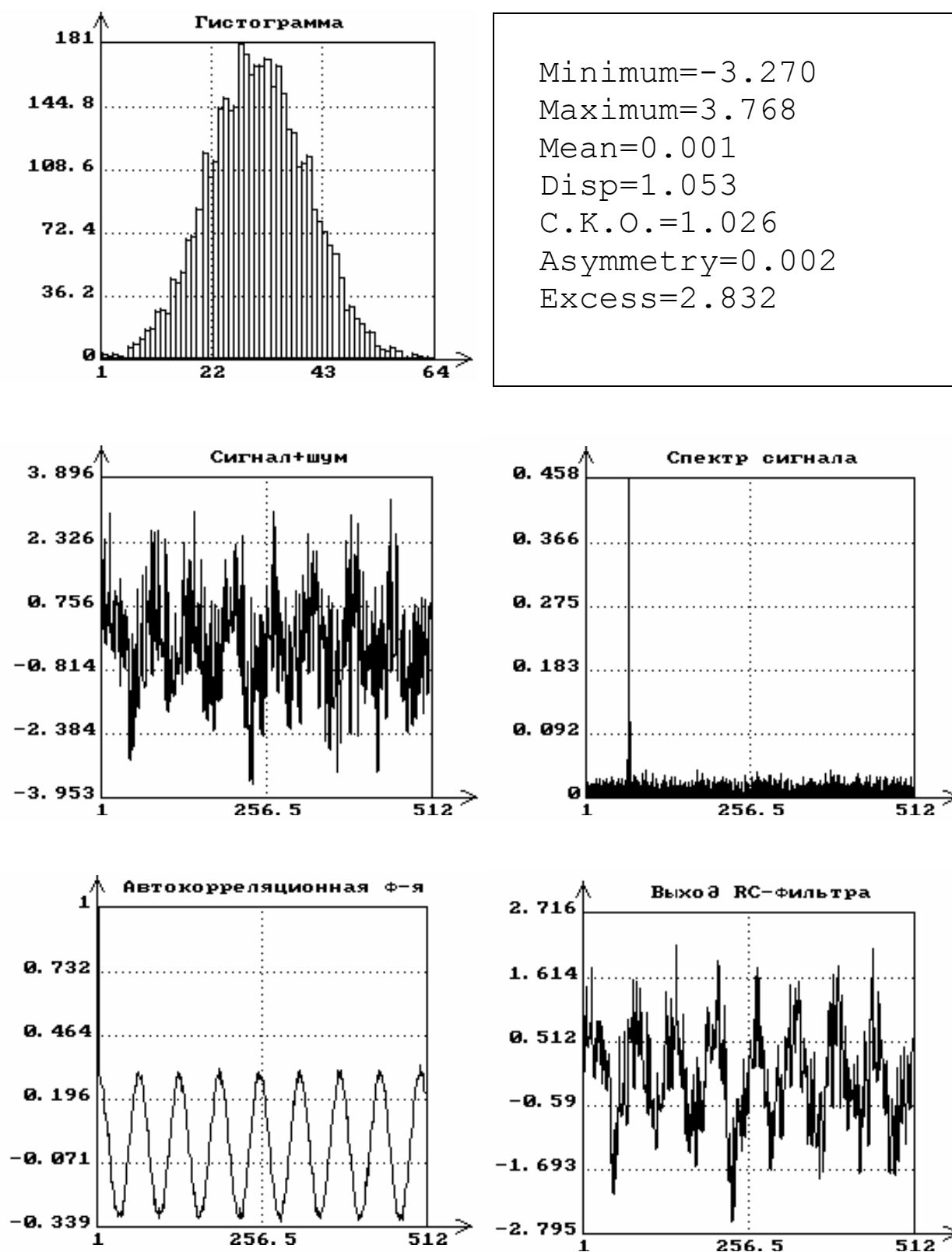


Рис. П2.1. Результаты моделирования

### Список используемой и рекомендуемой литературы

1. Баскаков С.И. Радиотехнические цепи и сигналы. – М.: Высшая школа, 1988. – 448 с.
2. Справочник по математике для научных работников и инженеров. Корн Г., Корн Т. – М.: Наука, 1984. – 831 с.
3. Бондарев В.Н., Трестер Г., Чернега В.С. Цифровая обработка сигналов: методы и средства. – Севастополь: Изд-во СевГТУ, 1999. – 398 с.
4. Вентцель Е.С., Овчаров Л.А. Теория вероятностей и ее инженерные приложения. – М.: Высшая школа, 2000. – 480 с.
5. Вентцель Е.С., Овчаров Л.А. Теория случайных процессов и ее инженерные приложения. – М.: Высшая школа, 2000. – 383 с.
6. Бендат Дж., Пирсол А. Прикладной анализ случайных данных: Пер. с англ. – М.: Мир, 1989. – 540 с.
7. Пугачев В.С. Теория случайных функций. – М.: ФизматГиз, 1960. – 883 с.
8. Вагапов В.Б. Основы автоматизации радиоэлектронных систем. – К.: Вища школа, 1995. – 358 с.
9. Быков В.В. Цифровое моделирование в статистической радиотехнике. – М.: Сов. радио, 1971. – 328 с.

## Содержание

Введение	3
<b>1. Методы теории подобия и моделирования при проектировании ЭА</b>	<b>4</b>
1.1. Понятия подобия и моделирования явлений	4
1.2. Классификация методов моделирования	5
1.3. Теоремы теории подобия и применение их выводов при моделировании и экспериментах	8
1.4. Способы определения критериев подобия явлений	11
1.5. Понятие об автомодельности явлений	15
1.6. Принципы применения методов теории подобия и моделирования	16
<b>2. Методы аналогового машинного моделирования</b>	<b>17</b>
2.1. Принципы построения аналоговых вычислительных машин	18
2.2. Машинные переменные	23
2.3. Методика моделирования на АВМ	25
2.4. Пример решения дифференциальных уравнений на АВМ	27
2.5. Моделирование детерминированных сигналов на АВМ	29
2.6. Моделирование случайных процессов на АВМ	31
<b>3. Методы цифрового моделирования на ЭВМ</b>	<b>33</b>
3.1. Основные принципы цифрового моделирования	33
3.2. Моделирование детерминированных сигналов на ЭЦВМ	35
3.3. Моделирование случайных сигналов и воздействий	39
3.3.1. Имитация значений стандартной равномерно распределенной величины	40
3.3.2. Случайные события и дискретные случайные величины	42
3.3.3. Марковские случайные процессы	44
3.3.4. Непрерывные случайные величины	48
3.3.5. Случайные векторы	52
3.3.6. Стационарные случайные процессы	55
3.3.7. Методы статистического анализа данных	57
3.4. Методы моделирования линейных динамических систем	62
3.4.1. Прямой алгоритм цифрового моделирования	63
3.4.2. Спектральный метод цифрового моделирования	65
3.4.3. Рекуррентный алгоритм цифрового моделирования	68
Приложение 1. Библиотека процедур моделирования и статистической обработки данных	76
Приложение 2. Пример программы моделирования	84
Список используемой и рекомендуемой литературы	86