

Санкт-Петербургский государственный технический университет

---

Е.Н.Бендерская, Д.Н.Колесников, В.И.Пахомова

# ФУНКЦИОНАЛЬНАЯ ДИАГНОСТИКА СИСТЕМ УПРАВЛЕНИЯ

Учебное пособие



Санкт-Петербург  
2000



**Министерство образования Российской Федерации**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Е.Н. Бендерская, Д.Н. Колесников, В.И. Пахомова**

# **Функциональная диагностика систем управления**

**Учебное пособие**

**Под ред. д-ра техн. наук Д.Н. Колесникова**

**Санкт-Петербург  
2000**

**УДК 681.326.74**

**Функциональная диагностика систем управления. Учебное пособие. / Е.Н. Бендерская, Д.Н. Колесников, В.И. Пахомова / Санкт-Петербургский государственный технический Университет, СПб, 2000.**

Пособие предназначено для изучения курса «Функциональная диагностика систем управления» при подготовке инженеров по специальности 210100 – «Управление и информатика в технических системах», а также магистров по направлению «Автоматизация и управление».

В нем рассмотрены основные принципы, подходы и методы построения алгоритмического обеспечения систем функционального диагностирования, а также предложены программы научно-исследовательских работ с использованием пакета MATLAB для изучения методов функциональной диагностики.

Пособие также может быть использовано студентами других специальностей при изучении курсов, связанных с применением методов технической диагностики.

Печатается по решению редакционно-издательского совета Санкт-Петербургского государственного технического университета.

## Содержание

	стр
Введение .....	5
1. Задачи и методы функционального диагностирования.....	7
1.1. Основные понятия и определения.....	7
1.2. Классификация методов функционального диагностирования.....	9
1.3. Принципы построения инвариантов.....	12
2. Методы контроля правильности функционирования динамических систем.....	21
2.1. Метод избыточных переменных.....	21
2.2. Диагностирование систем с использованием моделей.....	27
2.2.1. Редукция моделей.....	27
2.2.2. Проверка правильности функционирования в отдельных режимах.....	29
2.3. Контроль с использованием фильтра Калмана.....	31
2.3.1. Общие уравнения фильтра.....	31
2.3.2. Модели дефектов.....	35
2.3.3. Структура фильтра Калмана для моделей авторегрессии.....	37
2.3.4. Использование обновляющего процесса для обнаружения дефектов.....	41
2.4. Контроль с помощью обнаруживающих фильтров.....	43
3. Методы принятия решений в системах функционального диагностирования.....	46
3.1. Проверка статистических гипотез.....	46
3.2. Допусковые алгоритмы.....	52
3.3. Обнаружение разладки случайных процессов.....	53
3.3.1. Алгоритмы обнаружения.....	53
3.3.2. Принципы настройки алгоритмов обнаружения .....	58
3.4. Принятие решений с использованием доверительных и толерантных интервалов .....	61
3.5. Распознавание образов .....	64
4. Решение задач функциональной диагностики с использованием аппарата нейронных сетей.....	69
4.1. Основные понятия нейронных сетей (НС).....	69
4.2. Типовые структуры НС.....	74
4.3. Основные алгоритмы обучения.....	77
4.4. Подходы к формированию структуры НС и определение ее параметров.....	83
4.5. Моделирование работы НС.....	86
4.6. Диагностика многоканальной информационно-измерительной системы с использованием нейронной сети.....	96

5. Содержание программы работ по изучению методов функционального диагностирования .....	101
5.1. Моделирование случайных факторов с использованием MATLAB .....	101
5.2. Построение структурных инвариантов .....	106
5.3. Применение теории распознавания образов к диагностике состояния структурно-избыточных систем .....	110
5.4. Обнаружение дефектов с использованием метода избыточных переменных .....	113
5.5. Контроль с использованием фильтра Калмана.....	116
5.6. Отбраковка аномальных измерений в потоках диагностической информации .....	118
5.7. Использование аналитической избыточности для диагностирования двухканальных систем.....	123
5.8. Применение нейронных сетей в задачах диагностики.....	126
<b>Приложения к разделу 5</b>	
5.1. Программа моделирования случайной величины и вычисление ее характеристик.....	129
5.2. Программа моделирования случайных векторов.....	129
5.3. Программа моделирования случайного процесса, представленного моделью AP(1).....	130
5.4. Программа распознавания, основанная на геометрическом подходе.....	131
5.5. Программа распознавания, основанная на статистическом подходе.....	133
5.6. Программа моделирования скалярного фильтра Калмана.....	135
5.7. Программа моделирования медианного фильтра.....	139
5.8. Программа моделирования и исследования случайных величин при отсутствии и наличии аномальных измерений.....	140
<b>Литература.....</b>	142

## **Введение**

В настоящее время в отечественной и зарубежной литературе имеется большое число публикаций по различным вопросам диагностики технических систем и устройств, так как она является своеобразным индикатором и гарантом качества и надежности новой техники. Основное содержание технической диагностики составляет теория, методы и средства обнаружения и поиска дефектов.

Среди существующих публикаций имеются значительные работы по теоретическим аспектам диагностики, принципам построения диагностических систем и их аппаратурной реализации. Однако большинство публикаций имеет монографический характер, уровень и форма изложения в этих публикациях ориентированы, как правило, на научных работников, занимающихся исследованием и разработкой диагностических систем. Данное учебное пособие не ставит целью дать более или менее полное изложение всех вопросов, которые принято относить к технической диагностике.

Главное внимание в пособии уделено вопросам функционального диагностирования систем управления, т.е. проверке правильности их функционирования в процессе выполнения ими основных функций, предусмотренных назначением этих систем.

Данный круг вопросов приобретает особую актуальность в настоящее время, ибо многие аварийные ситуации на авиационном, морском, железнодорожном транспорте, на атомных электростанциях и других комплексах могут быть значительно уменьшены или совсем исключены, при своевременных и достоверных обнаружении, локализации и коррекции дефектов в процессе функционирования.

Учебно-методический характер работы, а также ее объем предопределили и критерий отбора материала.

Материал основан, в первую очередь, на многолетнем опыте исследований и проведения научно-исследовательских разработок, а также на опыте чтения лекций по проблемам функциональной диагностики для студентов старших курсов, специализирующихся в области управления техническими системами.

Пособие состоит из пяти разделов.

Раздел 1 посвящен общим вопросам, определениям, понятиям и принципам функционального диагностирования.

В разделе 2 рассматриваются конкретные методы и алгоритмы для построения систем функционального диагностирования.

Раздел 3 раскрывает основные подходы к принятию решений с использованием существующих методов математической статистики и случайных процессов.

Раздел 4 нацелен на анализ возможных подходов в применении нейронных сетей в задачах функциональной диагностики.

В разделе 5 предлагаются конкретные работы и их программы для изучения вопросов функциональной диагностики с использованием системы MATLAB.

Работа над пособием была распределена следующим образом:

- разделы 1–3 написаны Колесниковым Д.Н., – раздел 4 – Бендерской Е.Н.,
- раздел 5 – Колесниковым Д.Н. и Пахомовой В.И.

Хотим надеяться, что данное учебное пособие поможет читателю познакомиться с основными подходами к решению задач функционального диагностирования, чтобы при необходимости более обоснованно подойти к выбору путей и методов решения указанной проблемы.

# 1. Задачи и методы функционального диагностирования

## 1.1. Основные понятия и определения

В соответствии с [1] техническая диагностика – отрасль знаний, исследующая принципы, способы и средства оценки состояния технических объектов и направленная на повышение эффективности и качества работы исследуемых систем. Ее основной задачей является определение технического состояния проверяемого объекта с точки зрения правильности выполнения возложенных на него функций.

Функционирование объекта сопровождается возможным возникновением дефектов.

Под дефектом следует понимать несоответствие свойств рассматриваемого объекта диагностирования (ОД) заданным требованиям. Можно выделить три типа задач, решаемых при диагностировании:

- проверка исправности ОД,
- работоспособности,
- и правильности функционирования.

*Проверка исправности ОД* требует проведения полного комплекса испытаний для проверки соответствия всем требованиям нормативно-технической документации и поэтому наиболее сложна. Система считается исправной, если в ней нет ни одного дефекта.

*Проверка работоспособности ОД* предполагает анализ правильности выполнения всех заданных функций во всех режимах работы с сохранением значений заданных параметров и признаков в заданных границах, поэтому она также достаточно трудоемка.

*Проверка правильности функционирования* выполняется в рабочем режиме и оценивает работу системы в данный момент времени при конкретном рабочем входном сигнале. При этом возможно наличие дефектов, не проявляющихся в данном режиме или при данном входном сигнале. Достоинством проверки является оперативность получения информации о переходе объекта в неисправное состояние, недостатком – невысокая полнота контроля. Такая проверка целесообразна для динамических объектов, работающих длительное время, например, для бортовых систем управления, систем ядерной энергетики и др.

Основные этапы организации и проведения диагностических экспериментов:

*1. Формирование математической модели* ОД с учетом основных режимов и особенностей функционирования.

**2. Определение цели диагностирования.** В дальнейшем рассматривается только задача проверки правильности функционирования, которая может быть представлена различными целевыми установками:

- обнаружение дефекта – принятие решения о наличии или отсутствии дефекта;
- локализация дефекта – принятие решения об указании места дефекта;
- идентификация дефекта – принятие решения о виде дефекта;
- прогнозирование – принятие решения о степени развития дефекта;
- коррекция – устранение влияния дефекта на работу ОД.

**3. Построение диагностической модели,** отражающей цель исследования. Диагностическая модель – это математическая модель ОД, учитывающая возможные источники дефектов и погрешностей.

Различают дефекты: устойчивые (отказы), неустойчивые (сбои), одиночные и кратные, физические и логические, искажения параметров объекта и его входных или внутренних сигналов, внезапные и постепенные, первичные и вторичные ошибки и др.

Выбор той или иной модели дефектов производится с учетом конкретных особенностей ОД и опыта его эксплуатации.

**4. Анализ диагностируемости.** Построенная модель используется для выявления множества ненаблюдаемых дефектов, определения классов эквивалентных и неразличимых дефектов, оценки чувствительности диагностирования. При этом речь идет о потенциальных характеристиках диагностируемости, без привязки к конкретному методу или набору диагностических признаков.

**5. Выбор диагностических признаков (ДП).** Под диагностическими признаками понимаются характеристики объекта, используемые для определения его технического состояния.

Выбор совокупности непосредственно измеряемых диагностических признаков представляет собой ответственный этап, от которого во многом зависит качество, точность и эффективность всего эксперимента.

Требования, которым должны удовлетворять ДП включают: *измеряемость, информативность, инвариантность*. Последнее означает, что они должны иметь малую чувствительность к шумам и другим мешающим воздействиям.

**6. Выбор метода диагностирования и синтез устройства диагностирования.** Традиционно в технической диагностике различают функциональное и тестовое диагностирование. Как правило, первое из них выполняется в рабочем режиме, поэтому в ряде работ [ 2 ] оно определяется как рабочее диагностирование; второе реализуется в специальном режиме при подаче на вход тестовых воздействий.

**7. Синтез устройства диагностирования.** Данный этап предполагает создание алгоритмической структуры для реализации методов диагностирования, выбранных в п. 6. Так, например, типичным примером тестового диагностирования являются методы оценивания параметров, развитые в теории идентификации. Диагностирование проводится путем сравнения экспериментальных оценок параметров с их эталонными значениями.

**8. Обработка полученной диагностической информации и формирование окончательного решения.** Цель обработки – отбраковка недостоверных измерений, фильтрация помех, пересчеты данных прямых измерений в результаты диагностирования и оценка достоверности полученных результатов.

**9. Использование результатов диагностирования.** По результатам диагностирования может приниматься решение о реконфигурации ОД, либо изменении режима его работы, ремонте или замене неисправного блока и т.п.

## 1.2. Классификация методов функционального диагностирования

К основным классификационным признакам в соответствии с [1] относят:

- 2) вид математической модели;
- 3) режим диагностирования;
- 4) модель дефекта;
- 5) характер диагностических признаков;
- 6) принцип диагностирования;
- 7) правило принятия решений.

**1.** В зависимости от вида математической модели различают:

- методы диагностирования во временной и частотной областях;
- статические и динамические методы;
- детерминированные и вероятностные методы.

**2.** В зависимости от режима:

- функциональное диагностирование;
- тестовое диагностирование.

**3.** Существуют две наиболее распространенных модели дефектов:

- искажение сигналов ОД;
- искажение параметров ОД.

В соответствии с этим существуют две группы методов:

- оценивание переменных состояния и выходных сигналов ОД;
- оценивание параметров системы (коэффициентов математической модели.)

4. По характеру диагностических признаков (параметров или сигналов) различают методы диагностирования:

- в пространстве параметров;
- в пространстве сигналов.

В первом случае измеряются текущие значения параметров ОД и оцениваются их отклонения от номинального значения.

Во втором случае проверяются отклонения выходных сигналов от теоретических значений.

5. Анализ существующих методов диагностирования позволяет выделить три функциональных принципа:

- А. Применение теории инвариантов.
- Б. Применение теории моделей.
- В. Использование избыточности.

А. Первый принцип сводится к выявлению некоторых характеристик объекта, остающихся неизменными при нормальном функционировании ОД и изменяющихся при появлении дефектов. Далее эти характеристики (инварианты) используются в качестве прямых или косвенных диагностических признаков.

Б. Второй принцип опирается на использование моделей проверяемого объекта. С наибольшей очевидностью он проявляется в таких хорошо известных методах, как контроль на основе дублирования и резервирования, представляющих собой частные случаи контроля с помощью моделей, подключаемых параллельно объекту.

В. Третий принцип связан с использованием различных видов избыточности:

- структурной и функциональной (наличие избыточного числа устройств);
- естественной (наличие аналитических зависимостей между входом и выходом системы);
- искусственной (введение избыточных переменных);
- информационной (наличие априорной информации либо коррелированность измеряемых сигналов и др.).

Как отмечено в [1] указанные выше принципы диагностирования не являются изолированными, а выступают в тесном взаимодействии, образуя единую триаду (рис.1.1)



Рис 1.1.

Следует отметить приоритетный характер первого принципа как обобщающего другие и позволяющего с единых позиций представлять методологию диагностирования динамических систем.

#### 6. Правило принятия решений.

Как видно из предыдущего, при построении систем диагностирования должно быть сформулировано пространство диагностических признаков (ДП), в котором принимается окончательное решение о состоянии наблюдаемого объекта.

Алгоритмы формирования решений в этом пространстве могут быть представлены следующими классами:

##### А. Состояние ОД не изменяется в процессе диагностирования:

- допусковый (измеряемый или вычисляемый диагностический признак сравнивается с заданным допуском) [ 3 ].
- принятие решений на основе проверки статистических гипотез (непоследовательные и последовательные алгоритмы) [ 4 ].
- алгоритмы распознавания образов [ 5 ].

##### Б. Состояние ОД изменяется в процессе диагностирования. Алгоритмы принятия решений в этом случае представлены классом процедур обнаружения разладки случайных последовательностей [ 6 ].

В дальнейшем при рассмотрении различных методов диагностирования основное внимание будет уделено двум основным вопросам: принципам диагностирования и правилам формирования решений. При этом рассмотрение трех указанных выше принципов диагностирования будет представлено с общих методологических позиций, использующих теорию инвариантов.

### 1.3. Принципы построения инвариантов.

Под инвариантом в дальнейшем в данном пособии понимается некоторое алгебраическое соотношение (условие связи) между входными и выходными или только выходными сигналами ОД, которое однозначно выполняется при правильном функционировании ОД и которое нарушается, если в системе возникает дефект.

Понятие инварианта широко используется в различных областях научного знания: биологии, физике, математике, теории автоматического управления и др. [ 7 ].

На рис. 1.2. показано, как выглядит общая схема организации диагностирования применительно к системам управления на базе теории инвариантов. Оператор  $H$  на схеме – вспомогательный оператор, который формирует по информации от вектора входных сигналов  $u(t)$  и вектора выходных сигналов  $y(t) \in Y$  некоторый вектор  $z$  – вектор вспомогательных (избыточных) переменных. Сочетание операторов  $H$  и  $F$  создает устройство диагностирования (УД). УД на базе вектора  $y(t)$  составляет некоторое контрольное условие в виде инварианта  $\Delta = F(z, y(t))$ . Если оказывается, что  $\|\Delta\| < \delta$ , то ОД функционирует правильно, а если  $\|\Delta\| > \delta$ , то имеет место дефект, где  $\|x\|$  – норма вектора  $x$ . Если  $x$  – скалярная величина, то норма  $\|x\|$  переходит в модуль  $|x|$ .

Таким образом, при построении системы диагностирования происходит преобразование первичного сигнального пространства  $Y$  в пространство инвариантов  $\Delta$ , в котором рассматриваются и решаются задачи как обнаружения, так и локализации и идентификации дефектов.

*Общая схема организации диагностирования*

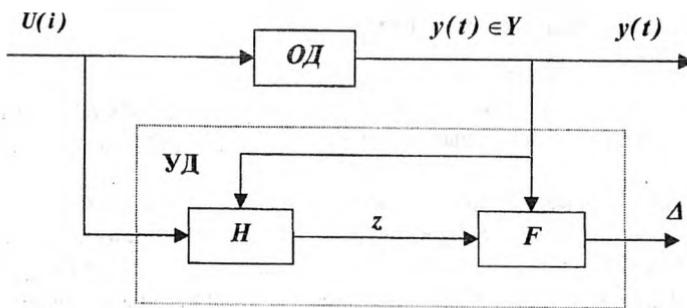


Рис. 1.2.

В зависимости от того, какие методы выбраны для построения инвариантов, можно говорить о различных методах функционального диагностирования:

- 1) использование структурной избыточности;
- 2) использование функциональной избыточности;
- 3) использование естественной избыточности;
- 4) использование искусственной избыточности (метод избыточных переменных);
- 5) использование математических моделей.

Рассмотрим кратко содержание перечисленных методов.

### 1.3.1. Использование структурной избыточности

Использование структурной избыточности подразумевает, что одновременно с основным ОД используются резервные каналы измерения или управления, информация от которых фигурирует при построении инвариантов. Обычно дополнительные каналы измерения или управления вводятся не только для диагностирования. Чаще это используется для повышения надежности и точности. Однако, одновременная обработка информации, поступающей из всех каналов, позволяет решать и задачи диагностирования.

Рассмотрим пример построения трехканального ОД при использовании структурной избыточности (рис. 1.3.).

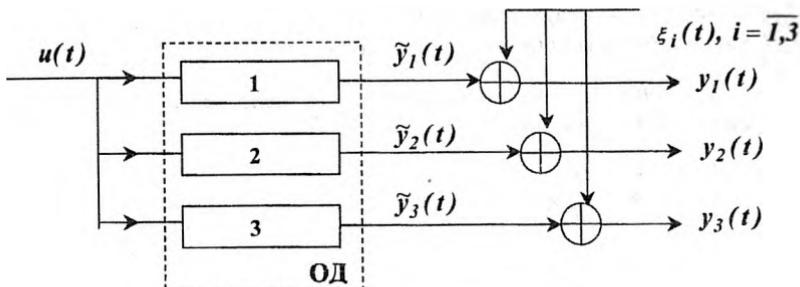


Рис. 1.3

Примем аддитивную модель выходного сигнала  $i$ -го канала:

$$y_i = \tilde{y}_i(t) + \zeta_i(t), \quad i = \overline{1, 3}, \quad (1.1)$$

где  $\tilde{y}_i$  – полезный выходной сигнал;

$\zeta_i(t)$  – случайная составляющая (возмущение).

Будем считать инвариантом такой вектор (в частном случае скаляр):

$$\Delta = F(y_1(t), y_2(t), y_3(t)), \quad (1.2)$$

который при правильном выполнении объектом своих задач является стационарным случайнм процессом; при изменении свойств ОД

стационарность нарушается, происходит изменение характеристик процесса (математического ожидания, дисперсии, корреляционной функции). В детерминированном случае, т.е. при отсутствии случайных возмущений, значение инварианта строго равно нулю при правильном функционировании и не равно нулю при наличии дефекта.

В данном случае можно использовать два способа построения инвариантов.

1. *Построение инварианта в виде вектора разности одноименных параметров.* При принятой аддитивной модели выходного сигнала, и, если считать, что  $\zeta_i(t)$  – стационарный случайный процесс, то каждый из инвариантов есть разность случайных составляющих соответствующих сигналов:

$$\begin{cases} \Delta_{12} = y_1(t) - y_2(t) = \zeta_1(t) - \zeta_2(t), \\ \Delta_{23} = y_2(t) - y_3(t) = \zeta_2(t) - \zeta_3(t), \\ \Delta_{31} = y_3(t) - y_1(t) = \zeta_3(t) - \zeta_1(t). \end{cases} \quad (1.3)$$

Появление дефекта в одном из каналов приводит к изменению свойств инварианта  $\Delta_{ij}$ . Для обнаружения дефекта в любом канале достаточно любых двух инвариантов из трех, а для локализации дефекта необходимо иметь вектор трех инвариантов.

2. *Формирование инвариантов с использованием обобщенных статистик  $T$  ( $y_1, y_2, y_3$ ):*

$$\begin{cases} \Delta_j = y_j - \frac{1}{3} \sum_{j=1}^3 y_j, \\ \Delta_j = y_j - med\{y_1, y_2, y_3\}, \\ \Delta_j = y_j - \sum_{j=1}^3 g_j y_j, \quad j = \overline{1,3}, \quad \sum_{j=1}^3 g_j = 1; \quad 0 < g_j < 1, \end{cases} \quad (1.4)$$

где  $\frac{1}{3} \sum_{j=1}^3 y_j$  – среднее арифметическое;

$med\{y_1, y_2, y_3\}$  – порядковое среднее;

$\sum_{j=1}^n g_j y_j$  – средневзвешенное с весовыми коэффициентами  $g_i$ ,  $g_i = \overline{1,3}$ .

Любой канал считается неисправным, если

$$|y_j - T(y_1, y_2, y_3)| \geq \delta, \quad (1.5)$$

где  $T$  – обобщенная статистика;

$\delta$  – некоторый порог.

### 1.3.2. Использование функциональной избыточности

В отличие от структурной избыточности функциональная избыточность предполагает, что сигнальная информация на выходе основного ОД может быть получена от другого канала управления или измерения, основанного на других физических принципах.

Пространство выходных сигналов второго (дополнительного) канала отличается от пространства выходных сигналов первого (основного) канала (рис. 1.4).

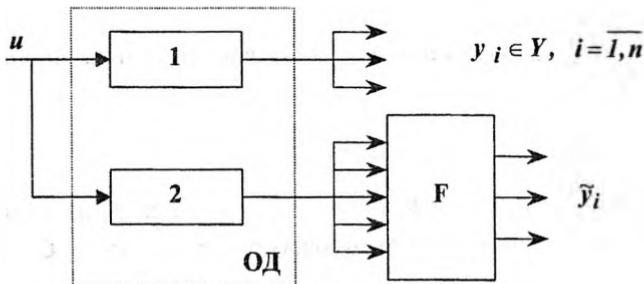


Рис. 1.4

При помощи некоторого функционального преобразования  $F$  могут быть получены аналогичные оценки сигналов основного канала. Инвариант в данном случае формируется как  $\Delta_i = y_i - \bar{y}_i, \quad i = \overline{1, n}$ . Метод, использующий функциональную избыточность, широко применяется в системах управления летательными аппаратами, в судовых системах автоматики, в энергетических системах, где измерение одного и того же параметра производится различными приборами, основанными на различных физических принципах.

### 1.3.3. Использование естественной избыточности.

Этот способ предполагает наличие некоторых математических соотношений, которые связывают выходные сигналы ОД и остаются постоянными в процессе правильного функционирования объекта:

$$F[y_i(t), i = \overline{1, n}] = C. \quad (1.6)$$

Возникновение дефекта приводит к нарушению связи и инвариант изменяет свои свойства. Поиск или определение таких соотношений требует подробного исследования ОД для выявления всех возможных взаимосвязей, существующих между сигналами  $y_i(t)$ .

Такого типа аналитические соотношения широко представлены в системах управления движущимися объектами, в энергетических системах и др. (например, аналитические взаимосвязи между параметрами, измеряемыми в различных системах координат).

### 1.3.4. Искусственная избыточность.

Данный способ предполагает формирование дополнительных контрольных соотношений, строго выполняемых при отсутствии дефектов. Чаще всего контрольные соотношения рассматриваются в виде линейных условий [8].

Рассмотрим пример (рис. 1.5).

Пусть имеется некоторый ОД, динамика которого описывается уравнением:

$$\begin{cases} \dot{x} = Ax + Bu, \\ Y = Cx \end{cases} \quad (1.7)$$

где  $u \in R^r$ ;  $y \in R^s$ ;  $x \in R^n$  – векторы входа, выхода и состояния,  $A$ ,  $B$ ,  $C$  – постоянные матрицы соответствующих размеров, причем  $C$  – единичная матрица.

Вводится скалярная искусственная переменная  $z$  так, чтобы выполнялось соотношение:

$$Mx + z = 0, \quad (1.8)$$

где  $M$  – матрица-строка постоянных коэффициентов.

Продифференцировав (1.8) и произведя преобразования, получим расширенную систему уравнений, описывающую динамику и переменных состояния ОД, и переменной  $z(t)$ :

$$\begin{cases} \dot{x} = Ax + Bu, \\ \dot{z} = -M\dot{x} - MBu. \end{cases} \quad (1.9)$$

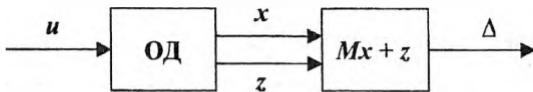


Рис. 1.5

При отсутствии дефекта в функционировании объекта инвариант  $\Delta = Mx + z$  строго равен нулю. Таким образом, рассмотренный подход может быть использован для обнаружения дефектов, а также может быть легко распространен на решение задачи локализации дефектов [ 7 ].

Идея метода избыточных переменных аналогична идеи избыточного кодирования, где аналогом инварианта является обычный синдром.

### 1.3.5. Метод, основанный на использовании математических моделей

Одним из этапов создания системы диагностирования является построение математических моделей ОД при отсутствии и при наличии всех видов дефектов. Модель ОД может быть представлена в виде передаточной функции, уравнений состояния, семейства характеристик и т.д.

Различают два способа формирования инвариантов на основе математических моделей:

- 1) на основе модели выходного сигнала;
- 2) на основе модели канала.

Пример построения инварианта первым способом представлен на рис. 1.6. Инвариант в данном случае – это разность выходного сигнала объекта  $y(t)$  и модели выходного сигнала  $\bar{y}(t)$ .

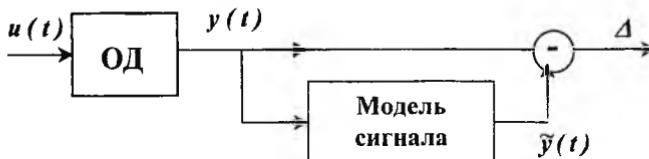


Рис 1.6.

Простейший случай построения инварианта с использованием модели выходного сигнала основан на выделении среднего и нормировании сигнала  $y_i(t)$  относительно среднего:

$$\Delta_i(t) = y_i(t) - M[y_i(t)] = y_i(t) - m_i(t), \quad i = \overline{1, n}, \quad (1.10)$$

где  $m_i(t)$  – математическое ожидание  $i$ -й компоненты выходного сигнала.

При усреднении выходного сигнала используют, например, экспоненциальное сглаживание. Вводится дискретизация времени:

$$t = k\Delta t, \quad k = 0, 1, 2 \dots$$

Сигналу  $y_i(t)$  сопоставляется сигнал  $\bar{y}_i(k)$ . Тогда сглаженный сигнал может быть представлен как:

$$\bar{y}_i(k) = (1 - \alpha) \bar{y}_i(k-1) + \alpha y_i(k), \quad (1.11)$$

где  $\alpha$  – некоторый коэффициент сглаживания,  $0 < \alpha < 1$ .

От коэффициента  $\alpha$  зависит эффект сглаживания, уровень подавления помех, а также уровень динамической ошибки.

Инвариант в данном случае формируется как  $\Delta(k) = y(k) - \bar{y}(k)$ .

В качестве моделей сигнала могут быть выбраны различные дискретные модели типа авторегрессии (AP), скользящего среднего (CC), авторегрессии со скользящим средним (APCC), интегрированные модели APCC.

В дальнейшем рассматриваем наиболее простой класс моделей – модели типа AP.

В соответствии с моделью AP дискретный сигнал  $y_i(k)$  можно представить в следующем виде [ 9 ]:

$$y_i(k) = \mu_i(k) + \sum_{l=1}^p a_{il} [y_i(k-l) - \mu_i(k)] + \beta_i z_i(k), \quad (1.12)$$

где  $\mu_i(k)$  – математическое ожидание  $i$ -ой компоненты сигнала;

$a_{il}$  – коэффициент авторегрессии  $l = \overline{1, p}$ ;

$p$  – порядок AP;

$\beta_i$  – остаточная дисперсия;

$z_i(k)$  – белый шум,

$z_i(k) \in N(0,1)$ .

Параметры AP-представления  $\mu_i, a_{il}, \beta_i$  оцениваются либо в

процессе предварительных экспериментов, либо на этапе, предшествующем этапу диагностирования. Строится модель ОД в виде AP, и в этом случае процесс диагностирования сочетается с процессом предварительного оценивания.

Наряду с моделями сигналов для построения инвариантов используют модели каналов ОД.

Пусть модель ОД, представленная на рис. 1.7а, задана в пространстве состояний:

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx. \end{cases}$$

Если объект и модель адекватны, то инвариант  $\Delta = y - y_m$  при правильном функционировании ОД является векторным стационарным

случайным процессом. Обычно модель не совсем адекватна объекту и не воспроизводит всех возмущений, которые действуют на ОД.

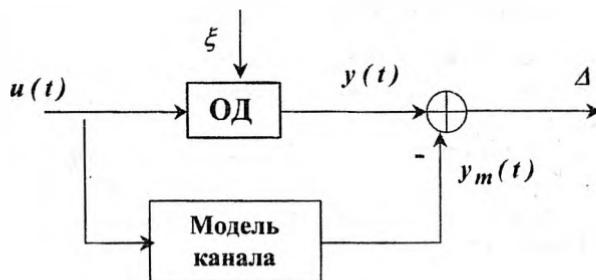


Рис. 1.7а.

Следовательно, при правильном функционировании ОД инвариант  $\Delta$  оказывается нестационарным процессом за счет накопления расхождений ОД и модели. Поэтому, при построении инвариантов включение модели проводится с учетом решения задачи оценивания вектора состояний с использованием фильтра Калмана (рис. 1.7.б) [ 8 ].

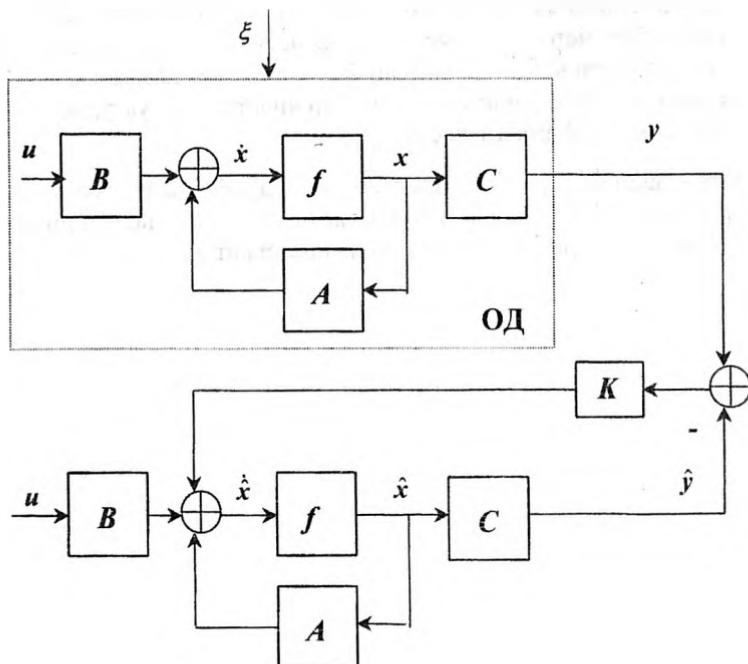


Рис. 1.7б.

Калмановская фильтрация использует идею прогнозирования состояния объекта. Прогнозируемое значение выходного сигнала сравнивается с измеренным, и невязка  $u - \hat{u}$  используется для коррекции оценки состояния. Коррекция производится в виде аддитивного сигнала, полученного при умножении невязки на корректирующий коэффициент (коэффициент усиления).

Таким образом, модель, включенная параллельно с ОД, отслеживает те возмущения, которые имеются в реальном объекте. Однако, вычислительная сложность устройства формирования инвариантов существенно увеличивается.

Рассмотрев различные подходы к преобразованию сигнального пространства в пространство инвариантов ( $Y \rightarrow \Delta$ ), мы можем сделать следующий вывод:

для исследуемого ОД должны быть выбраны принципы построения инвариантов, исходя из свойств и структуры объекта, в зависимости от характера задач, решаемых рассматриваемым ОД, и сформулированных целей. Причем при выборе метода диагностирования возможно появление и комбинированных инвариантов: с использованием структурно-функциональной избыточности, структурно-аналитической избыточности и др.

В следующем разделе рассмотрены более подробно основные подходы к построению алгоритмов диагностирования с использованием рассмотренных выше принципов построения инвариантов.

## 2. Методы контроля правильности функционирования динамических систем

### 2.1. Метод избыточных переменных

Общетеоретическое рассмотрение метода избыточных переменных для создания системы функционального диагностирования (СФД) изложено в работе [8]. Реализация метода зависит от того, в каком виде представлена математическая модель ОД. Рассмотрим две основные модификации.

#### *Модификация 1*

Предполагается, что наблюдаемый ОД – линейный стационарный и задан уравнением в пространстве состояний и наблюдений:

$$\begin{aligned}\dot{x} &= Ax + Bu, \\ y &= Cx,\end{aligned}\tag{2.1}$$

где  $x \in R^n$ ,  $u \in R^r$ ,  $y \in R^s$ ;  $A$ ,  $B$ ,  $C$  – постоянные матрицы.

Внутренние переменные  $x$  считаются недоступными для наблюдения.

Устройство диагностирования вырабатывает сигнал  $z$ , который при любом входном сигнале  $u(t)$  и в любой момент времени  $t$  удовлетворяет условию:

$$\Delta = M^T Y + z = 0, \tag{2.2}$$

где  $M^T = (m_1, m_2, \dots, m_s)$  – вектор-строка постоянных коэффициентов.

Если в результате непрерывного функционирования ОД произошло искажение его выходных сигналов по причине возникновения дефекта, это должно привести к нарушению контрольного уравнения (2.2) и появлению сигнала рассогласования  $\Delta(t) \neq 0$ .

Предполагая, что структура устройства диагностирования УД имеет вид, представленный на рис. 2.1, с учетом допущений, сделанных в [8], получены необходимые и достаточные условия существования УД порядка  $k$ :

$$\begin{aligned}1. \quad &[\alpha_0 + \gamma_0 M, \alpha_1 + \gamma_1 M, \dots, \alpha_{k-1} + \gamma_{k-1} M, M] \begin{bmatrix} C \\ CA \\ \vdots \\ CA^k \end{bmatrix} = 0; \\ 2. \quad &\beta_{k-1} = -MCB; \\ 3. \quad &\beta_{k-2} = -[MCA + (\alpha_{k-1} + \gamma_{k-1} M)CA]B; \\ \cdot \quad &\dots \\ k. \quad &\beta_0 = -[MC A^{k-1} + (\alpha_{k-1} + \gamma_{k-1} M)CA^{k-2} + \dots \\ &\quad \dots + (\alpha_2 + \gamma_2 M)CA + (\alpha_1 + \gamma_1 M)CB],\end{aligned}\tag{2.3}$$

где  $\alpha_i \in R^s$ ,  $\beta_i \in R^r$ ,  $M \in R^s$ ,  $\gamma_i$  – скалярные величины,  $i = \overline{0, k-1}$ .

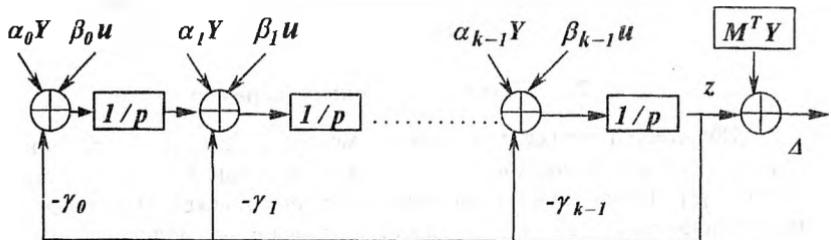


Рис 2.1

Синтез УД при условии, что на вход диагностирования блока УД можно подавать как входные, так и выходные сигналы объекта представляет наибольший практический интерес, так как приводит к построению более простых УД.

В [ 8 ] доказана следующая теорема: *минимальный порядок УД заданного вида при любом входном сигнале  $u(t)$  и любом векторе  $M$  равен индексу наблюдаемости проверяемой системы  $\nu_0$ .*

Заметим, что индексом наблюдаемости называется наименьшее целое положительное число  $\nu_0$ , для которого

$$\text{rank } [C^T, \dots, (A^T)^{\nu_0} C^T] = n. \quad (2.4)$$

Можно показать, что индекс наблюдаемости  $\nu_0$  системы произвольного порядка удовлетворяет следующему неравенству:

$$n/s \leq \nu_0 \leq n - s + 1, \quad (2.5)$$

где  $n$  – размерность вектора  $x$ ,  $s$  – размерность вектора  $y$ .

Методика синтеза выглядит следующим образом.

1. Выходной сигнал  $z$  может быть смоделирован в соответствии с рис.2.1 в виде:

$$z = \frac{1}{p} (\alpha_{k-1} y + \beta_{k-1} u) + \dots + \frac{1}{p^k} (\alpha_0 y + \beta_0 u). \quad (2.6)$$

Скалярные коэффициенты  $\gamma_i$ ,  $i = \overline{0, k-1}$  приняты для упрощения структуры УД равными нулю.

2. Находим минимальный порядок  $k$  для УД из соотношений (2.4) и (2.5).

3. Для выбранного вектора  $M$  из первого матричного уравнения системы (2.3) находим значения компонент векторов  $\alpha_l$ ,  $l = \overline{0, k-1}$ .

4. По найденным значениям векторов  $\alpha_1$  и  $M$  определяем из остальных уравнений системы (2.3) значения векторов  $\beta_l$ ,  $l = \overline{0, k-1}$ .

Для иллюстрации данной методики рассмотрим простой пример.

### Пример 2.1.

Пусть имеем систему:

$$\dot{x} = ax + bu,$$

$$y = cx,$$

Индекс наблюдаемости в соответствии с (2.5) равен  $v_0 = I$ , следовательно  $\kappa = 1$ .

Выходной сигнал  $z$  от УД выглядит следующим образом :

$$z = \frac{1}{p}(\alpha_0 y + \beta_0 u),$$

где  $\alpha_0$  и  $\beta_0$  – скалярные величины.

Система уравнений для нахождения  $\alpha_0$  имеет вид:

$$(\alpha_0, M) \begin{bmatrix} C \\ CA \end{bmatrix} = (\alpha_0, M) \begin{bmatrix} I \\ a \end{bmatrix} = 0; \quad c = 1.$$

При  $M = I$   $\alpha_0 = -a$ .

В соответствии с этим  $\beta_0 = -M C B = -b$  и выходной сигнал  $z$  от УД выглядит следующим образом:  $z = \frac{1}{p}(-ay - bu)$ .

Во временной форме общая модель функционирования ОД и УД выглядит следующим образом:

ОД:  $\dot{x} = ax + bu,$   
 $y = cx,$

УД:  $\dot{z} = -ay - bu; \quad \Delta = y + z.$

Если  $\Delta = 0$  – дефект отсутствует, в противном случае  $\Delta \neq 0$ , имеет место дефект.

### Модификация 2

Наблюдаемый линейный ОД представляет некоторое соединение звеньев с известными передаточными функциями, т.е. задан структурной схемой соединения звеньев с передаточными функциями  $W_1(p), \dots, W_n(p)$ .

УД вырабатывает избыточный сигнал  $z = y_{n+1}$ , удовлетворяющий условию (2.2). Сигнал  $y_{n+1}$  формируется следующим образом:

$$y_{n+1} = \sum_{i=1}^n (\alpha_{0i} x_i + \beta_{0i} y_i) + W_{n+1}(p) \sum_{i=1}^n (\alpha_{1i} x_i + \beta_{1i} y_i). \quad (2.7)$$

Коэффициенты  $\alpha_{0i}, \beta_{0i}, \alpha_{1i}, \beta_{1i}$  и передаточная функция  $W_{n+1}(p)$  должны быть выбраны так, чтобы выполнялось условие (2.2).

В [ 8 ] показано, что формирование  $y_{n+1}$  в соответствии с (2.7) возможно только для линейных систем удовлетворяющим следующим условиям:

$$W_i(p) = \frac{c_i + d_i W(p)}{a_i + b_i W(p)}, \quad i = \overline{1, n}, \quad (2.8)$$

где  $W(p)$  – произвольная передаточная функция;

$a_i, b_i, c_i, d_i$  – любые действительные числа, для которых:

$$\begin{vmatrix} a_i & b_i \\ c_i & d_i \end{vmatrix} \neq 0. \quad (2.9)$$

При анализе конкретных систем целесообразно в качестве  $W(p)$  выбрать передаточную функцию простого звена и непосредственной подстановкой выявить те звенья, для которых удовлетворяются условия (2.8) и (2.9).

Практически важный случай  $W = 1/p$ . Тогда система состоит из звеньев с передаточными функциями

$$W_i(p) = \frac{d_i + c_i p}{b_i + a_i p}, \quad i = \overline{1, n}, \quad (2.10)$$

соединенными произвольным образом.

В соответствии с (2.7), (2.2) имеем:

$$y_{n+1} = \sum_{i=1}^n (\alpha_{0i} x_i + \beta_{0i} y_i) + \frac{1}{p} \sum_{i=1}^n (\alpha_{1i} x_i + \beta_{1i} y_i), \quad (2.11)$$

где все постоянные коэффициенты определяются по формулам:

$$\alpha_{0i} = -m_i c_i, \quad \beta_{0i} = m_i (a_i - 1), \quad (2.12)$$

$$\alpha_{1i} = -m_i d_i, \quad \beta_{1i} = m_i b_i.$$

## Пример 2.2

Пусть система ОД имеет вид, представленный на рис 2.2.

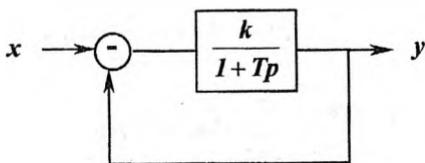


Рис. 2.2

Передаточная функция замкнутой системы имеет вид:

$$W(p) = \frac{y(p)}{x(p)} = \frac{k_1}{1 + T_1 p} ,$$

$$\text{где } k_1 = \frac{k}{k+1} ; \quad T_1 = \frac{T}{k+1} .$$

Из соотношения (2.10) следует, что

$$\begin{aligned} d &= k_1 ; & c &= 0 ; \\ b &= 1 ; & a &= T_1 . \end{aligned}$$

Соотношение (2.9) выполняется, и коэффициенты соотношения (2.11) при  $M = 1$  определяются из формул (2.12):

$$\begin{aligned} \alpha_0 &= 0, & \beta_0 &= T_1 - 1; \\ \alpha_1 &= -k_1, & \beta_1 &= 1. \end{aligned}$$

Избыточный сигнал  $z$  формируется в соответствии с (2.11) и равен:

$$z = y_{n+1} = (T_1 - 1)y + \frac{1}{p}(-k_1x + y).$$

Более общий случай рассматривается в [ 8 ] , когда элементы исходной системы не могут быть представлены в виде (2.8), т.е. дробно-функциональной функцией 1-го порядка относительно порождающей функции  $W(p)$  . В этом случае :

$$W_i(p) = \frac{a_{mi}W^m(p) + \dots + a_{1i}W(p) + a_{0i}}{b_{mi}W^m(p) + \dots + b_{1i}W(p) + b_{0i}} = \frac{A_i[W(p)]}{B_i[W(p)]}, \quad i = \overline{1, n}. \quad (2.13)$$

На коэффициенты  $a_{ij}$  и  $b_{ij}$  не накладывается никаких ограничений, кроме:

$$\begin{vmatrix} a_{1i} & a_{0i} \\ b_{1i} & b_{0i} \end{vmatrix} \neq 0$$

В [8] показано, что если передаточные функции звеньев системы имеют вид (2.13), то избыточная переменная всегда может быть получена из уравнения:

$$z = y_{n+1} = \sum_{i=1}^n (\alpha_{0i} x_i + \beta_{0i} y_i) + W(p) \sum_{i=1}^n (\alpha_{1i} x_i + \beta_{1i} y_i) + \dots + W^m(p) \sum_{i=1}^m (\alpha_{mi} x_i + \beta_{mi} y_i). \quad (2.14)$$

Из соотношения (2.2) могут быть получены следующие формулы для вычисления коэффициентов:

$$\begin{aligned} \alpha_{0i} &= -m_i a_{0i}; \quad \beta_{0i} = m_i (b_{0i} - 1); \\ \alpha_{j0i} &= -m_i a_{ji}; \quad \beta_{j0i} = m_i b_{ji}; \quad i = \overline{1, n}; \quad j = \overline{1, m}. \end{aligned} \quad (2.15)$$

### Пример 2.3

Структура ОД представлена на рис 2.3. Передаточная функция замкнутой системы выглядит следующим образом:

$$W(p) = \frac{k}{k + 1 + T_1 p + T_2 p^2}.$$

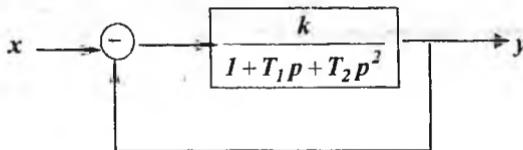


Рис. 2.3

Пусть базовая функция  $W(p) = p$ , тогда в соответствии с (2.13) имеем:

$$W_i(p) = \frac{a_2 W^2(p) + a_1 W(p) + a_0}{b_2 W^2(p) + b_1 W(p) + b_0},$$

где  $a_2 = 0$ ,  $a_1 = 0$ ,  $a_0 = k$ ,  $b_2 = T_2$ ,  $b_1 = T_1$ ,  $b_0 = k + 1$ .

В соответствии с (2.14) имеем:

$$y_{n+1} = (\alpha_0 x + \beta_0 y) + p (\alpha_1 x + \beta_1 y) + p^2 (\alpha_2 x + \beta_2 y),$$

где  $m = 1$ ,  $\alpha_0 = -k$ ,  $\alpha_1 = 0$ ,  $\alpha_2 = 0$ ,  $\beta_0 = k$ ,  $\beta_1 = T_1$ ,  $\beta_2 = T_2$ .

Сигнал  $z$ , сформированный УД, выглядит следующим образом:

$$z = -kx + ky + T_1 py + T_2 p^2 y.$$

## 2.2. Диагностирование систем с использованием моделей

### 2.2.1. Редукция моделей

Одним из универсальных и распространенных подходов к организации функционального контроля является использование моделей, подключенных параллельно ОД. Известно, что эффективность контроля во многом зависит от сложности УД, поэтому центральной проблемой при синтезе УД является обеспечение простоты моделей.

В теории динамических систем выделено три подхода к построению упрощенных моделей:

- дискретизация,
- линеаризация,
- редукция.

В дальнейшем хотелось более подробно остановиться на третьем подходе, так как он позволяет уменьшить размерность исходной задачи. При этом термин «редукция» связан с понижением порядка математического описания ОД. К настоящему времени известно множество методов редукции и аппроксимации линейных динамических систем. В соответствии с [ 1 ] их можно условно разделить на шесть классов:

1. Упрощенная модель строится на основе физических и инженерных соображений, при этом отбрасываются некоторые параметры или переменные, слабо влияющие на динамику объекта.
2. Методы редукции с использованием хорошо развитой теории идентификации [ 10 ].
3. Для этого класса характерна строгая постановка задачи редукции, при которой требуется построить модель оптимальную в смысле заданного критерия. При этом из соображений простоты чаще всего ограничиваются рассмотрением стандартных входных сигналов в виде дельта-функции или единичного скачка, что приводит к задаче аппроксимации весовой или переходной функции во временной или частотной области.
4. Данный класс методов назван по имени французского математика А.Паде. Принцип аппроксимации Паде состоит в сопоставлении разложений исходной и редуцированной моделей передаточных функций в ряды Тейлора и приравнивании возможно большего числа начальных коэффициентов этих разложений.
5. Проведение редукции моделей с сохранением различных типов инвариантов. Так, например, если в набор базисных инвариантов включить инварианты, отвечающие за сохранение устойчивости

(коэффициенты Раусса или корни характеристического полинома), то будет гарантирована устойчивость редуцированной модели.

6. Методы редукции данного класса опираются на описание в пространстве состояний. В соответствии с ними сначала находят матрицы  $A$ ,  $B$ ,  $C$  описания исходной системы в соответствии с (2.1) в некотором базисе пространства состояний, после чего редукция осуществляется путем отбрасывания части переменных состояния. Это эквивалентно вычеркиванию соответствующих строк и столбцов матриц  $A$ ,  $B$ ,  $C$  описания (2.1) и уменьшению их размеров, что приводит к получению редуцированных матриц  $A_p$ ,  $B_p$ ,  $C_p$ .

#### Пример 2.4

Пусть требуется организовать контроль с помощью редуцированной модели для проверки правильности функционирования контура управления самолетом по каналу высоты.

Указанный контур, включающий управляющий объект, сервопривод управления рулем, датчик высоты и регулятор, характеризуется передаточной функцией девятого порядка:

$$W(p) = \frac{B(p)}{A(p)} \quad (2.16)$$

где  $B(p) = -2.406 p^5 - 11.691 p^4 - 4.3977 p^3 + 2.2681 p^2 + 0.3221 p + 0.0080$ ;

$$\begin{aligned} A(P) = & p^9 + 19.275 p^8 + 95.272 p^7 + 224.86 p^6 + 313.52 p^5 + 260.36 p^4 + \\ & + 130.352 p^3 + 34.360 p^2 + 3.42766 p + 0.0837495. \end{aligned}$$

Данная передаточная функция получена линеаризацией уравнений движения самолета в окрестности номинальной траектории.

Для построения редуцированной модели используем модальный подход, основанный на сокращении близких нулей и полюсов системы. В соответствии с ним числитель и знаменатель передаточной функции (ПФ) раскладываем на простейшие множители, в результате чего ПФ распадается на произведение сомножителей первого и второго порядков. Редукция производится отбрасыванием сомножителей, которые близки к единице.

Выполняя описанное разложение ПФ (2.16) на простейшие сомножители, получаем ее представление в виде произведения семи сомножителей:

$$W(p) = \frac{p + 0.0035}{p + 0.0036} \cdot \frac{p + 0.7}{p + 0.582} \cdot \frac{p + 0.089}{p + 0.123} \cdot \frac{p + 4.396}{p + 2.462} \cdot \\ \cdot \frac{p - 0.0361}{p^2 + 0.918p + 0.0489} \cdot \frac{1}{p^2 + 1.922p + 0.48} \cdot \frac{-2.406}{p + 13.2322}. \quad (2.17)$$

Из него видно, что у системы имеется три пары близких нулей и полюсов. Осуществляя «сокращение» этих нулей и полюсов (фактически отбрасывая их), приходим к редуцированной модели. Ее передаточная функция получается из (2.17) отбрасыванием первых трех сомножителей.

Если полученную модель шестого порядка подключить параллельно проверяемому объекту, то разность их выходных сигналов при правильном функционировании объекта будет близка к нулю [ 1 ].

### 2.2.2. Проверка правильности функционирования в отдельных режимах

Уменьшение сложности модели связано с ограничением числа режимов работы, в которых производится диагностирование. Так, например, если изменение входных сигналов, приводящее к появлению переходных процессов в проверяемом объекте, происходит редко, и большую часть времени объект работает при постоянных входных сигналах, то можно ограничиться проверкой правильности функционирования только в установленвшемся режиме. В установленшемся режиме все производные равны нулю, и модель ОД является статической и описывается системой алгебраических уравнений.

Из (2.1) следует:

$$\begin{aligned} Ax + Bu &= 0, \\ y &= Cx. \end{aligned} \quad (2.18)$$

Для устойчивых объектов матрица  $A$  неособенная, что позволяет непосредственно выразить выходные сигналы через входные:

$$y = Du, \quad (2.19)$$

где  $D = -C A^{-1}B$ .

Контроль производится сравнением выходных сигналов модели с соответствующими выходными сигналами ОД.

Полученная модель имеет  $s$  выходов, т.к.  $y \in \mathbb{R}^s$ . Можно добиться дальнейшего уменьшения сложности модели, применяя прием сокращения числа выходов модели.

Минимальной сложностью, очевидно, будет обладать модель с одним выходом:

$$z = M^T y.$$

Чтобы найти, описывающее ее уравнение, умножим обе части равенства (2.19) слева на вектор  $M$ :

$$z = M^T D u = G u, \quad (2.20)$$

где  $G = M^T C A^{-1} B$ .

Тогда контроль правильности функционирования производится проверкой равенства:

$$\Delta = M^T y - z = 0, \quad (2.21)$$

которое должно выполняться непрерывно в процессе функционирования объекта, включая время протекания переходных процессов.

В случае, если модель ОД задана дискретной передаточной функцией, то в установившемся режиме в соответствии с [11] рекомендуется метод контроля правильности функционирования с использованием передаточных чисел  $\lambda$ :

$$\lambda = W(z)|_{z=I}. \quad (2.22)$$

Оценка передаточного числа в установившемся режиме определяется следующим образом:

$$\hat{\lambda}_k(N) = \frac{\sum_{l=k}^{k+N} y_l}{\sum_{l=k}^{k+N} u_l}, \quad (2.23)$$

где  $y_l$  – входной сигнал,

$u_l$  – выходной сигнал,

$N$  – период усреднения,

$$\frac{1}{N} \sum_{l=k}^{k+N} u_l \neq 0; \quad \frac{1}{N} \sum_{l=k}^{k+N} y_l \neq 0.$$

В [ 11 ] показано:

$$\lim_{N \rightarrow \infty} \hat{\lambda}_k(N) = \lambda.$$

Критерий правильности функционирования ОД выглядит следующим образом:

$$|\hat{\lambda}_k(N) - \lambda| \leq \delta. \quad (2.24)$$

Контроль с помощью модели, включенной параллельно ОД, имеет существенный недостаток, который состоит в следующем. Любой технический ОД подвержен внешним возмущениям, вызываемым, например, силами трения, изменением нагрузки, нестабильностью источников энергии и т. п. Эти возмущения априори неизвестны и не могут быть учтены в модели.

Кроме того, при составлении математических моделей допускаются различные упрощения и предположения. В результате при отсутствии дефектов выходные сигналы ОД и модели не совпадают, причем с течением времени их разность может достигнуть значительной величины.

Чтобы устраниТЬ или, по крайней мере, ослабить описанный эффект, нужно принять специальные меры для непрерывного "подтягивания" модели к ОД. Одним из эффективных способов решения этой задачи является использование фильтра Калмана [ 8 ].

Фильтром Калмана в теории управления называют устройство (или алгоритм), предназначенный для получения оптимальных (в смысле среднеквадратичного критерия) оценок переменных состояния ОД (линейной динамической системы) по результатам измерения его выходных сигналов, искаженных аддитивным шумом.

## 2.3. Контроль с использованием фильтра Калмана

### 2.3.1. Общие уравнения фильтра

Рассмотрим линейную вероятностную динамическую систему (ДС) с дискретным временем. Такая система в режиме нормального функционирования, т.е. при отсутствии дефектов, описывается линейными разностными уравнениями состояния и наблюдений [12]. Уравнение состояния характеризует динамику системы, уравнение наблюдений определяет механизм образования данных, доступных измерению. Эти уравнения в общем виде можно записать следующим образом.

Уравнение состояния:

$$x(k+1) = F(k+1, k) x(k) + B(k+1, k) u(k) + G(k+1, k) w(k). \quad (2.25)$$

Уравнение наблюдений:

$$y(k) = H(k) x(k) + D(k) u(k) + \Psi(k) v(k), \quad (2.26)$$

где  $x(k)$  –  $n$ -мерный вектор состояния системы;

$F(k+1, k)$  – ее переходная матрица размера  $n \times n$ ;

$u(k)$  –  $p$ -мерный вектор управлений (вектор входа);

$B(k+1, k)$  – переходная матрица управления размера  $n \times p$ ;

$w(k)$  – случайный  $r$ -мерный вектор гауссовских шумов возмущений (шумов системы) с нулевым средним и корреляционной матрицей

$$E \left[ w(k) w^T(j) \right] = Q(k) \delta(kj),$$

где  $E$  – оператор статического усреднения,

$T$  – знак транспонирования,

$\delta(kj)$  – символ Кронекера;

$G(k+1, k)$  – переходная матрица шумов возмущений размера  $n \times r$ ;

$y(k)$  –  $s$ -мерный вектор наблюдений;

$H(k)$  – матрица наблюдений системы размера  $s \times n$ ;

$v(k)$  – случайный ( $m$ -мерный) вектор гауссовских шумов измерений с нулевым средним и корреляционной матрицей:

$$E[v(k)v^T(j)] = R(k)\delta(kj);$$

$D(k)$  и  $\Psi(k)$  – матрицы системы размера  $s \times p$  и  $s \times m$  соответственно.

Для полного статистического описания поведения ДС необходимо задать плотность распределения вероятностей ее начального состояния  $f[x(0)]$ . Так как все случайные переменные, входящие в уравнение (2.25), гауссовые, то начальное состояние ДС полностью описывается вектором средних  $E[x(0)] = \bar{x}(0)$  и корреляционной матрицей:

$$P(0) = E\{[x(0) - \bar{x}(0)][x(0) - \bar{x}(0)]^T\}.$$

Если в ДС, описываемой уравнениями (2.25) и (2.26), отсутствуют дефекты (нарушения функционирования), то для получения оценки вектора состояния  $x(k)$  по последовательности наблюдений  $y(k)$  можно воспользоваться результатами теории линейной фильтрации в пространстве состояний [12], приводящей к фильтру Калмана.

Рассмотрим ДС, заданную уравнениями состояния:

$$X(k+1) = F(k+1, k)x(k) + G(k+1, k)W(k) \quad (2.27)$$

и наблюдений:

$$y(k) = H(k)x(k) + v(k), \quad (2.28)$$

полученными из уравнений (2.25) и (2.26) соответственно при  $u(k) = 0$  и  $\Psi(k) = I$  ( $I$  – единичная матрица). Начальное состояние  $x(0)$  представляет собой гауссовский вектор со средним значением  $\bar{x}(0)$  и корреляционной матрицей  $P(0)$ . Шумы возмущений  $w(k)$  и измерений  $v(k)$  предполагаются некоррелированными, т.е.  $E\{w(k)v^T(j)\} = 0$  для всех  $k, j$ .

Задача фильтрации заключается в том, чтобы по совокупности последовательных измерений  $\{y(1), y(2), \dots, y(k)\} = Y_1^k$  найти оптимальную оценку вектора состояний  $\hat{x}(k)$ , удовлетворяющую критерию минимума среднеквадратического значения ошибки.

Выход уравнений фильтра Калмана можно найти в [13]. Приведем его результаты.

Оптимальный алгоритм оценивания вектора состояния линейной дискретной ДС, заданной уравнениями (2.27) и (2.28), описывается системой рекуррентных уравнений следующего вида.

Уравнение оценки:

$$\begin{aligned}\hat{x}(k/k) &= F(k, k-1)\hat{x}(k-1/k-1) + \\ &+ K(k)[y(k) - H(k)F(k, k-1)\hat{x}(k-1/k-1)] = \\ &= \hat{x}(k/k-1) + K(k)\tilde{z}(k/k-1),\end{aligned}\quad (2.29)$$

где  $K(k)$  – матричный коэффициент усиления оптимального фильтра:

$$\begin{aligned}K(k) &= P(k/k)H^T(k)R^{-1}(k) = \\ &= P(k/k-1)H^T(k)[H(k)P(k/k-1)H^T(k) + R(k)]^{-1};\end{aligned}\quad (2.30)$$

$\hat{x}(k/k-1)$  – оценка экстраполяции (предсказания) на один шаг, т.е. оценка вектора состояния на  $k$ -ом шаге по  $(k-1)$  измерениям;

$\tilde{z}(k/(k-1))$  – так называемый "обновляющий" процесс, играющий важную роль в теории калмановской фильтрации:

$$\tilde{z}(k/(k-1)) = y(k) - H(k)\hat{x}(k/k-1). \quad (2.31)$$

Корреляционная матрица ошибок фильтрации:

$$\begin{aligned}P(k/k) &= P(k/k-1)H^T(k) \times \\ &\times [H(k)P(k/k-1)H^T(k) + R(k)]^{-1}H(k)P(k/k-1).\end{aligned}\quad (2.32)$$

Корреляционная матрица ошибок экстраполяции:

$$\begin{aligned}P(k/k-1) &= F(k, k-1)P(k-1/k-1)F^T(k, k-1) + \\ &+ G(k, k-1)Q(k-1)G^T(k, k-1).\end{aligned}\quad (2.33)$$

Начальные условия:

$$\hat{x}(0/0) = \overline{x(0)}, \quad P(0/0) = P(0). \quad (2.34)$$

Оптимальный фильтр, алгоритм работы которого описывается уравнениями (2.29)–(2.34), называется фильтром Калмана. Структурная схема этого фильтра, построенная в соответствии с приведенными уравнениями показана на рис. 2.4. Работа фильтра начинается с установки начальных значений  $\overline{x(0)}$  и  $P(0)$ , известных на основе априорной информации, и ввода величин элементов корреляционных матриц шумов возмущения  $Q(k)$  и шумов измерения  $R(k)$ .

Оценки вектора состояния вычисляются рекуррентно по мере поступления новых измерений. При этом, как видно из (2.29), для вычисления текущего значения оценки  $\hat{x}(k/k)$  нет необходимости запоминать все предыдущие измерения  $\{y(1), y(2), \dots, y(k-1)\}$ , т.к. вся информация о них содержится в оценке  $\hat{x}(k-1/k-1)$ , полученной на предыдущем шаге. Фильтр Калмана, как это следует из его структурной схемы, представляет собой динамическую систему с переменным матричным коэффициентом усиления, величина которого зависит от точности текущих значений и уровня шумов измерений.

Текущая оценка в соответствии с формулой (2.29) является суммой оценки экстраполяции  $\hat{x}(k/k-1)$  и корректирующей поправки  $K(k)\tilde{z}(k/k-1)$ .

Оценка экстраполяции формируется из оценки фильтрации на предыдущем шаге путем умножения ее на матрицу перехода системы, которая описывает изменение вектора состояния при отсутствии внешних воздействий на систему. Величина поправки определяется тем весом, который придается новым измерениям на текущем шаге оценивания. Этот вес зависит от уровня шумов измерений и текущей точности оценивания, уменьшаясь по мере уточнения оценок.

Сформулируем основные особенности фильтра Калмана:

- 1) фильтр представляет собой рекуррентный, удобный для реализации на ЭВМ, алгоритм вычисления оценки состояния ДС при полностью известной ее модели;
- 2) оценка, получаемая с помощью фильтра, является линейной относительно наблюдений;
- 3) корреляционная матрица ошибок фильтрации  $P(k/k)$  вследствие линейности фильтра не зависит от наблюдений  $y(k)$  и, следовательно, может быть вычислена заранее (поэтому заранее можно вычислить и матричный коэффициент усиления фильтра  $K(k)$ );
- 4) алгоритмы фильтрации легко распространяются на случай многоканальных систем;
- 5) так как параметры фильтра Калмана изменяются во времени, то такой фильтр минимизирует среднеквадратическое значение ошибки оценивания не только в установившемся режиме, но и в течение переходного процесса.

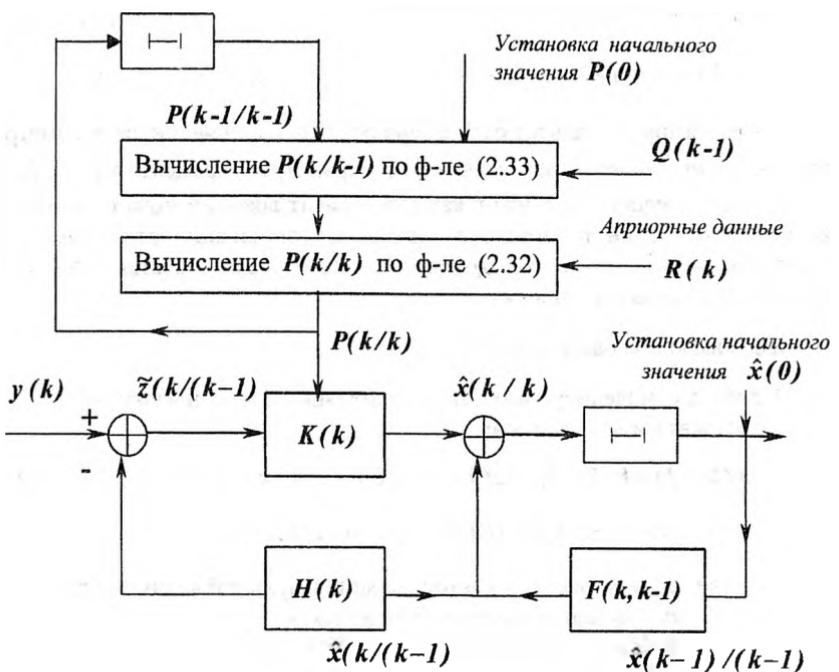


Рис. 2.4. Структурная схема фильтра Калмана

### 2.3.2. Модели дефектов

Под дефектом будем понимать возможные скачкообразные изменения параметров или структуры ДС, происходящие в случайные моменты времени. Поэтому для описания ДС, в которых необходимо учитывать возможность появления дефектов, введем вектор параметров  $\gamma(k)$ , характеризующий на данный момент структуру или параметры ДС. В дальнейшем рассматривается обширный класс моделей дефектов ДС, который характеризуется однократными (или достаточно редкими) скачкообразными изменениями вектора параметров  $\gamma(k)$ , происходящими в случайные моменты времени. Примером таких моделей могут служить такие явления, как внезапные дефекты в объекте или канале измерения, появление мешающих сигналов, приводящих к возникновению постоянных смещений в измерениях, и ряд других.

Уравнение состояния и наблюдений ДС оказываются в этом случае зависящими от изменяющегося вектора параметров  $\gamma(k)$ , и могут быть записаны следующим образом:

$$x(k+1) = F\{x(k), w(k), \gamma [k, v_1(m_1), v_2(m_2), \dots, v_N(m_N)]\}, \quad (2.35)$$

$$y(k) = H\{x(k), v(k), \gamma [k, v_1(m_1), v_2(m_2), \dots, v_N(m_N)]\}, \quad (2.36)$$

где вектор  $\gamma$  зависит от  $N$  скачкообразно изменяющихся параметров, причем  $v_i(m_i)$  – величина скачка  $i$ -го параметра в момент  $m_i$  ( $i = \overline{1, N}$ ). Кроме того, предположим, что в каждый момент времени может измениться не более чем один параметр, а частота возникновения этих изменений настолько низка, что в распоряжении наблюдателя имеется достаточно времени для принятия решения.

Выделим классы дефектов [12]:

- 1) дефекты, моделируемые в виде аддитивных сдвигов в уравнениях состояния и наблюдений:

$$x(k+1) = F(k+1, k)x(k) + w(k) + v \cdot I(k+1, m), \quad (2.37)$$

$$y(k) = H(k, k)x(k) + v(k) + v \cdot I(k, m), \quad (2.38)$$

где  $v$  – неизвестный вектор, характеризующий величину дефекта,

$m$  – момент возникновения дефекта,

$I(k, m)$  – единичная ступенчатая функция;

- 2) дефекты, моделируемые как изменение дисперсии шума в канале измерения и в канале наблюдения:

$$x(k+1) = F(k+1, k)x(k) + w(k) + v \cdot I(k, m), \quad (2.39)$$

$$y(k) = H(k)x(k) + v(k) + v \cdot I(k, m), \quad (2.40)$$

где  $v$  – неизвестный вектор, характеризующий величину среднеквадратичного отклонения (СКО) шума;

$m$  – момент изменения СКО шума;

- 3) дефекты, моделируемые как изменение параметров ДС.

Уравнения (2.37) и (2.38) описывают модели дефектов, связанных с появлением сдвигов в каналах измерения и наблюдения.

Уравнения (2.39) и (2.40) описывают модели дефектов, связанных с изменениями в параметрах шумовых компонент сигнала.

Модели 3-го класса описывают дефекты в правильном функционировании самой ДС.

Далее рассмотрим пример построения фильтров Калмана в классе авторегрессионных моделей объекта диагностирования.

### 2.3.3. Структура одномерного фильтра Калмана для моделей авторегрессии

#### Модель AP(1)

Устройство калмановской оценки реализует процесс параметрического оценивания, основанный на авторегрессионной модели генерации сигнала. AP - модель процесса первого порядка показана на рис. 2.5, а соответствующая модель измерений – на рис. 2.6.

*Белый шум*

$g(n-1)$

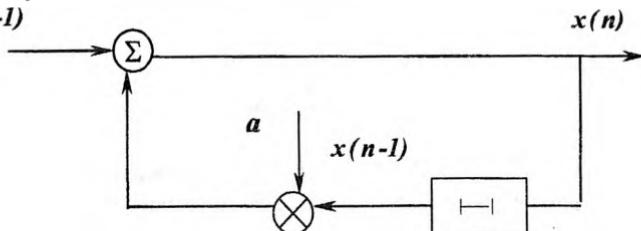


Рис. 2.5. Рекурсивная модель генерации сигнала первого порядка

Модель измерений представляет собой просто усилительное звено  $C$  и источник аддитивного белого шума  $v(n)$ .

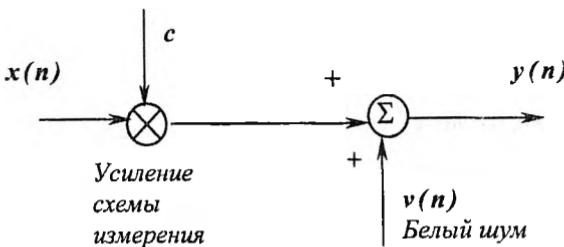


Рис. 2.6. Модель схемы измерения данных

Приняв эту модель генерации сигнала, поступающую выборку сигнала с номером  $n$  можно определить как

$$y(n) = c x(n) + v(n). \quad (2.41)$$

Рекурсивная формула оценки первого порядка [14] имеет вид:

$$\hat{x}(n) = b(n) \hat{x}(n-1) + k(n) y(n)$$

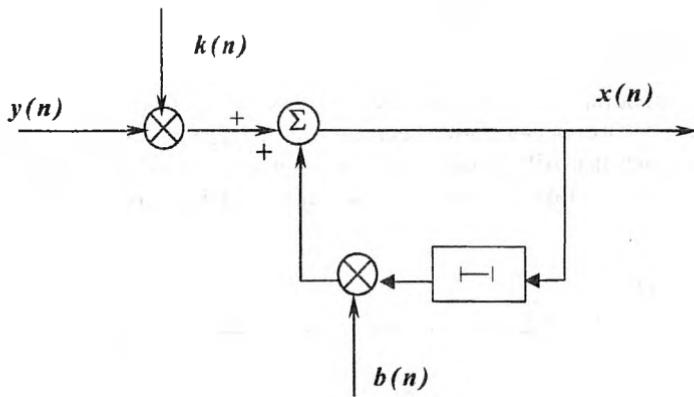


Рис. 2.7. Обобщенная структурная схема рекурсивного устройства оценки первого порядка

Для получения оптимального устройства оценки среднеквадратичная ошибка  $p(n)$  дифференцируется по  $b(n)$  и  $k(n)$ , а результаты приравниваются к нулю:

$$p(n) = E \{ \hat{x}(n) - x(n) \}^2,$$

$$\frac{\partial p(n)}{\partial b(n)} = 0, \quad \frac{\partial p(n)}{\partial k(n)} = 0.$$

Подробный вывод можно найти в [ 14 ], приведем результаты.

Окончательное соотношение между  $b(n)$  и  $k(n)$ :

$$b(n) = a / [1 - ck(n)],$$

и тогда уравнение оценки:

$$\hat{x}(n) = a\hat{x}(n-1) + k(n)[y(n) - ac\hat{x}(n-1)]. \quad (2.42)$$

Уравнение (2.42) является определением оптимального рекурсивного устройства оценки первого порядка, или скалярного фильтра Калмана. Первый член  $a\hat{x}(n-1)$  предсказывает текущую выборку, а второй член корректирует на основании оценки ошибки с учетом калмановского коэффициента  $k(n)$ .

Структура такого фильтра приводится на рис. 2.8.

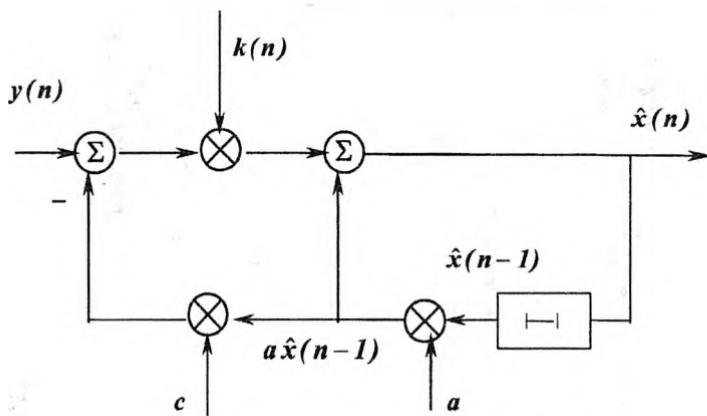


Рис. 2.8. Схема скалярного фильтра Калмана первого порядка

Выражения для определения  $p(n)$  и  $k(n)$ :

$$k(n) = \frac{c[a^2 p(n-1) + \sigma_g^2]}{\sigma_v^2 + c^2 \sigma_g^2 + c^2 a^2 p(n-1)}, \quad (2.43)$$

$$p(n) = \frac{1}{c} \sigma_v^2 k(n) \quad (2.44)$$

Сначала, зная  $p(n-1)$ , надо рассчитать  $k(n)$ , а затем уже  $p(n)$ .

Три уравнения: (2.42), (2.43) и (2.44) – являются рекурсивными уравнениями, необходимыми для реализации фильтра Калмана первого порядка.

Общая схема моделирования фильтра в соответствии с (2.42), (2.43) и (2.44) приведена на рис.2.9.

### Модель AP(2)

Рассмотрим структуру фильтра Калмана 2-го порядка, моделирующего процесс авторегрессии 2-го порядка.

Уравнение AP(2):

$$x(n) = \psi_1 x(n-1) + \psi_2 x(n-2) + g(n-1), \quad (2.45)$$

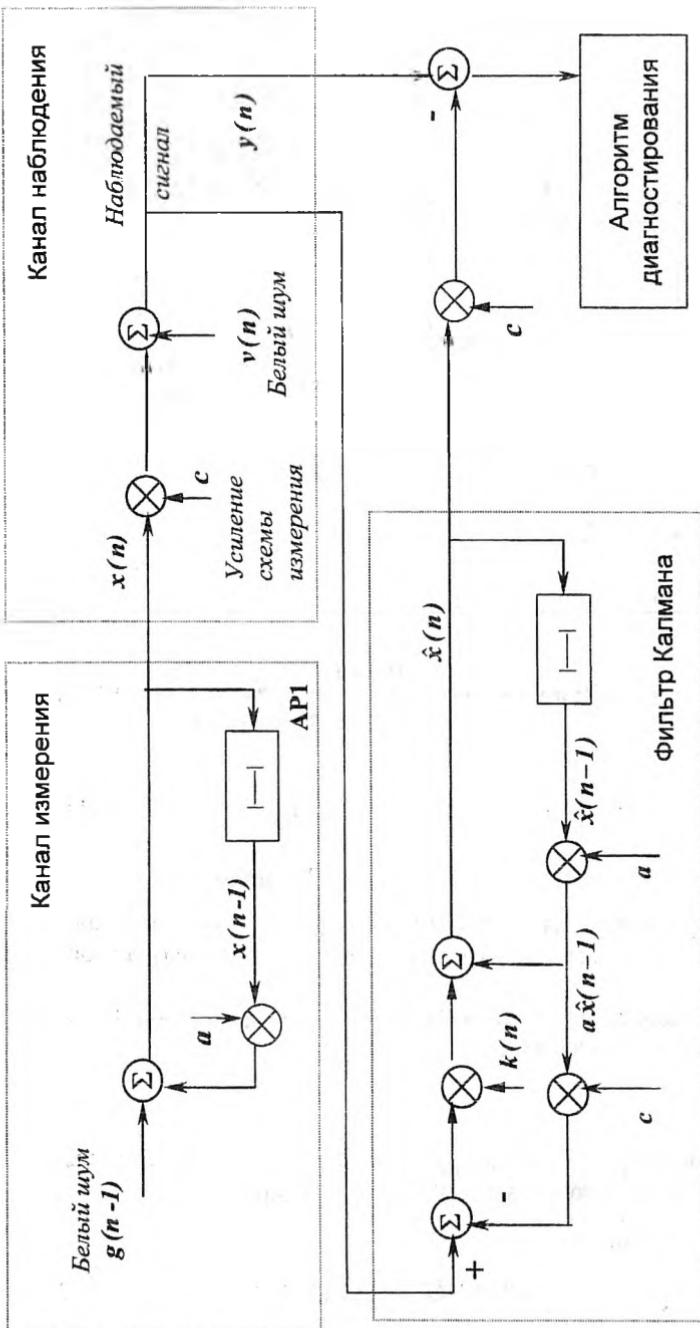


Рис. 2.9. Общая схема диагностирования.

Зададим переменные состояния  $x_1(n)$  и  $x_2(n)$ :

$$x_1(n) = x(n), \quad x_2(n) = x(n-1)$$

и перепишем уравнение (2.45) в виде пары уравнений состояния:

$$\begin{cases} x_1(n) = \psi_1 x_1(n-1) + \psi_2 x_2(n-1) + g(n-1) \\ x_2(n) = x_1(n-1) \end{cases}. \quad (2.46)$$

Записывая (2.46) в виде матричного уравнения, имеем:

$$\begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = \begin{bmatrix} \psi_1 & \psi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \end{bmatrix} + \begin{bmatrix} g(n-1) \\ 0 \end{bmatrix}$$

или

$$X(n) = A X(n-1) + G(n-1).$$

Уравнения фильтра Калмана для задачи оценивания представлены теперь в векторной форме, но они имеют такой же вид, как и для скалярного фильтра.

$$\begin{aligned} \hat{X}(n-1) &= A \hat{X}(n-1) + K(n) [Y(n) - C A \hat{X}(n-1)], \\ K(n) &= P^T C^T [C P_I(n) C^T + Z(n)]^{-1}, \\ P_I(n) &= A P(n-1) A^T + Q(n-1), \\ P(n) &= P_I(n) - K(n) C P_I(n), \end{aligned} \quad (2.47)$$

где дисперсия скалярного наблюдаемого шума  $\sigma_v^2$  и дисперсия шума системы  $\sigma_g^2$  заменены корреляционными матрицами  $Z(n)$  и  $Q(n)$  соответственно.

### 2.3.4. Использование обновляющего процесса для обнаружения дефектов

Для обнаружения вышеописанных дефектов можно использовать обновляющие процессы, обладающие тем свойством, что если ДС функционирует нормально, то обновляющий процесс

$$\hat{z}(k/k-1) = y(k) - H(k) \hat{x}(k/k-1)$$

в согласованном с моделью динамики калмановском фильтре представляет собой белый шум с нулевым средним и корреляционной матрицей:

$$P_z(k) = H(k) P(k/k-1) H^T(k) + R(k). \quad (2.48)$$

Если в системе происходят дефекты, то статистические характеристики процесса  $\hat{z}(k/k-1)$  изменяются, и эти изменения можно использовать для обнаружения дефектов.

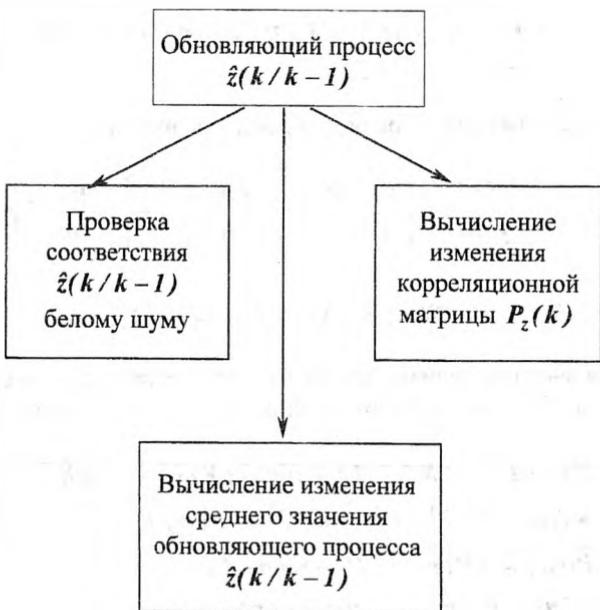


Рис 2.10. Классификация методов обнаружения дефектов, основанных на использовании обновляющих процессов

Учитывая вышеизложенное, можно провести классификацию методов обнаружения дефектов, использующих обновляющие процессы (рис.2.10).

Различные способы обнаружения в ДС по обновляющей последовательности фильтра Калмана рассматриваются в [13]. В данном учебном пособии рассматриваются только методы обнаружения, основанные на выявлении изменения среднего значения обновляющего процесса.

Эти же методы могут быть использованы и для локализации дефектов [13]. В этом случае выделяется множество возможных состояний ОД (гипотез), характеризующих дефекты различных элементов ОД (датчиков, приводов, органов управления).

Для каждой гипотезы синтезируется фильтр Калмана. На основании невязок фильтра, характеризуемых векторным обновляющим процессом и в соответствии с принятым методом обнаружения выясняется, какая из гипотез имела место. Недостаток такого подхода состоит, во-первых, в сложности его

реализации при работе ОД в реальном масштабе времени, во-вторых, в невысокой достоверности локализации дефекта, так как принятый метод обнаружения не всегда может различить, какой из возможных дефектов имел место на самом деле.

## 2.4. Контроль с помощью обнаруживающих фильтров

Методы этой группы [8] основаны на том, что для объекта диагностирования синтезируется наблюдатель, невязка которого чувствительна к дефекту, т.е. при разных дефектах вектор невязки принимает разные значения.

В качестве примера обнаруживающего фильтра рассмотрим фильтр Бирда [8]. Пусть ОД описывается уравнениями в пространстве состояний:

$$\dot{x} = Ax + Bu; \quad y = Cx. \quad (2.49)$$

В качестве устройства диагностирования возьмем модель объекта, имеющую структуру фильтра Калмана:

$$\dot{\hat{x}} = A\hat{x} + k(y - C\hat{x}) + Bu, \quad \hat{y} = C\hat{x}. \quad (2.50)$$

При отсутствии дефектов выходные и внутренние сигналы ОД (2.49) и фильтра Калмана (2.50) должны совпадать т.е.,  $y = \hat{y}$ ,  $\hat{y} = c\hat{x}$ . Наличие погрешностей и дефектов в функционировании ОД приведет к появлению невязок:

$$\Delta = y - \hat{y}; \quad \varepsilon = x - \hat{x}. \quad (2.51)$$

Для удобства дальнейшего анализа этой информации найдем дифференциальное уравнение для невязки. Вычитая уравнение (2.50) из (2.49) и учитывая (2.51), получим:

$$\dot{\varepsilon} = (A - kC)\varepsilon. \quad (2.52)$$

Пусть имеет место полная доступность вектора состояния для непосредственных измерений, т.е.  $\varepsilon = \Delta$ ,  $C = E$ , и уравнение (2.52) можно представить в виде:

$$\dot{\Delta} = (A - k)\Delta. \quad (2.53)$$

Невязка  $\Delta$ , в отличие от невязки  $\varepsilon$ , является реальным физическим сигналом, полученным как разность выходов объекта и фильтра.

Для исключения перекрестного влияния составляющих невязки друг от друга матрица  $K$  выбирается следующим образом:

$$K = \alpha E + A, \quad (2.54)$$

где  $E$  – единичная матрица.

Подставляя (2.54) в уравнение (2.53), получим:

$$\dot{\Delta} = -\alpha \Delta. \quad (2.55)$$

или в скалярной записи:

$$\dot{\Delta}_I = -\alpha \Delta_I, \dots, \dot{\Delta}_n = -\alpha \Delta_n. \quad (2.56)$$

Решение этих дифференциальных уравнений имеет вид загувающих экспонент:

$$\Delta_I = c_I e^{-\alpha t}, \dots, \Delta_n = c_n e^{-\alpha t},$$

т.е. при отсутствии дефектов все невязки независимо стремятся к нулю, причем скорость стремления к нулю можно регулировать выбором коэффициента  $\alpha$ .

После упрощающих предположений  $C = E$  и  $K = \alpha E + A$  получим следующие уравнения, описывающие объект и фильтр:

$$\dot{x} = Ax + Bu; \quad y = x. \quad (2.57)$$

$$\dot{\hat{x}} = -\alpha \hat{x} + (\alpha E + A)y + Bu, \quad \hat{y} = \hat{x}. \quad (2.58)$$

Фильтр, задаваемый уравнением (2.58), называется обнаруживающим фильтром Бирда. Фильтр, построенный таким образом, позволяет сравнительно просто обнаруживать дефекты и осуществлять их локализацию.

Пусть в рамках принятой модели ОД (2.57) проявление дефекта эквивалентно искажению  $i$ -го столбца в матрице  $B$ . Таким образом, уравнения (2.57) и (2.58) принимают вид:

$$\begin{aligned}\dot{x} &= Ax + Bu + vb_i, \\ \dot{\hat{x}} &= Ax + Bu + \alpha(x - \hat{x}),\end{aligned}$$

где  $v$  – уровень вносимой ошибки

Дифференциальное уравнение для невязки получим, исходя из их разности:

$$\dot{\Delta} = -\alpha \Delta + vb_i. \quad (2.59)$$

Если до момента возникновения дефекта  $t_I$ , вектор невязок, определяемый уравнением (2.55) был равен нулю, то после момента  $t_I$  в соответствии с уравнением (2.59) получим:

$$\dot{\Delta}_I + \alpha \Delta_I = vb_{II}, \dots, \dot{\Delta}_n + \alpha \Delta_n = vb_{In}, \quad t \geq t_I.$$

Решение этих уравнений при нулевых начальных условиях имеет вид:

$$\Delta_I = \frac{v}{\alpha} b_{I1} (1 - e^{-\alpha(t-t_1)}) ;$$

.....;

$$\Delta_n = \frac{v}{\alpha} b_{In} (1 - e^{-\alpha(t-t_1)}), \quad t \geq t_1 ;$$

Следовательно, после окончания переходного процесса, вызванного появлением дефекта, через интервал времени  $3\tau$ , где  $\tau = \frac{1}{\alpha}$ , для вектора невязок можно записать:

$$\Delta = \frac{v}{\alpha} b_I, \quad t \geq t_1 + \frac{3}{\alpha}. \quad (2.60)$$

Из формулы (2.60) следует, что при данном типе дефекта вектор невязок оказывается пропорциональным  $i$ -му столбцу матрицы  $B$ . Таким образом, для локализации таких дефектов достаточно выяснить, какому из столбцов матрицы  $B$  пропорционален вектор невязок; номер этого столбца будет связан с характерной физической неисправностью в рассматриваемом ОД, а коэффициент пропорциональности будет характеризовать величину искажения. Примерно по такой же схеме анализируется влияние дефектов, возникающих в датчиках (вектор  $y$ ) на вектор невязки  $\Delta$ .

В заключении перечислим основные ограничения и упрощения, характерные для фильтра Бирда:

- датчики и исполнительные механизмы считаются безынерционными;
- дефекты предполагаются константными;
- обработка данных в УД считается надежной;
- число датчиков равно числу компонент обобщенного вектора состояний;
- размерность УД равна размерности ОД.

Наиболее существенным из этих ограничений является предпоследнее.

В [8] предложены способы для снижения размерности УД и построения обнаруживающего фильтра для ОД с неполной информацией о состояниях.

В данном разделе сделана попытка отразить основные направления и подходы при контроле правильности функционирования ОД, используемые в современных и перспективных разработках систем управления.

Весьма интересным, но не охваченным содержанием данного пособия является разработка экспертных систем. Работы в этом направлении активно ведутся в последнее время [15].

Далее рассмотрим основные методы и подходы к построению алгоритмов принятия решений в выбранном пространстве диагностических признаков.

### 3. Методы принятия решений в системах функционального диагностирования

#### 3.1. Проверка статистических гипотез

Пусть наблюдаемый объект находится в одном из возможных состояний:

$$S_0, S_1, \dots, S_k, \dots, S_m,$$

которые представляют собой полную группу  $m$  событий.

Пусть  $P_0, P_1, \dots, P_k, \dots, P_m$  ( $\sum_{k=1}^m P_k = 1$ ) – априорное

распределение вероятностей этих событий. Пусть  $x_1, x_2, \dots, x_n$  – результаты наблюдений (выборка), которые зависят от того, в каком из состояний в действительности находится наблюдаемый объект.

Пусть  $W_n(x_1, x_2, \dots, x_n / S_k)$  – условное распределение выборочных значений, соответствующее состоянию  $S_k$ ,  $k = 0, 1, \dots, m$ . Имеется набор решений  $\gamma_0, \gamma_1, \dots, \gamma_i, \dots, \gamma_m$  относительно истинного состояния, правило  $\delta(\gamma_0, \gamma_1, \dots, \gamma_i, \dots, \gamma_m)$  предписывает каждому возможному результату наблюдений  $x_1, x_2, \dots, x_n$  одно из решений  $\gamma_i$  ( $i = 0, 1, \dots, m$ ). Задана функция потерь  $P(S_k, \gamma_i)$ , учитывающая последствия выбора решения.

Необходимо выбрать наилучший алгоритм выбора решений по наблюдаемой выборке  $x_1, x_2, \dots, x_n$ .

Рассмотрим далее поставленную проблему в двухальтернативном варианте, т.е. имеем два состояния  $S_0$  и  $S_1$ , причем  $S_0, S_1$  – состояния, соответствующие правильному и неправильному функционированию ОД. Определяющей статистикой в соответствии с [4] является отношение правдоподобия:

$$l(x_1, x_2, \dots, x_n) = \frac{W(x_1, x_2, \dots, x_n / S_1)}{W(x_1, x_2, \dots, x_n / S_0)} \quad (3.1)$$

или логарифм отношения правдоподобия:  $\ln l(x_1, x_2, \dots, x_n)$ .

Согласно теории проверки статистических гипотез оптимальное правило формирования решений выглядит следующим образом:

$$l(x_1, x_2, \dots, x_n) \geq C, \quad (3.2)$$

где выбор порога зависит от наличия необходимой априорной информации:

1. Байесов подход (полная информация):

$$C = \frac{P_\theta(\Pi_{\theta I} - \Pi_{\theta \theta})}{P_I(\Pi_{I \theta} - \Pi_{II})}. \quad (3.3)$$

При этом подходе минимизируется средний риск или средние потери.

2. Максимум апостериорной информации (отсутствует информация о потерях):

$$C = \frac{P_\theta}{P_I}. \quad (3.4)$$

В этом случае максимизируется частота правильных решений.

3. Максимум правдоподобия (отсутствует информация о потерях и априорном распределении вероятностей состояний):

$$C = 1. \quad (3.5)$$

Имеем частный случай второго подхода при  $P_\theta = P_I = 1/2$ .

Количественная оценка качества алгоритмов формирования решений состоит в вычислении ошибок первого и второго рода.

Ошибка 1<sup>го</sup> рода ( $\alpha$ ) состоит в принятии решения  $\gamma_I$  при истинном состоянии  $S_\theta$ . Ошибка 2<sup>го</sup> рода  $\beta$  состоит в принятии решения  $\gamma_\theta$  при истинном состоянии  $S_I$ .

При отсутствии априорной информации, как это имеет место в третьем случае, часто используется критерий Неймана-Пирсона, согласно которому порог  $C$  выбирается из условия, что вероятность ошибки первого рода не больше заданной величины  $\alpha_3$ .

Особенностью всех рассмотренных условий выбора правила принятия решения была неизменность заранее заданного объема выборки. Существует другой подход к установлению правила выбора решения, при котором отказываются от постоянного размера объема выборки, но ограничивают эту величину в процессе эксперимента в зависимости от результата уже выполненных наблюдений.

Такой подход, как показал Вальд [4] среди правил выбора решений (последовательных и непоследовательных), для которых условные вероятности ошибок  $\alpha, \beta$  не превосходят заданных величин  $\alpha_3, \beta_3$  приводит к наименьшим средним объемам выборки для принятия окончательного решения.

Оптимальное последовательное правило выбора решений, вытекающее из данного подхода может быть сформулировано следующим образом:

– при  $n$  наблюдениях принимается решение  $\gamma_0$ , если

$$\frac{\beta_3}{1-\alpha_3} < l(x_1, x_2, \dots, x_i) < \frac{1-\beta_3}{\alpha_3}, \quad i = 1, 2, \dots, n-1. \quad (3.6)$$

$$l(x_1, x_2, \dots, x_n) \leq \frac{\beta_3}{1-\alpha_3}. \quad (3.7)$$

– принимается решение  $\gamma_1$ , если выполняется неравенство (3.6) и

$$l(x_1, x_2, \dots, x_n) \geq \frac{1-\beta_3}{\alpha_3} > 1. \quad (3.8)$$

Аналогично (3.6), (3.7), (3.8) можно сформулировать правила принятия решения для  $ln[l(x_1, x_2, \dots, x_n)]$ .

Величины средних размеров выборки  $m(n/S_\theta)$  и  $m(n/S_0)$ , соответствующие данному алгоритму формирования решений определяются следующим образом:

$$m[n/S_\theta] = \frac{(1-\alpha) \ln \frac{\beta}{1-\alpha} + \alpha \ln \frac{1-\beta}{\alpha}}{m_0}, \quad (3.9)$$

$$m[n/S_1] = \frac{\beta \ln \frac{\beta}{1-\alpha} + (1-\beta) \ln \frac{1-\beta}{\alpha}}{m_1}, \quad (3.10)$$

где  $m_0 = m_0 [ln l(x)/S_\theta] = \int_{-\infty}^{+\infty} W(x/S_\theta) \ln \frac{W(x/S_1)}{W(x/S_\theta)} dx, \quad (3.11)$

$$m_1 = m_1 [ln l(x)/S_1] = \int_{-\infty}^{+\infty} W(x/S_1) \ln \frac{W(x/S_1)}{W(x/S_\theta)} dx, \quad (3.12)$$

Индекс «з» в формулах (3.9), (3.10) для упрощения обозначений опущен. Для иллюстрации рассмотренных выше правил рассмотрим примеры.

### Пример 3.1

Определяющим критерием будем считать критерий максимума правдоподобия, причем в качестве решающей статистики выберем:

$$\ln \{ l(x_1, x_2, \dots, x_n) \}.$$

$$1. \text{ Пусть } W(x_1, x_2, \dots, x_n / S_k) = \prod_{i=1}^n W_i(x_i / S_k),$$

т.е. имеем независимую выборку, где

$$W(x_i / S_k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i - a_k)^2}{2\sigma^2}},$$

т.е. состояние  $S_k$  однозначно связано со значением математического ожидания  $a_k$ ,  $k = (0, 1)$ .

Тогда:

$$\begin{aligned} \ln \{ l(x_1, x_2, \dots, x_n) \} &= \sum_{i=1}^n \ln l(x_i) = \sum_{i=1}^n \ln \frac{\frac{1}{\sigma \sqrt{2\pi}} \exp \left[ -\frac{(x_i - a_1)^2}{2\sigma^2} \right]}{\frac{1}{\sigma \sqrt{2\pi}} \exp \left[ -\frac{(x_i - a_0)^2}{2\sigma^2} \right]} = \\ &= \frac{a_1 - a_0}{\sigma^2} \sum_{i=1}^n x_i - \frac{n(a_1^2 - a_0^2)}{2\sigma^2}. \end{aligned}$$

С учетом (3.5) правило принятия решений выглядит следующим образом:

принимаем решение о состоянии  $S_1$ , если

$$\frac{1}{n} \sum_{i=1}^n x_i \geq \frac{a_0 + a_1}{2}; \quad a_1 \geq a_0;$$

и о состоянии  $S_0$ , если:

$$\frac{1}{n} \sum_{i=1}^n x_i \leq \frac{a_0 + a_1}{2};$$

Ошибки первого и второго родов (при  $a_1 > a_0$ ) в соответствии с [4] вычисляется следующим образом:

$$\begin{aligned} \alpha &= 1 - F\left(\frac{h - a_0}{\sigma} \sqrt{n}\right), \\ \beta &= F\left(\frac{h - a_0}{\sigma} \sqrt{n}\right), \end{aligned} \tag{3.13}$$

где

$$h = \frac{a_0 + a_1}{2}, \quad F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t}{2}z^2} dz. \quad (3.14)$$

Из (3.13) и (3.14) следует:

$$\alpha = \beta = 1 - F \left( \frac{a_1 - a_0}{2\sigma} \sqrt{n} \right).$$

### Пример 3.2

Пусть  $x_1, x_2, \dots, x_n$  независимая выборка дискретных значений:

Состояние  $S_0$ ,  $x_i = \begin{cases} 0 & - 1 - p_0 = q_0; \\ 1 & - p_0; \end{cases}$

Состояние  $S_1$ ,  $x_i = \begin{cases} 0 & - 1 - p_1 = q_1; \\ 1 & - p_1. \end{cases}$

Тогда имеем:

$$W(x_1, x_2, \dots, x_n / S_k) = \prod_{i=1}^n q_k^{(1-x_i)} p_k^{x_i}, \quad (3.15)$$

где  $\kappa = (0, 1)$ .

Решающее правило выглядит следующим образом:

$$k \ln \frac{p_1}{p_0} + (n-k) \ln \frac{1-p_1}{1-p_0} >< \theta, \quad (3.16)$$

где  $\kappa$  – число единиц в течение  $n$  наблюдений.

Если  $\ln \{l(x_1, x_2, \dots, x_n)\} > 0$ , то имеем  $S_1$ , в противном случае  $-S_0$ .

### Пример 3.3

Имеем многомерное распределение:

$$W(x_1, x_2, \dots, x_n / S_k) = \frac{1}{(\sqrt{2\pi})^n \sqrt{\det M_k}} * \exp \left[ -\frac{1}{2} (x - a_k)^T M_k^{-1} (x - a_k) \right], \quad (3.17)$$

где  $a_k$  – вектор математических ожиданий,

$M_k$  – корреляционная матрица ( $\kappa = 0, 1$ ).

Пусть  $M_0 = M_I = M$ , тогда решающее правило выглядит следующим образом:

$$x^T M^{-1} (a_I - a_0) >< \frac{1}{2} (a_I - a_0)^T M^{-1} (a_I - a_0). \quad (3.18)$$

Если принять, что выборка независимая и значения нормированы, т.е.

$\frac{x_i}{\sigma_i} = \tilde{x}_i$ , то в соответствии с (3.18) имеем:

$$\|\tilde{x} - \tilde{a}_I\|^2 >< \|\tilde{x} - \tilde{a}_0\|^2, \quad (3.19)$$

где  $\|y\|$  – евклидова норма вектора  $y$ .

Правило (3.19) соответствует правилу минимального расстояния в соответствии с выбранной нормой.

Ошибки первого и второго родов в соответствии с [4] определяются следующим образом:

$$\begin{aligned} \alpha &= F(d/2), \\ \beta &= F(-d/2), \end{aligned} \quad (3.20)$$

где  $d^2 = (a_I - a_0)^T M^{-1} (a_I - a_0)$ ,

$F(x)$  – соответствует соотношению (3.14).

#### Пример 3.4

Рассмотрим реализацию примера 3.1 в соответствии с последовательным методом Вальда.

Решение о состоянии  $S_0$  (среднее значение  $a_0$ ) при  $a_I > a_0$  на  $n$ -ом шаге принимается, если

$$\frac{1}{n} \sum_{i=1}^n x_i \leq \frac{a_0 + a_I}{2} + \frac{\sigma^2 \ln \frac{\beta}{1-\alpha}}{n(a_I - a_0)}. \quad (3.21)$$

Решение о состоянии  $S_I$  (среднее значение  $a_I$ ) принимается, если

$$\frac{1}{n} \sum_{i=1}^n x_i \geq \frac{a_0 + a_I}{2} + \frac{\sigma^2 \ln \frac{1-\beta}{\alpha}}{n(a_I - a_0)}, \quad (3.22)$$

и продолжаются наблюдения, если не выполняются ни одно из указанных неравенств (3.21) и (3.22).

### 3.2. Допусковые алгоритмы

Пусть рассматривается стационарный режим работы ОД. Определяющим диагностическим признаком правильности функционирования ОД является параметр  $x$ , который наблюдается и содержит измерительную погрешность  $\xi$ :

$$y = x + \xi = x + m + \delta, \quad (3.23)$$

где  $m$  – систематическая погрешность,

$\delta$  – случайная составляющая ошибки наблюдения.

Заданы закон распределения параметра  $x - F(x)$  и закон распределения  $\varphi(\delta)$ . Пусть  $F(x)$  и  $\varphi(\delta)$  – нормальные законы распределения с параметрами  $(\mu_x, \sigma_x)$  и  $(0, \sigma_n)$  соответственно. Задан контрольный допуск  $[a, b]$ , т.е. если  $y \in [a, b]$ , то объект правильно функционирует и неправильно, если  $y \notin [a, b]$ .

В соответствии с теорией проверки гипотез возможны два типа ошибок:

– ошибка первого рода:

$$x \in [a, b], \quad y \notin [a, b];$$

– ошибка второго рода:

$$x \notin [a, b], \quad y \in [a, b].$$

Ошибки первого рода  $\alpha$  вычисляются в соответствии с формулой условных вероятностей:

$$\alpha = \frac{P(y \notin [a, b], x \in [a, b])}{P(x \in [a, b])}, \quad (3.24)$$

где  $P(y \notin [a, b], x \in [a, b]) =$

$$= \int_a^b f(x) \left[ \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{a-x-m}{\sigma_n}} e^{-\frac{1}{2}z^2} dz + \frac{1}{\sqrt{2\pi}} \int_{\frac{b-x-m}{\sigma_n}}^{\infty} e^{-\frac{1}{2}z^2} dz \right] dx. \quad (3.25)$$

$$P(x \in [a, b]) = \int_a^b f(x) dx = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{1}{2}\frac{(x-\mu_x)^2}{\sigma_x^2}} dx. \quad (3.26)$$

Интегралы (3.25), (3.26) могут быть вычислены соответственно с помощью таблиц двумерного и одномерного нормальных распределений.

Аналогично определяется ошибка второго рода:

$$\beta = \frac{P(y \in [a, b], x \notin [a, b])}{P(x \notin [a, b])}, \quad (3.27)$$

В случае, если диагностический признак является вектором и содержит  $n$  независимых составляющих, то в [16] получены обобщенные выражения для ошибок 1<sup>го</sup> и 2<sup>го</sup> родов при условиях:

- ОД считается правильно функционирующим, если все составляющие ДП находятся в допуске;
- ОД считается неправильно функционирующим, если хотя бы одна из составляющих ДП находится не в допуске.

Ошибки первого и второго родов определяются следующими выражениями:

$$\alpha = 1 - \prod_{i=1}^n (1 - \alpha_i), \quad (3.28)$$

$$\beta = \frac{\prod_{i=1}^n [P_i(1 - \alpha_i) + (1 - P_i)\beta_i] - \prod_{i=1}^n P_i(1 - \alpha_i)}{1 - \prod_{i=1}^n P_i}, \quad (3.27)$$

где  $P_i = P(x \in [a_i, b_i]), \quad i = \overline{1, n}$ .

Обычно трудности в количественном определении ошибок  $\alpha$  и  $\beta$  для алгоритмов допускового контроля связаны с очень частым отсутствием необходимой априорной информации о законах распределения  $F(x)$ ,  $\varphi(\delta)$ , а также с неправомерностью гипотезы о независимости составляющих ДП. Это существенно усложняет процесс аргументированного назначения допусков при формировании решений о состоянии ОД.

### 3.3. Обнаружение разладки случайных процессов

#### 3.3.1. Алгоритмы обнаружения

Общий круг практических задач, в которых естественно возникает потребность обнаруживать изменение (разладку) свойств случайных процессов достаточно широк [6]:

- Задачи контроля и диагностики в автоматизированных системах управления технологическими процессами (АСУ ТП); контроль исходного сырья, полуфабрикатов и готовой продукции; контроль исправности функционирования технологических агрегатов и установок; контроль исправности измерительных и управляющих систем.

- Задачи автоматизации обработки научных наблюдений, обнаружение изменения характеристик наблюдаемого явления по временному ряду измеряемых параметров (задача обнаружения изменений электрокардиограммы, энцефалограммы, задача анализа сейсмозаписей), автоматический отбор участков записей, содержащую полезную информацию по каналам с ограниченной пропускной способностью.

Рассмотрим формальную постановку задачи.

Пусть дана случайная последовательность  $[x_1, x_2, \dots, x_N]$ , далее  $\{x_i\}$ , которая в момент  $t_0$  скачком изменяет свои свойства, однозначно определяемые вектором параметров  $\theta$ . Это значит, что до момента  $t_0$   $\theta = \theta_1$ , а начиная с момента  $t_0$  вектор  $\theta = \theta_2$ .

В дальнейшем рассматривается только последовательное обнаружение, т. е. решение о появлении разладки формируется в темпе поступления информации.

Далее предполагается что  $t_0$  – целочисленная случайная величина. В этих условиях обычно рассматриваются следующие основные варианты критерии для синтеза и сравнения алгоритмов:

1. Необходимо отыскать правило подачи сигнала о разладке, минимизирующее среднее время  $\bar{\tau}$  запаздывания в обнаружении разладки:

$$\bar{\tau}_{обн} = M(\tau / t_a \geq t_0), \quad (3.30)$$

где  $\tau = t_a - t_0 + 1$ ,  $t_a$  – момент подачи сообщения о разладке при заданном математическом ожидании  $M(N)$  числа ложных тревог, подаваемых до момента появления разладки  $t_0$ .

2. Необходимо минимизировать  $\bar{\tau}_{обн}$  в соответствии с (3.30) при заданной вероятности ложной тревоги  $P_{ло}$ :

$$P_{ло} = P(t_a < t_0). \quad (3.31)$$

Кратко охарактеризуем некоторые основные алгоритмы обнаружения при следующих допущениях:

1. Последовательность  $x_1, x_2, \dots, x_N$  – скалярная и может быть представлена следующими моделями:

– независимая последовательность

$$x_i = \mu_i + \sigma_i z_i, \quad (3.32)$$

где  $\theta_i = (\mu_i, \sigma_i)$ ,  $x_i \in N(\mu_i, \sigma_i)$ ,  $z_i \in N(0, 1)$ ;

— зависимая последовательность с корреляционной функцией  $k_x(\tau) = \sigma^2 e^{-\alpha|\tau|}$ ,  
тогда имеем:

$$x_i = \mu_1 + a_1(x_{i-1} - \mu_1) + \beta_1 z_i, \quad (3.33)$$

$$\text{где } \theta_1 = (\mu_1, a_1, \beta_1), \quad \beta_1 = \sigma_1 \sqrt{1 - a_1^2},$$

$$x_i \in N(\mu_1, \sigma_1), \quad z_i \in N(0, 1).$$

2. Разладка проявляется скачком в изменении только параметра  $\mu$ , значение параметра  $\mu_2$  в момент возникновения разладки  $t_0$  неизвестно.

Второе предположение сужает поле рассматриваемых разладок, т.е. речь идет только об обнаружении сдвигов (изменений среднего) случайной последовательности. Приведем только некоторые алгоритмы для решения указанной задачи из весьма обстоятельный обзора, приведенного в [17].

### *Алгоритмы Пейджса [6]*

1. Односторонний, т.е. ( $\mu_2 > \mu_1$ ):

$$g(n) = S_1^n - \min_{1 \leq k \leq n} S_1^k \gg h, \quad n = 1, 2, \dots, N, \quad (3.34)$$

$$\text{где } S_1^k = \sum_{i=1}^k (x_i - \mu_1 - v_\theta/2),$$

$$|v| = |\mu_2 - \mu_1| \geq v_\theta > 0.$$

2. Двусторонний, т.е. ( $\mu_2 > \mu_1$  или  $\mu_2 < \mu_1$ ):

$$g_1(n) = S_1^n - \min_{1 \leq k \leq n} S_1^k \gg h, \quad \text{при } \mu_2 > \mu_1,$$

$$\text{где } S_1^k = \sum_{i=1}^k (x_i - \mu_1 - v_\theta/2), \quad (3.35)$$

$$g_2(n) = \max_{1 \leq k \leq n} S_1^k - S_1^n \gg h, \quad \text{при } \mu_2 < \mu_1,$$

$$\text{где } S_1^k = \sum_{i=1}^k (x_i - \mu_1 + v_\theta/2).$$

Значение  $v_\theta$  определяет порог чувствительности в обнаружении сдвига случайного процесса.

### *Алгоритм кумулятивных сумм (AKC) с отражающим экраном [6]*

Этот алгоритм представляет модифицированную форму записи алгоритмов Пейджа [6]. В случае (3.34) имеем:

$$g(n) = \{ g(n-1) + (x_n - \mu_1 - v_0/2) \}^+ < h, \quad (3.36)$$

где  $\{a\}^+ = \max\{0, a\}$ ,  $g(0) = 0$ .

В случае (3.35) имеем:

$$\begin{aligned} g(n) &= \{ g(n-1) + (x_n - \mu_1 - v_0/2) \}^+ < h, \\ g(n) &= \{ g(n-1) - (x_n - \mu_1 + v_0/2) \}^+ < h, \end{aligned} \quad (3.37)$$

### *Алгоритмы обнаружения, основанные на принципе «невязок» [18]*

Невязка предполагает расхождение наблюдаемой и принятой моделей случайного процесса.

В случае модели случайного процесса AP(0) (3.32) решающая функция  $g(n)$  выглядит следующим образом:

$$g(n) = \frac{\sum_{i=1}^n \left( \frac{x_i - \mu_1}{\sigma_1} \right)^2 - 1}{\sqrt{2n}} < h, \quad (3.38)$$

а в случае модели AP(1) (3.33) имеем:

$$g(n) = \frac{\sum_{i=1}^n \left\{ \frac{|x_i - \mu_1 - \alpha_1(x_{i-1} - \mu_1)|^2}{\beta_1^2} - 1 \right\}}{\sqrt{2n}} < h \quad (3.39)$$

При отсутствии разладки  $M[g(n)] = 0$ , а при  $\mu_2 > \mu_1$  или при  $\mu_2 < \mu_1$  функция  $M[g(n)]$  начинает увеличиваться, причем в отличие от предыдущих алгоритмов обнаруживается разладка как в случае  $\mu_2 > \mu_1$ ,  $\mu_2 < \mu_1$ , так и в случае изменения других параметров  $\theta$  (дисперсии  $\sigma^2$  и коэффициента корреляции  $a_1$ ).

### *Алгоритм сигнального отношения [19]*

Решающая функция выглядит следующим образом:

$$g(n) = \frac{S_n}{R_n}, \quad (3.40)$$

где  $S_n = (1 - \alpha) S_{n-1} + \alpha e_n$ ,

$$R_n = (1 - \alpha) R_{n-1} + \alpha |e_n|,$$

$$e_n = (x_n - \mu_1),$$

$\alpha$  – коэффициент сглаживания,  $0 < \alpha < 1$ .

При отсутствии разладки предполагается, что  $M(e_n) = 0$ .

В случае возникновения разладки  $\mu_2 > \mu_1$  функция  $g(n)$  стремится к единице, а при  $\mu_2 < \mu_1$   $g(n)$  стремится к  $(-1)$ .

В случае отсутствия разладки  $M[g(n)] = 0$ , поэтому выставляя двусторонние пороги  $h_1, h_2$ , т.е.

$$-1 < h_1 < g(n) < h_2 < 1, \quad (3.41)$$

можно обнаружить не только факт наличия разладки, но и характер ее проявления. Более того, предлагаемый вниманию алгоритм обладает еще замечательной особенностью: возможностью обнаруживать многократные разладки с фиксацией как момента возникновения очередной разладки так и момента ее исчезновения.

При исследовании указанных алгоритмов возникают две задачи :

1. Выбрать порог обнаружения  $h$  и параметры алгоритма, гарантирующие требуемое качество обнаружения в виде предложенной системы критериев ( $P_{lo}$ ,  $\bar{\tau}_{obn}$ ).
2. Выбрать наилучший алгоритм с точки зрения минимизации времени  $\bar{\tau}_{obn}$  при фиксированном уровне ложного срабатывания  $P_{lo}$ .

Исследование обычно проводится в двух режимах работы моделирующего алгоритма:

**Режим 1** (при отсутствии разладки). В этом режиме:

- моделируется соответствующий случайный процесс, либо в соответствии с выбранной моделью наблюдаемый временной ряд;
- моделируется выбранный алгоритм обнаружения при фиксированных параметрах и порогах;
- производится оценка  $P_{lo}$  в виде частоты ложных срабатываний и выбирается порог, гарантирующий заданный уровень  $P_{lo}$ .

**Режим 2** (при наличии разладки). В этом режиме:

- в определенный момент времени  $t_0$  вводится разладка в соответствии с зафиксированным параметром вектора  $\theta$ ;

- производится оценка времени обнаружения для заданных порогов, накапливается статистика и строятся оценки  $\bar{\tau}_{обн}$  и  $\sigma_{\tau_{обн}}$ .

Точное аналитическое решение указанных задач связано со значительными математическими трудностями, поэтому соответствующее статистическое моделирование обычно является основным подходом при исследовании указанных алгоритмов. Однако при этом не следует отбрасывать приближенные математические модели по оценке  $P_{ло}$  и  $\bar{\tau}_{обн}$ . Они оказываются весьма полезными как для настройки имитационных моделей, так и для контроля точности проведенного статистического эксперимента.

### 3.3.2. Принципы настройки алгоритмов обнаружения

Настройка алгоритмов обнаружения разладки предполагает наличие точных или приближенных математических моделей для выбора параметров алгоритмов. Построение моделей связано обычно с серьезным математическим исследованием. Поэтому приведем окончательные результаты исследования по оценке показателей качества обнаружения с учетом двух выделенных режимов.

#### *Алгоритмы кумулятивных сумм*

Оценка среднего времени до ложной тревоги определяется в соответствии с [6]:

$$\bar{\tau}_{ло} = \frac{2\sigma^2}{(\mu_2 - \mu_1)^2 (1 - \sum_{i=1}^p a_i)^2} (e^h - h - 1). \quad (3.42)$$

Оценка среднего времени обнаружения разладки  $\bar{\tau}_{обн}$ :

$$\bar{\tau}_{обн} = \frac{2\sigma^2}{(\mu_2 - \mu_1)^2 (1 - \sum_{i=1}^p a_i)^2} (e^{-h} + h - 1), \quad (3.43)$$

где  $p$  – порядок модели авторегрессии,

$a_i$  – коэффициент авторегрессии,  $h$  – порог.

При выводе соотношений (3.42), (3.43) предполагалось использование алгоритма кумулятивных сумм в форме (3.37) при  $v_0 = 0$ .

### Алгоритм, основанный на принципе невязок

В соответствии с (3.39) решающую статистику  $g(n)$  можно представить следующим образом:

$$g_n = \frac{\sum_{i=1}^n (g_i - 1)}{\sqrt{2n}} , \quad (3.44)$$

где  $g_i = \frac{[(x_n - \mu_1 - a_i(x_i - \mu_1))]^2}{\beta_i^2}$ .

Вероятность ложного обнаружения для этого алгоритма может быть определена на основе использования марковских процессов [20]. Такой подход позволяет получить следующее выражение для  $P_{lo}$

$$P_{lo} = \left( 1 - \prod_{i=1}^{N_c} p_i \right) \left/ \left( 2 + \sum_{j=1}^{N_c-1} \prod_{i=1}^j p_i - \prod_{i=1}^{N_c} p_i \right) \right. \quad (3.45)$$

$$p_i = \begin{cases} F_i(h\sqrt{2i} + i) - F_i(-h\sqrt{2i} + i) & \text{при } h \leq \sqrt{i/2}; \\ F_i(h\sqrt{2i} + i) & \text{при } h > \sqrt{i/2}; \end{cases} \quad (3.46)$$

где  $F_i$  – функция  $\chi^2$  распределения с  $i$  степенями свободы.

Полученное значение  $P_{lo}$  в соответствии с (3.45) является гарантированной оценкой вероятности ложных срабатываний на периоде накопления  $N_c$ . При построении алгоритма предполагается, что в случае  $g_i \leq h$ ,  $i = \overline{1, N_c}$  в момент  $N_c$  решающая статистика  $g_n g_n$  обнуляется и процесс ее накопления начинается заново. Зависимость  $P_{lo} = F(h, N_c)$ , вычисленная в соответствии с (3.46), представлена в таблице 3.1.

**Зависимость  $P_{lo} = F(h, N_c) \cdot 10^3$**

Таблица 3.1

$N_c$	$h$			
	2	3	4	5
2	47.7	15.1	8.3	3.4
4	46.6	11.0	6.4	2.4
10	43.7	9.0	4.1	1.3
20	42.0	8.7	2.7	0.8
40	41.0	8.4	1.7	0.4
60	41.0	5.4	1.3	0.3
100	41.0	5.0	0.9	0.1

К сожалению, удобные в вычислительном отношении выражения для расчета среднего времени обнаружения получить не удается, поэтому далее в таблице 3.2 приводятся только оценки полученные методом имитационного моделирования при  $P_{lo} = 10^{-2}$ .

Таблица 3.2

*Зависимость  $\bar{\tau}_{обн} = F(h, N_c)$*

Значения параметров			Величина скачка $k = (\mu_2 - \mu_1)/\sigma_1$		
$N_c$	$h$	$a_I$	1	2	3
2	3.5	0.0	17	3.6	2
		- 0.5	5	2	1.4
		0.5	40	11	3
10	24	0.0	13	3	1.5
		- 0.5	4.8	1.7	1.2
		0.5	34	6.5	1.6
24	16	0.0	10	2.2	1.2
		- 0.5	3.6	1.5	1.1
		0.5	32	4.1	1.3
100	14.5	0.0	9.5	2	1.2
		- 0.5	3.4	1.5	1
		0.5	31	3.6	13

### *Алгоритм сигнального отношения*

Для алгоритма (3.40) получена в [19] аналитическая оценка вероятности ложных срабатываний с использованием аппроксимации распределения  $g(n)$  нормальным распределением с нулевым средним и дисперсией

$$\sigma_g^2 = \frac{\pi}{2} \frac{\alpha}{(2 - \frac{2}{\pi})^2 (2 - \alpha)} \quad (3.47)$$

при ограничении на  $\alpha \leq 0.176$ . В этом случае значение  $P_{lo}$  при симметричных порогах  $h_1 = h_2 = h$  определяется следующим образом:

$$P_{lo} = 1 - 2 \int_0^h \frac{1}{\sqrt{2\pi\sigma_g^2}} e^{-\frac{(g-h)^2}{2\sigma_g^2}} dg. \quad (3.48)$$

Оценка среднего времени обнаружения разладки в виде скачка математического ожидания выглядит так:

$$\bar{\tau}_{обн} = \left[ \frac{\ln \left[ \frac{r(1-h)}{r(1-h) + 2h/\sqrt{2\pi}} \right]}{\ln(1-\alpha)} \right] - 1, \quad (3.49)$$

где  $r = \mu_2/\sigma_1$ ,  $\mu_2 > 0$ ;  $r = 1, 2, \dots$ ;  $\mu_1 = 0$ ;  $\lceil x \rceil$  – ближайшее целое большее  $x$ . В таблице 3.3 приведены результаты вычисления  $P_{ло}$  в соответствии с (3.48).

Таблица 3.3

Зависимость  $P_{ло} = F(h)$  при  $\alpha = 0.15$

Порог $h$	0.1	0.16	0.23	0.30	0.37	0.46	0.58	0.77
Значение $P_{ло}$	0.72	0.59	0.46	0.34	0.23	0.14	0.04	0.01

Далее рассмотрим несколько иной подход к обнаружению разладки случайных процессов, основанный на идеи перекрывающихся интервалов (доверительных и толерантных) [21].

#### 3.4. Принятие решений с использованием доверительных и толерантных интервалов

Доверительный интервал вводится для неслучайной величины  $a$ .

Пусть имеем нормально распределенную величину  $x$  с законом  $N(a, \sigma)$ . Параметр  $\sigma$  считается известным. Оценка величины  $a$ :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.50)$$

распределена нормально с законом  $N\left(\bar{x}, \frac{\sigma}{\sqrt{n}}\right)$ .

Для любой доверительной вероятности  $1-q$  можно найти такой квантиль  $t_q$ , что

$$P(|x-a| < t_q \frac{\sigma}{\sqrt{n}}) = 1-q. \quad (3.51)$$

Неравенство:  $|x-a| < t_q \frac{\sigma}{\sqrt{n}}$

эквивалентно такому:

$$\bar{x} - t_q \frac{\sigma}{\sqrt{n}} < a < \bar{x} + t_q \frac{\sigma}{\sqrt{n}}, \quad (3.52)$$

следовательно интервал  $\left[\bar{x} - t_q \frac{\sigma}{\sqrt{n}}, \bar{x} + t_q \frac{\sigma}{\sqrt{n}}\right]$  для величины  $a$  будет

доверительным интервалом, соответствующим доверительной вероятности  $1-q$ .

Толерантным называется интервал, который с заданной вероятностью  $\gamma$  содержит не менее, чем заданную часть  $\delta$  всей совокупности случайной величины.

При этом предел допуска  $(\gamma, \delta)$  толерантного интервала говорит о том, что  $100 \delta \%$  будущих наблюдений с  $100 \gamma \%$  доверием будут находиться в этом интервале.

Обычно величины  $\gamma$  и  $\delta$  выбирают больше 0.9. Так как контролируемый процесс распределен по нормальному закону, то по выборочным характеристикам

$$\begin{aligned}\bar{x} &= \frac{1}{n} \sum_{i=1}^n x_i, \\ S_x &= \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2},\end{aligned}\tag{3.53}$$

вычисляем нижнюю  $I_1$  и верхнюю  $I_2$  толерантные границы по формулам:

$$\begin{aligned}I_1 &= \bar{x} - KS_x, \\ I_2 &= \bar{x} + KS_x,\end{aligned}\tag{3.54}$$

где  $K$  – толерантный множитель.

Для определения толерантных границ  $I_1$  и  $I_2$  следует сначала вычислить  $\bar{x}$  и  $S_x$ . Затем необходимо найти такое  $K$ , чтобы с доверительной вероятностью  $\gamma$ , близкой к единице, интервал  $[\bar{x} - KS_x, \bar{x} + KS_x]$  содержал заданную долю  $\delta$  всей совокупности рассматриваемой случайной величины  $x$ . Значение  $K$ , являющееся функцией  $n, \delta, \gamma$  приближенно выражается формулой:

$$K = Z_\infty \left( 1 + \frac{Z_\gamma}{\sqrt{2n}} + \frac{5Z_\gamma^2 + 10}{12n} \right),\tag{3.55}$$

где  $Z_\infty, Z_\gamma$  – абсциссы нормированной функции Лапласа  $\Phi_0(Z)$ , определяемые соответственно из следующих условий:

$$\begin{aligned}\delta &= 2 \Phi_0(Z_\infty), \\ \Phi_0(Z_\gamma) &= \gamma - 0.5,\end{aligned}\tag{3.56}$$

$$\text{где } \Phi_0(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{z^2}{2}} dz.$$

В [21] предложена теория обнаружения дефекта с использованием критерия перекрываемости двух областей: одной доверительной, построенной относительно номинальной траектории системы, в которой отсутствуют отказы, другой толерантной, построенной относительно оценок фильтра Калмана, получаемых в процессе обработки реальных измерений.

На основе построенных доверительного и толерантного интервалов для нормализованной обновляющей последовательности с использованием предложенного условия перекрываемости интервалов предлагается следующий подход к прогнозированию технического состояния системы.

Если указанные интервалы (доверительный и толерантный) перекрываются, то истинное значение  $a$  может находиться в любом из них и предполагается, что система будет функционировать правильно. В случае же, когда эти интервалы не пересекаются, состояние системы существенно отличается от номинального и предполагается, что в системе возникнет дефект.

Таким образом, задача принятия решения о техническом состоянии в рассматриваемой постановке сводится к нахождению единственной точки, присутствие которой одновременно в обоих указанных интервалах является необходимым и достаточным условием для их перекрытия, а отсутствие – необходимым и достаточным условием прогнозирования дефекта в системе.

При прогнозировании технического состояния системы в реальных условиях ее эксплуатации предложенный алгоритм сводится к следующей последовательности вычислений:

- 1) строится заранее доверительный интервал  $[u_1, u_2]$  для нормализованной последовательности  $\tilde{x}_i \in N(0, I)$ ,  $\tilde{x}_i = \frac{x_i - a}{\sigma}$ ;
- 2) вычисляется значение нормализованной последовательности  $\tilde{x}_i$  на данном шаге  $i$ ;
- 3) находятся значение выборочных средних и выборочных среднеквадратических отклонений величины  $\tilde{x}_i$  в соответствии с формулами:

$$\begin{aligned}\bar{\tilde{x}} &= \frac{1}{N} \sum_{j=i-N+1}^i \tilde{x}_j, \\ S_{\tilde{x}} &= \sqrt{\frac{1}{N-1} \sum_{j=i-N+1}^n (\tilde{x}_j - \bar{\tilde{x}})^2},\end{aligned}\tag{3.57}$$

где  $N$  – число используемых реализаций ("ширина окна");

4) на основании (3.54) – (3.56) при заданных  $\gamma$  и  $\delta$  определяются границы толерантного интервала  $[l_1, l_2]$  для  $\tilde{x}_t$ , говорящие о том, что  $100 \delta \%$  будущих реализаций  $\tilde{x}$  с  $100 \gamma \%$  доверием будут находиться в интервале  $[l_1, l_2]$ ;

5) проверяется условие перекрываемости интервалов  $[u_1, u_2]$  и  $[l_1, l_2]$ . Для этого достаточно проверить выполнимость условий:

$$\begin{aligned} u_2 &\geq l_1, \\ l_2 &\geq u_1. \end{aligned} \tag{3.58}$$

Невыполнение хотя бы одного из неравенств (3.58) говорит о возможном возникновении дефекта в системе, при этом принимается решение о необходимости введения управляющих воздействий на объект. В противном случае предполагается, что система функционирует нормально и последовательность вычислений повторяется с п.2, для следующего момента времени.

Таким образом, использование, например, условия перекрываемости доверительного и толерантного интервалов для обновляющей последовательности фильтра Калмана позволяет решать задачу прогнозирования технического состояния динамических систем в условиях реальной эксплуатации [21].

### 3.5. Распознавание образов

Определим основные понятия.

**Образ (класс)** – множество всех объектов (явлений, процессов, ситуаций), «сходных между собой в некотором фиксированном отношении».

**Распознавание объекта** – определение номера или наименования образа (класса), которому данный объект принадлежит.

**Решающее правило** (алгоритм) – некоторая разделяющая (дискриминантная) функция или система дискриминантных функций, позволяющая распознать объект.

**Обучающая выборка** – множество объектов, которыми образы (классы) представлены при обучении.

**Экзаменующая (прверочная) выборка** – множество объектов, на которых проверяются результаты обучения.

**Признак** – количественное описание того или иного свойства объекта

Остановимся кратко на классификации задач распознавания образов и методов их решения. Можно выделить три типа задач распознавания образов [5].

1. Задачи распознавания при наличии «обучения с учителем».
2. Задачи таксономии (автоматической классификации, кластер-анализа).
3. Задачи выбора информативных признаков из заданной системы описания объектов.

Наиболее распространенными в области функциональной диагностики являются задачи первого типа. Суть их состоит в том, чтобы в результате анализа обучающей последовательности конечной длины построить алгоритм классификации объектов, принадлежащих заданному классу.

Существует большое количество методов решения задачи первого типа. Все эти методы можно разделить на два класса: статистические и детерминированные.

В статистических методах построения решающих правил предполагается, что каждый объект с некоторой вероятностью может принадлежать любому образу из заданного алфавита.

Рассмотрим некоторые методы из указанных классов.

### *Детерминированные методы*

Класс задан совокупностью объектов в выбранном пространстве признаков:

$$x^{(S)} = \{x_1^{(S)}, x_2^{(S)}, \dots, x_i^{(S)}, \dots, x_n^{(S)}\}.$$

где  $S$  – индекс класса  $S = \overline{1, l}$ ;

$i$  – индекс признака,  $i = \overline{1, n}$ ;

$N_S$  – число объектов заданного класса.

#### *1. Геометрический подход*

Вычисляются оценки:

$$a_i^{(S)} = \frac{1}{N_S} \sum_{i=1}^{N_S} x_i^{(S)}; \quad (3.59)$$

$$\sigma_i^{(S)} = \sqrt{\frac{1}{\sqrt{N_S - 1}} \sum_{k=1}^{N_S} (x_k^{(S)} - a_i^{(S)})^2} \quad (3.60)$$

При вычислении оценок (3.59), (3.60) используется обучающая выборка для каждого класса объемом  $N_S$ , где  $S = \overline{I, l}$ .

Правило принятия решений при использовании только оценок (3.59) выглядит следующим образом: относим неизвестный объект  $x$  к тому классу, где достигается

$$\min_S \|x - a^{(S)}\|, \quad (3.61)$$

где  $\|y\| = \sqrt{\sum_{i=1}^n y_i^2}$ .

Учет оценок (3.59), (3.60) воспроизводит правило (3.61) только в безразмерном нормированном пространстве, где

$$\|y\| = \sqrt{\sum_{i=1}^n I \frac{y_i}{\sigma_i} J^2}. \quad (3.62)$$

## 2. Метод потенциальных функций [22]

Предполагается, что каждый объект обладает некоторым потенциалом, максимальным в данной точке и убывающим по всем направлениям от нее. В качестве потенциальной функции можно выбрать, например, функцию:

$$\varphi(R) = \frac{P}{1 + \alpha R^2}, \quad \alpha > 0, \quad (3.63)$$

где  $P$  – коэффициент, называемый весом точки,

$R$  – расстояние между точкой-источником и точкой, в которой вычисляется потенциал.

Величину  $\varphi$  в каждой точке пространства можно считать мерой близости этой точки к точке-источнику.

Будем распознавать объекты из обучающего множества  $X^{ob}$ . Если возникает такая ситуация, что некоторые объекты из  $X^{ob}$  неправильно распознаются, т.е. относятся не к своему классу, то увеличением весов этих объектов можно достичь правильного распознавания всех объектов множества  $X^{ob}$ . Таким образом, процесс обучения сводится к подбору весовых коэффициентов объектов обучающего множества.

Данный подход может быть реализован в виде следующего алгоритма:

1. Веса всех точек множества  $X^{ob}$  принимаются равными единице.
2. В каждой точке  $x \in X^{ob}$ , где  $x = (x_1, x_2, \dots, x_n)$ , вычисляется потенциал всех классов по формуле:

$$\Phi(x, X_S^{ob}) = \frac{1}{N_S} \sum_{r=1}^{N_S} \frac{P(x)}{1 + \alpha \sum_{i=1}^n (x_i - x_i^{(r)})^2}, \quad (3.64)$$

где  $N_S$  – количество векторов во множестве  $X_S^{ob}$ ,  $S = \overline{I, l}$ ;  $\alpha = 1/n$ .

Если окажется, что в некоторой точке  $\tilde{x} \in X_S^{ob}$ :

$$\Phi(\tilde{x}, X_I^{ob}) \leq \Phi(\tilde{x}, X_l^{ob}), \quad I \neq S, \quad (3.65)$$

то вес этой точки увеличивается на  $\Delta P = 1$ .

3. Если все точки из множества  $X^{ob}$  распознаются правильно, то этап обучения заканчивается и проводится распознавание проверочных и рабочих объектов. В противном случае происходит переход к выполнению п.2.

Выполнению алгоритма обучения обычно предшествует операция нормирования признаков. Для этого в формуле (3.63) константа  $\alpha$  заменяется на величину  $k_i = \alpha / (\Delta x_i)^2$ , где  $\Delta x_i$  – разность между максимальным и минимальным значением  $i$ -го признака, найденная в результате анализа обучающей информации.

### *Статистические методы описания объектов*

Для каждого класса используется модель многомерного нормального распределения с параметрами  $a^{(S)}$ ,  $M_S$ , где  $a^{(S)}$  – вектор математических ожиданий,  $M_S$  – корреляционная матрица,  $S = \overline{I, l}$ . Для распознавания используется метод, основанный на использования функции правдоподобия [4].

Неизвестный объект  $x$  относим к тому классу, где решающая функция, а также логарифм функции правдоподобия

$$-\frac{1}{2}(x - a^{(S)})^T M_S^{-1} (x - a^{(S)}) - \frac{1}{2} \ln |\det M_S| \quad (3.66)$$

достигает максимума,  $S = \overline{I, l}$ .

Если  $M_S = M$  для  $S = \overline{I, l}$ , то правило (3.66) упрощается:

$$-\frac{1}{2}(x - a^{(S)})^T M^{-1} (x - a^{(S)}) \Rightarrow \max. \quad (3.67)$$

В случае, если признаки независимы, то правило (3.67) выглядит следующим образом:

$$\sum_{i=1}^n I \frac{x_i - a_i^{(s)}}{\sigma_i^s} I^2 \Rightarrow \min, \quad S = \overline{I, l} . \quad (3.68)$$

Приведенные выше подходы к принятию решений на основе алгоритмов распознавания образов не исчерпывают всего их разнообразия, а показывают только основные принципы их построения.

В следующем разделе анализируются алгоритмы принятия решений с использованием методов искусственного интеллекта, а в частности нейронных сетей.

## 4. Решение задач функциональной диагностики с использованием аппарата нейронных сетей

Перспективным направлением развития методов и средств диагностирования является их интеллектуализация, позволяющая не только улучшить характеристики систем диагностирования, но и расширить их функциональные возможности. Основой для интеллектуализации систем диагностирования является аппарат нейронных сетей (НС), нечеткой логики и генетических алгоритмов. Также как и решение задач распознавания образов с использованием традиционных (классических) методов, указанные математические аппараты основаны на знаниях. Форма представления знаний, их использование и обработка отличаются от описанных в главе 3.

В данном разделе рассматриваются основные модели нейронных сетей, их свойства, алгоритмы обучения и подходы к построению нейросетевых решений некоторых задач функциональной диагностики.

### 4.1. Основные понятия теории НС

Нейронная сеть представляет собой многослойную сетевую структуру взаимосвязанных однотипных обрабатывающих элементов - искусственных нейронов (добавление слова искусственный подчеркивает огромную разницу между биологическим и формальным нейронами).

Искусственный нейрон (иногда называемый пороговым элементом) состоит из  $n$  входов  $x_i$  ( $i = 1..n$ ), весовых коэффициентов входов (весов)  $w_i$ , функции активации  $a$  и передаточной (выходной) функции  $\varphi$ . Структура искусственного нейрона (в дальнейшем, просто нейрона) представлена на рис. 4.1.

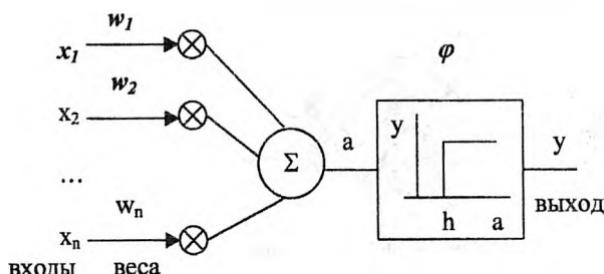


Рис. 4.1 Структура простейшего нейрона

Математическое описание функционирования простейшего статического нейрона (значение выходов нейрона зависят только от значений входных сигналов и не зависят от значений функции активации или выхода нейрона в предыдущие моменты времени) с пороговой передаточной функцией:

$$a = \sum_{i=1}^n w_i x_i, \quad (4.1)$$

$$y = \varphi(a) = \begin{cases} 1, & \text{если } a \geq h \\ 0, & \text{если } a < h \end{cases}, \quad (4.2)$$

где  $x_i$  –  $i$ -й вход нейрона,

$w_i$  –  $i$ -й весовой коэффициент,

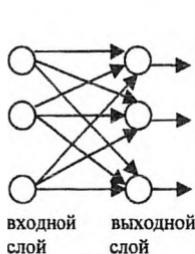
$a$  – функция активации,

$y$  – выход нейрона,

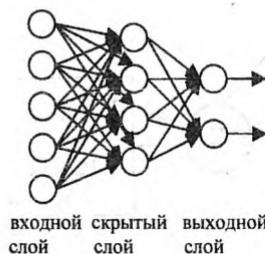
$\varphi$  – передаточная (выходная) функция,

$h$  – порог.

Разнообразие моделей нейронных сетей определяется существованием различных моделей нейронов (возможны разные виды функции активации и передаточной функций) а также топологий их соединения и взаимодействия (сети прямого распространения сигнала, с обратными связями и т.д.) [24]. Кроме того, могут быть различные алгоритмы обучения НС. Возможные варианты объединения нейронов в НС показаны на рис. 4.2 - 4.4. При этом группы нейронов образуют слои НС, а слои, соответственно - многослойные сети нейронов. Входной слой нейронов (фактически играющих роль рецепторов) в подсчете общего числа слоев сети не учитывают.



а)



б)

Рис. 4.2 а) Однослойная сеть прямого распространения  
б) Двухслойная сеть прямого распространения

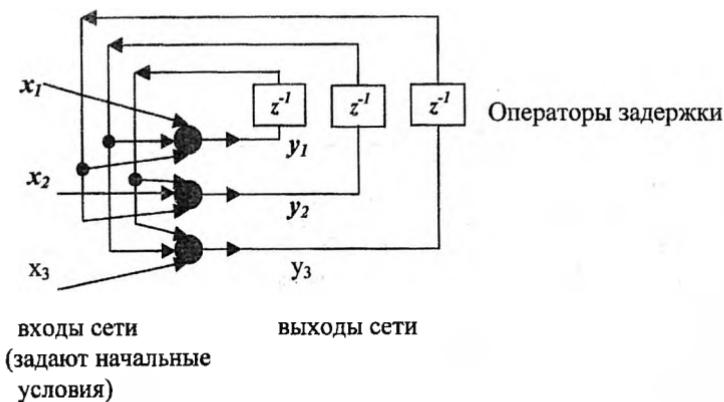


Рис. 4.3 Сеть с обратными связями (рекуррентная сеть) без скрытых слоев и без связей нейронов типа «сам на себя» (петель)

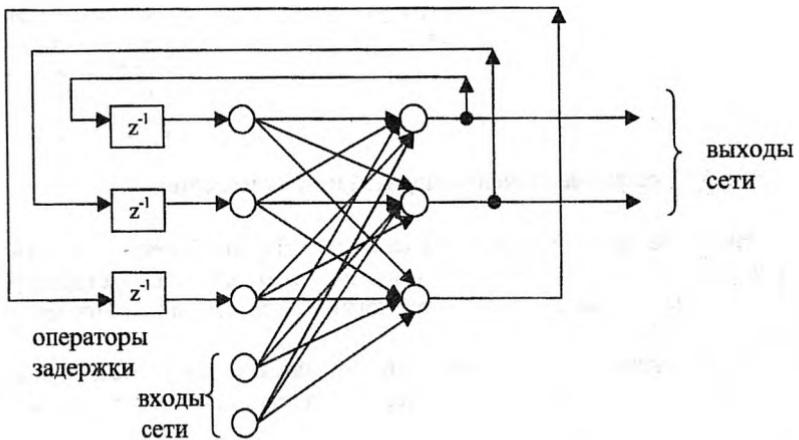


Рис. 4.4 Рекуррентная сеть со скрытым слоем

Основное деление моделей нейронных сетей на дискретные и континуальные (непрерывные) осуществляется по виду возможных значений входов  $x_i$  и выходов  $y_i$  нейронов. Составные элементы, определяющие модель нейронной сети, представлены на рис. 4.5, причем некоторые из них в отдельных случаях взаимосвязаны.

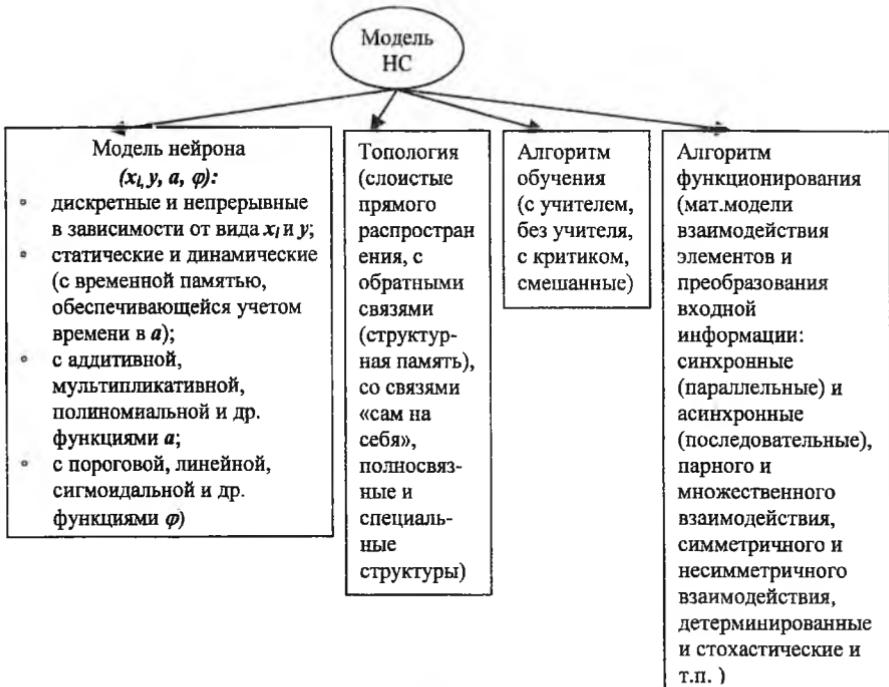


Рис. 4.5 Составные элементы моделей нейронных сетей

Наиболее применяемые типы выходных функций нейронов показаны на рис. 4.6. Они могут использоваться в нейронах для любых структур НС, рассмотренных выше и кроме того могут иметь смещение относительно начала координат.

Обрабатывающие способности НС хранятся в весах межэлементных соединений, полученных в процессе обучения (адаптации) по набору обучающих примеров.

Особенности нейровычислений:

- информация хранится в весах, а не в программах или данных;
- НС оперируют с "невидимыми" образами, которые получает на основе обобщения из обучающей последовательности;
- НС устойчива к наличию зашумленных сигналов на входе - небольшое изменение сигнала на входе не повлияет на выходное значение;
- НС устойчива к выходу из строя отдельных элементов сети за счет параллелизма и распределенной обработки - снижается только качество функционирования, но НС может продолжать работать;

- НС не требует традиционного программирования, так как сама обучается решению различных задач;
- НС является универсальной однородной вычислительной средой - разные задачи превращаются в однотипные;
- нейровычисления являются высокопараллельными и поэтому обеспечивают высокое быстродействие.

В связи с наличием у НС указанных особенностей они нашли широкое применение во многих областях науки и техники для решения как плохо формализуемых задач так и для задач классической математики.

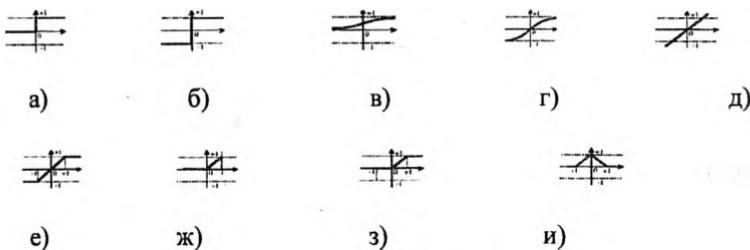


Рис. 4.6 Передаточные функции нейронов

- а), б) шаговые несимметричные и симметричные функции  
 в), г) сигмоидальные несимметричные и симметричные  
 функции  
 д) линейная функция  
 е) линейная с зонами насыщения  
 ж), з) линейная положительная без и с зонами насыщения  
 и) треугольная функция

Для задач диагностики особенно важны возможности НС принимать решения в условиях нечетких, неполных, зашумленных данных в условиях изменяющейся окружающей среды. Кроме того, появляется возможность раннего обнаружения небольших неисправностей с различным характером протекания во времени и возможность контроля процесса в его переходных состояниях.

Аппарат НС является развитием методов распознавания образов и позволяет решать не только задачи классификации и кластеризации, но и многие другие, в том числе те, которые могут быть представлены как задачи распознавания образов. Необходимо отметить, что многие модели НС представляют собой реализацию известных классических методов решения задач распознавания, представленных в нейросетевом логическом базисе (с учетом операций, выполняемых нейронами) и обеспечивают параллельность в получении решения. В этом случае НС являются только средством реализации – заложенные в них алгоритмы функционирования соответствуют определенным алгоритмам распознавания.

## 4.2. Типовые структуры НС

По структуре связей НС могут быть сгруппированы в два основных класса: сети прямого распространения и сети с обратными связями [23].

В сетях *прямого распространения* (см. рис.4.2.) нейроны первого слоя получают входные сигналы  $x$ , преобразуют их и передают нейронам второго слоя. Далее срабатывает второй слой и т.д. до  $l$ -го слоя, который выдает выходные сигналы сети. Чаще всего каждый выходной сигнал  $j$ -го слоя подается на вход всех нейронов  $j+1$ -го:

$$\begin{aligned} y_i^{j+1} &= \phi\left(\sum_{k=1}^{N_j} w_{ik}^{j+1} y_k^j\right), \quad i=1..N_{j+1}, \\ y_i^j &= \phi\left(\sum_{j=1}^n w_{ij}^j x_j\right), \quad i=1..N_l, \end{aligned} \quad (4.3)$$

где  $y^{j+1}$  – выходное значение  $i$ -ого нейрона  $j+1$ -ого слоя,

$w_{ik}^{j+1}$  – весовой коэффициент связи  $k$ -ого нейрона  $j+1$  слоя с  $i$ -ым нейроном  $j$ -ого слоя,

$n$ -число входных нейронов (размерность входного сигнала),

$N_j$  – число нейронов в  $j$ -ом слое.

Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. Стандартный способ подачи входных сигналов: все нейроны первого слоя получают каждый входной сигнал.

Сети прямого распространения являются статическими в том смысле, что на заданный вход они вырабатывают одну совокупность выходных значений, не зависящих от предыдущего состояния сети.

Сети с обратными связями являются динамическими, так как в силу обратных связей в них модифицируются входы нейронов, что приводит к изменению состояния сети. Основные модели НС с обратными связями - сети ассоциативной памяти (сети Хопфилда), соревновательные сети и их разновидности.

*Сеть ассоциативной памяти* – сеть Хопфилда по топологии представляет собой однослойную рекуррентную сеть без петель - элементы сети связаны по принципу «каждый с каждым», без образования связи «сам на себя». При этом связи симметричны по весовым значениям. На рис. 4.3 представлена сеть, состоящая из трех нейронов.

Вначале на сеть подается входной набор сигналов, задающих начальные условия функционирования сети, а затем за счет обратных связей продолжается процесс изменения выходных значений нейронов сети до тех пор, пока не установится устойчивое состояние сети. Функционирование сети может быть представлено следующим образом:

$$y_j(\theta) = x_j, j = 1..N,$$

$$y_j(t+1) = \phi\left(\sum_{i=1}^N w_{ij} y_i(t)\right), j = 1..N, \quad (4.4)$$

где  $w_{ij}$  –  $i$ -й вес  $j$ -го нейрона,

$x_j$  –  $j$ -й элемент входного сигнала сети,

$y_j$  – выход  $j$ -го нейрона,

$N$  – количество элементов (размерность) входного сигнала – количество нейронов в сети.

Функционирование сети заканчивается, если на некотором шаге  $T$  для всех  $j$  выполняется:

$$y_j(T) = y_j(T-1), \quad (4.5)$$

что говорит о достижении устойчивого состояния.

Для описания динамики поведения НС Хопфилда используется понятие энергии сети, которая является функцией от выходных значений нейронов и их весовых коэффициентов:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} y_i y_j. \quad (4.6)$$

В работах Хопфилда и его последователей показано, что в процессе функционирования сети уменьшается ее энергетическая функция. При этом устойчивые состояния сети находятся в точках минимума энергии, так что из любого начального состояния сеть придет к одному из локальных минимумов функции энергии сети.

Это основное свойство НС Хопфилда используется для построения НС ассоциативной памяти, когда необходимо по неполному или недостоверному входному образу восстановить полный образ, хранящийся в памяти совместно с другими образами-эталонами. Понятно, что образы, подлежащие хранению в памяти должны быть связаны с точками минимумов функции энергии сети. Для этого существует правило формирования весовых коэффициентов – правило Хебба (см. соотношение 4.13), приведенное в п.п.4.3. В этом случае НС осуществляет преобразование входного вектора  $x$  так, чтобы на выходе получить вектор  $y$ , представляющий собой один из хранящихся в памяти эталонов и такой, что больше всего соответствует входному вектору  $x$  и является правильным ответом. Однако для корректной работы НС должен выполняться ряд ограничений, связанных с емкостью памяти, с коррелированностью образов-эталонов и наличием нежелательных минимумов энергии, которые появляются в дополнение к энергетическим минимумам образов-эталонов, а также со степенью искажения входного образа [23, 24].

В отличие от решения задачи восстановления исходного образа по зашумленному с использованием кодовых расстояний (расстояния Хэмминга, евклидова расстояния и других) в ассоциативной памяти нет

необходимости в хранении всех образов-эталонов в явном виде, информация о них хранится в виде обобщенного образа, представленного весовыми коэффициентами связей между нейронами сети, число нейронов соответствует размерности хранимых образов.

*Соревновательные сети* работают по принципу «победитель получает все». На рис. 4.7 представлена простая соревновательная сеть. Первый слой – входной, распределяющий информацию по узлам следующего слоя в соответствии с коэффициентами входного слоя, а второй слой – соревновательный и полносвязный в пределах этого слоя: каждый узел связывается положительной связью с самим собой и отрицательными связями – со всеми остальными узлами, входящими в соревновательный слой:

$$w_{ij}^{cc} = \begin{cases} 1, & i = j \\ -\varepsilon, & i \neq j, \varepsilon \leq \frac{1}{N}, \quad i = 1..N, j = 1..N, \end{cases} \quad (4.7)$$

где  $w_{ij}^{cc}$  – весовые коэффициенты связей нейронов внутри соревновательного слоя.

За счет такой организации связей в соревновательном слое на любое входное воздействие будет «откликаться» только один нейрон, то есть на выходе соревновательного слоя после окончания переходного процесса, связанного с наличием внутренних обратных связей нейронов, будет активным (ненулевое выходное значение) только один «победивший» нейрон.

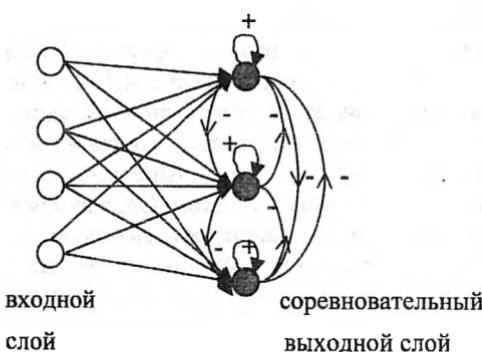


Рис. 4.7 Простая соревновательная сеть

Соревновательная динамика может быть представлена следующими соотношениями:

$$a_i(\theta) = \sum_{j=1}^M w_j x_j, \quad i = 1..N$$
$$a_j(t+1) = a_j(t) - \varepsilon \sum_{i \neq j} a_i(t), \quad i = 1..N, j = 1..N, \quad (4.8)$$

где  $w_j$  – вес  $i$ -ого входа и  $j$ -го нейрона соревновательного слоя,

$x_j$  –  $j$ -й элемент входного сигнала сети,

$a_j$  – активация  $j$ -го нейрона,

$M$  – количество элементов (размерность) входного сигнала,

$N$  – количество нейронов в соревновательном слое.

На каждой итерации большие сигналы активации нейронов подавляют более слабые сигналы активации других нейронов. Если в начале функционирования сети один из нейронов имел большую активацию в ответ на входное воздействие, то в конце функционирования все выходы нейронов, кроме максимального, будут иметь значения, близкие к нулю. Таким образом нейрон с наибольшей активацией побеждает.

Функционирование сети заканчивается, если на некотором шаге  $T$  для всех  $j$  выполняется соотношение:

$$a_j(T) = a_j(T-1).$$

При этом  $y_j = \phi(a_j)$  для всех  $j = 1..N$ .

Иногда моделирование работы соревновательной сети удобнее проводить без моделирования соревновательной динамики, представленной соотношением 4.8, а просто поиском максимальной активации и последующим определением выходного значения нейрона с максимальной активацией.

### 4.3. Основные алгоритмы обучения

Под обучением нейронной сети подразумевается процесс настройки весовых коэффициентов входов нейронов и порогов нейронов (в случае их ненулевых значений) с целью получения адекватной реакции сети на предъявляемый входной набор. Основные парадигмы обучения: с учителем, без учителя (самообучения) и смешанная. В первом случае нейронная сеть располагает правильными ответами на каждый входной пример, в то время как обучение без учителя не требует знания правильных ответов на каждый пример обучающей выборки. При смешанном обучении часть весов определяется посредством обучения с учителем, а оставшаяся часть – с помощью самообучения. Существует обучение с критиком, когда нейронной сети сообщается в правильном направлении или нет идет формирование выходного сигнала.

Наиболее распространены 3 правила обучения, на основе которых построены соответствующие алгоритмы обучения [25].

*Обобщенное дельта правило  
и алгоритм обратного распространения ошибки.*

Данный алгоритм используется в основном для обучения сетей прямого распространения, хотя возможно его применение и для рекуррентных сетей.

Введем следующие обозначения, предполагая, что рассматривается  $I$ -слойная НС с  $m$  входами и нейронами с аддитивной функцией активации и передаточной функцией  $\varphi$  и справедливы соотношения 4.3 :

$w = [w^1, w^2, \dots, w^{I+1}]$  – вектор-строка весовых коэффициентов всей сети, состоящий из ( $I+1$ )-ой матрицы размерностью  $N_k * N_{k-1}$  ( $N_j$  – число нейронов в  $j$ -ом слое),

$w^l = [w_1^{lT}, w_2^{lT}, \dots, w_m^{lT}]^T$  – вектор-столбец весовых коэффициентов первого слоя сети (на него поступают «взвешенные» сигналы с входного слоя),

$w^k = [w_1^{kT}, w_2^{kT}, \dots, w_{N_k}^{kT}]^T$  – вектор-столбец весовых коэффициентов нейронов  $k$ -ого слоя,

$w_i^k = [w_{i1}^k, w_{i2}^k, \dots, w_{iN_{k-1}}^k]$  – вектор-строка весовых коэффициентов  $i$ -ого нейрона  $k$ -ого слоя,

$w_{ij}^k$  – весовой коэффициент, связывающий  $j$ -ый нейрон  $k$ -ого слоя с  $i$ -ым нейроном предшествующего ( $k-1$ )-ого слоя,

$t^l = (t_1^l, \dots, t_{N_l}^l)$  – вектор желаемых значений выходов нейронов  $I$ -го слоя,

$y^l = (y_1^l, \dots, y_{N_l}^l)$  – вектор действительных выходов нейронов  $I$ -го слоя,

$\Delta w$  – изменение веса,

$\alpha$  – скорость обучения,

$E$  – общая ошибка выхода.

Пусть  $I$ -ый слой – выходной слой сети из  $N_l$  нейронов, тогда

$$E = \frac{1}{2} \sum_{j=1}^{N_l} (t_j^l - y_j^l)^2 \quad . \quad (4.10)$$

Общая ошибка выхода является функцией весов  $w_{ij}^l$ , которые, являясь ее параметрами, должны быть оптимизированы для минимизации этой функции. Цель этой оптимизации – приблизиться (а в лучшем случае и достичь) глобального минимума на многомерной поверхности ошибки.

Один из путей достижения этого – метод градиентного спуска.

Отрицательный градиент  $-\frac{dE}{dw_{ij}^l}$  поверхности ошибки показывает локальное направление спуска и, таким образом направление поиска нового веса на пути к глобальному минимуму (иногда и к локальному).

Вычисление нового значения весов сети вычисляется в соответствии со следующим выражением:

$$w_{ij}^l(n+1) = w_{ij}^l(n) + \Delta w_{ij}^l,$$

где изменение веса  $\Delta w_{ij}^l$  на  $(n+1)$ -ом шаге оптимизации функции ошибки пропорционально отрицательному градиенту  $-\frac{dE}{dw_{ij}^l}$ :

$$\Delta w_{ij}^l = -\frac{\alpha \cdot dE}{dw_{ij}^l}$$

Коэффициент пропорциональности  $\alpha$  – это скорость обучения, определяющая величину каждого шага по поверхности ошибки. Выбор большого коэффициента скорости обучения может привести к невозможности достижения минимума – система будет колебаться возле него. Также выбор коэффициента скорости обучения слишком маленьким приведет к медленной сходимости.

Градиент общей ошибки вычисляется следующим образом:

$$\frac{dE}{dw_{ij}^l} = \frac{dE}{dy_i^l} \cdot \frac{dy_i^l}{da_i^l} \cdot \frac{da_i^l}{dw_{ij}^l},$$

$$\text{где } a_i^l = \sum w_{ij}^l y_j^{l-1},$$

$$y_i^l = \varphi(a_i^l).$$

Так как

$$\frac{dE}{dy_i^l} = \frac{d\left(\frac{1}{2} \cdot \sum (t_i^l - y_i^l)^2\right)}{dy_i^l} = -(t_i^l - y_i^l),$$

$$\frac{dy_i^l}{da_i^l} = \frac{d\varphi(a_i^l)}{da_i^l} = \varphi'(a_i^l),$$

$$\frac{da_i^l}{dw_{ij}^l} = \frac{d(\sum w_{ij}^l y_j^{l-1})}{dw_{ij}^l} = y_j^{l-1},$$

то получим:

$$\frac{dE}{dw_{ij}^l} = -(t_i^l - y_i^l) \cdot \varphi'(a_i^l) \cdot y_j^{l-1}.$$

Отсюда, получаем выражение для обновления весов:

$$\Delta w_{ij}^l = \alpha \cdot (t_i^l - y_i^l) \cdot \varphi'(a_i^l) \cdot y_j^{l-1} = \alpha \cdot \delta_i^l \cdot y_j^{l-1}, \quad (4.11)$$

где величина  $\delta_i^l = (t_i^l - y_i^l) \cdot \varphi'(a_i^l)$  называется локальной ошибкой.

Полученное выражение отображает суть простого *дельта правила*. Но простое дельта правило может быть применено только к однослойным нейросетям, так как ошибка  $\delta_i^l$  может быть посчитана только для нейронов выходного слоя. В многослойных сетях это правило может быть применено только к весам нейронов последнего слоя.

Для скрытых слоев значение  $\delta_i^k$  не может быть определено непосредственно, таким образом, нам необходимо получить выражение для локальной ошибки  $\delta_i^k$ , не зависящее от  $y_i^l$ .

Выполняя аналогичные рассмотренным выше преобразования для скрытого слоя  $k$  и предшествующего ему слоя  $k-1$ , получим:

$$\Delta w_{ij}^k = \alpha \cdot \delta_i^k \cdot y_j^{k-1},$$

$$\delta_i^k = \varphi'(a_i^k) \cdot \sum_{j=1}^{N_{k+1}} \delta_j^{k+1} \cdot w_{ji}^{k+1}, \quad (4.12)$$

где  $N_{k+1}$  – число нейронов  $k+1$  слоя, с которыми связан  $i$ -ый нейрон  $k$ -ого слоя.

Соотношения 4.12 являются обобщенным дельта правилом обновления весовых коэффициентов (обобщение дельта правила 4.11 на скрытые слои).

Алгоритм обучения на базе обобщенного дельта правила – алгоритм обратного распространения ошибки, представляет собой следующую последовательность шагов:

1. Из имеющейся обучающей последовательности выбирается очередное значение входного сигнала и подается на вход НС. В результате получается выходной сигнал НС в ответ на входное воздействие.
2. По полученному выходному значению НС и соответствующему желаемому выходному значению из обучающей последовательности вычисляется общая ошибка и локальные ошибки. Затем проводится изменение весовых коэффициентов сети в соответствии с соотношениями 4.11 (для выходного слоя) и соотношениями 4.12 (для скрытых слоев). Общая ошибка от примера к примеру накапливается и запоминается.
3. Если выбраны все обучающие примеры из обучающей последовательности, то переходят к шагу 4, иначе – к шагу 1.
4. Проводится сравнение текущей накопленной общей ошибки обучения с заданной. Если она больше максимально допустимой, то ее обнуляют и переходят к шагу 1. Иначе – обучение закончено.

Из представленного выше описания шагов алгоритма обучения видно, что в процессе обучения обучающая последовательность предъявляется НС много раз, каждое предъявление всех примеров обучения часто называют проходами обучения.

Представленный алгоритм является общим алгоритмом обучения. Для пункта 4 обучения возможны различные модификации. Например, закончить обучение, если изменение суммарной ошибки за текущий проход меньше заданного предела или если достигнуто заданное число проходов обучения и т.п.. Так же существуют различные модификации правила формирования новых весовых коэффициентов: для ускорения сходимости процедуры обучения может быть включена информация о второй производной функции ошибки, может быть включена информация о предыдущих изменениях весовых коэффициентов (так называемое правило моментов) и др.

Необходимо отметить, что изменение параметров нейронной сети не ставит своей целью получить точное соответствие между входным сигналом и требуемым выходным. Это связано с тем, что задача получения точного решения для каждого примера не ставится; решение должно быть правильным для большинства примеров, поэтому изменение параметров нейронной сети производится в направлении требуемого ответа, но с множителем - скоростью обучения (или пределом обучения). Таким образом формируется правильное решение не в частном, а в целом, и работоспособность сети оценивается не только точностными характеристиками, но и вероятностными.

В общем случае, при обучении НС предполагается, что кроме настройки весовых коэффициентов происходит настройка пороговых значений нейронов (или смещений характеристик нейронов (см. рис. 4.1 и рис. 4.7)). Для унификации процедуры обучения на модели простейшего нейрона выполняется преобразование, позволяющее проводить вычисление новых значений пороговых коэффициентов по тем же соотношениям, что и вычисление значений весовых коэффициентов.

Для этого, вводится дополнительный вход нейрона с постоянным входным воздействием, равным “-1” и весовым коэффициентом  $h$  (см. рис. 4.8). В этом случае можно рассматривать несмещенные характеристики выходных функций нейронов (считать порог сравнения в соотношении 4.2 равным “0”), а подстройку порога  $h$  (смещения) проводить как обычного весового коэффициента по соотношениям 4.11 и 4.12.

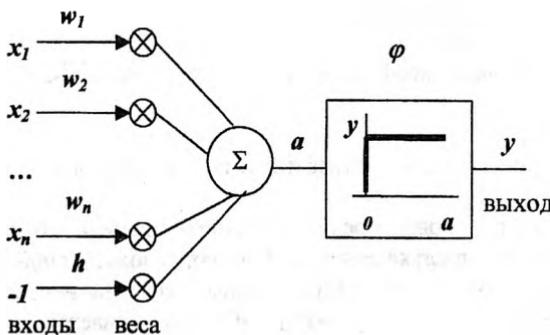


Рис.4.8 Структура простейшего модифицированного нейрона

Соотношение 4.2 в этом случае приобретает следующий вид:

$$y = \varphi(a) = \begin{cases} 1, & \text{если } a \geq 0 \\ 0, & \text{если } a < 0 \end{cases} .$$

Алгоритм обучения в этом случае не изменяется, только увеличивается на единицу размерность входных сигналов нейронов, дополнительные входы обучаются также как и все остальные входы нейронов .

*Правило Хебба* определяет вес связи между  $i$ -м и  $j$ -м нейроном в соответствии со следующим выражением:

$$w_{ij} = \begin{cases} \sum_{k=1}^p x_i^k \cdot x_j^k, & i \neq j \\ 0, & i = j \end{cases}, \quad (4.13)$$

где  $p$  – количество входных наборов данных, подлежащих сохранению в сети ассоциативной памяти,  $x_m^k$  – значение, подлежащее хранению в  $m$ -ом нейроне для  $k$ -ого образа-эталона.

Обучение НС Хопфилда в отличие от обучения НС прямого распространения сигнала предполагает вычисление весовых коэффициентов только один раз без дальнейшего их изменения.

*Обучение методом соревнования*, удовлетворяющее принципу «победитель получает все» выполняется в соответствии со следующими соотношениями:

$$\begin{aligned} w_{ij}(n+1) &= w_{ij}(n) + \Delta w_{ij}, \\ \Delta w_{ij} &= \alpha \cdot (x_j - w_{ij}) \cdot y_i, \quad i=1..N, \quad j=1..M, \end{aligned} \quad (4.14)$$

где  $N$  – число нейронов в соревновательном слое,

$M$  – число входных нейронов,

$\Delta w_{ij}$  – изменение весового коэффициента  $i$ -ого нейрона, связанного с  $j$ -ым входом,

$x_j$  –  $j$ -ый вход НС,

$w_{ij}(n+1)$  – значение весового коэффициента на очередном шаге обучения.

Алгоритм обучения на основе соревновательного правила обучения состоит из последовательного предъявления на вход соревновательной сети входных обучающих сигналов и изменении весовых коэффициентов по соотношению 4.14. Обучение (самообучение) НС заканчивается, когда нейронной сети будет показана вся обучающая последовательность, состоящая только из входных наборов (указания учителя о требуемых выходных сигналов отсутствуют). Необходимо отметить, что весовые коэффициенты связей внутри соревновательного слоя, вычисленные по соотношению 4.7, вообще не изменяются, а для качественного обучения обучающую последовательность предъявляют сети не один раз. При этом существуют различные критерии окончания обучения и методы изменения скорости обучения на различных итерациях обучения, а также способы исключения «мертвых» узлов (узлы, которые никогда не выигрывают и следовательно не обучаются) [25].

Так как в соответствии со свойствами соревновательного слоя только один нейрон имеет ненулевое выходное значение, то на каждом шаге обучения проводится изменение только одного весового коэффициента. Поэтому при моделировании обучения соревновательной сети можно по максимальной активации заранее определить победителя и применять соотношение 4.14 только к выигравшему нейрону.

#### **4.4. Подходы к формированию структуры НС и определению ее параметров**

Для решения конкретных возникающих задач может быть предложено большое количество разнообразных вариантов НС. Основные подходы к формированию структуры НС:

- 1) Использование нейросети как средства реализации известных традиционных алгоритмов с возможностью распараллеливания процесса решения. При данном подходе операции алгоритма представляются в нейросетевом логическом базисе (с учетом операций нейронов). При этом формулируется не начальная задача, а алгоритм ее решения и ставится задача построения нейронной сети, выполняющей заданные преобразования. Например, сколько решают статистик, столько и вариантов сетей, но с учетом того, что каждая включает в себя накопление сигнала рассогласования, нормировку и сравнение с порогом, то в пределах одной архитектуры нейронной сети в принципе могут быть охвачены разные алгоритмы. Основная задача – выбор параметров с учетом включения (не включения) априорной информации.
- 2) Формирование нейронной сети исходя из начальной постановки задачи и рассмотрение ее как общей задачи распознавания образов. Так называемый общий подход позволяет получить действительно параллельное решение. При этом не рассматриваются конкретные нейросетевые модели и алгоритмы обучения, они формируются исходя из задачи и требований, предъявляемых к качеству функционирования сети (сформированным критериям оптимальности) [26]. Основная задача такого подхода – формирование функционалов качества решения задачи, подлежащих дальнейшей оптимизации.
- 3) Анализ задачи и нахождение соответствующих ей нейросетевых моделей. В данном случае учитываются свойства каждой модели и выбирается наиболее подходящий комбинированный вариант. При этом так же возможно разделение исходной задачи на этапы и реализация отдельных этапов в виде нейронных сетей. В этом случае функциональная декомпозиция чаще всего осуществляется исходя из существующих нейросетевых моделей (в отличие от подхода 2, где разделение проводится исходя из свойств задачи).

Первый подход практически не дает никаких дополнительных преимуществ решению задачи, за исключением тех, которые обеспечиваются нейросетевой реализацией (см. п.п. 4.1 особенности нейровычислений), в частности распараллеливание процесса получения решения.

Второй подход, несмотря на универсальность, представляется достаточно сложным с точки зрения определения структуры сети и алгоритма ее обучения, представляющего собой в этом случае алгоритм оптимизации построенного функционала качества решения задачи.

Наибольшее применение получил третий подход. В дальнейшем будет рассматриваться только третий подход, как наиболее простой и в то же время эффективный Для его реализации необходимо знание о свойствах каждой из моделей НС, о том, в каком классе задач целесообразно их применение, а также какие алгоритмы обучения могут быть для них использованы. В связи с этим в рамках третьего подхода могут быть даны следующие рекомендации.

- Для решения задач, которые могут быть формализованы как задачи классификации, аппроксимации и прогнозирования, обычно используют сети прямого распространения сигнала. Обучение при этом чаще всего осуществляется по алгоритму обратного распространения ошибки или его модификациям [25].
- Задачи кластеризации (классификации без учителя) могут быть решены с применением соревновательных сетей. Обучение проводится с использованием алгоритмов соревнования.
- Для построения ассоциативной памяти, когда известны образы, подлежащие хранению в НС, а также для решения задач, формализация которых позволяет отыскивать решение с помощью нахождения минимума некоторого функционала качества, целесообразно применение нейронной сети Хопфилда. Обучение при этом должно проводиться с использованием правила Хебба. При поиске минимума функционала обучение состоит в вычислении значений весов по выражению, полученному в результате сопоставления функции энергии сети и минимизируемого функционала.

После того, как определена модель НС, необходимо также выделить информацию, на основе которой будет выполнять обучение, а затем и функционирование сети.

Нейронная сеть обучается путем установления связи между входами и выходами. Необходимо учесть по возможности все разнообразие входных данных, влияющих на получение желаемого результата и подготовить их в виде достаточного числа примеров, включающих значения входных данных и правильные ответы, если обучение будет проводиться с учителем.

Для построения сети прямого распространения сигналов помимо определения числа входных и выходных нейронов, соответствующих числу входных данных и ожидаемых ответов, необходимо определить число внутренних слоев и внутренних нейронов.

В настоящее время не существует законченной методики определения необходимого числа внутренних слоев и числа нейронов в каждом слое. Одно из эвристических правил для двухслойной сети заключается в использовании в качестве числа нейронов скрытого слоя среднего значения, полученного из числа входных и выходных нейронов с последующим уточнением на основе результатов обучения сети. Опасность наличия избыточного числа внутренних нейронов заключается в возможности перехода сети в режим запоминания примеров, вместо обучения с обобщением. Когда это происходит, сеть отлично функционирует при обучении, но плохо работает на примерах, которым не обучалась. В этом случае для получения приемлемого качества обучения (обобщения) потребуется больше примеров и еще больше времени. При недостаточном числе нейронов в скрытом слое не удается обучить НС распознаванию даже всех заданных обучающих примеров, так как она их «забывает».

Таким образом построение НС с использованием третьего похода состоит из следующих шагов, включающих определение всех составляющих модели нейронной сети (см. рис. 4.5):

1. Формулировка задачи. Определение входной информации для сети и основных ее функций.
2. Выбор способов представления информации для сети (кодирование, нормализация и т.п.) и подготовка ее в виде набора примеров (обучающей выборки).
3. Определение топологии нейронной сети. Выбор входов и выходов НС, а также моделей взаимодействия нейронов и преобразования входной информации. В результате должны быть получены следующие параметры НС:
  - $N_{\text{вх}}$  – число входных нейронов;
  - $N_{\text{вых}}$  – число выходных нейронов;
  - $n$  – число скрытых слоев;
  - $N_i$  – число нейронов в каждом  $i$ -ом слое ( $i=1..n$ );
  - вид функции активации " $a$ ";
  - вид передаточной (выходной) функции нейрона  $\varphi$ .
4. Задание критерия останова обучения НС и значения допустимой ошибки обучения, а также выбор алгоритма обучения и параметров обучения (скорости обучения  $\alpha$  и других, если необходимо) с целью получения значений весовых коэффициентов для всех нейронов сети или их вычисление по соответствующим математическим соотношениям. Проведение обучения НС.
5. Тестирование обученной НС с целью определения достаточности обучающей последовательности и нейронов в НС, а также оценки показателей качества функционирования НС.

## 4.5. Моделирование работы НС

Построение программной модели нейронной сети и проверка ее работоспособности включает в себя несколько этапов:

1 этап. Формирование обучающей и проверочной последовательностей. Создание нейронов или слоев НС (задание структуры и типа НС), их инициализация (задание параметров НС и начальных значений).

2 этап. Обучение НС на обучающей последовательности и вычисление показателей качества обучения.

3 этап. Проверка обученной НС на проверочной последовательности и вычисление показателей качества функционирования.

В случае обучения с учителем обучающая и проверочная последовательности имеют одинаковую структуру и состоят из примеров входных сигналов и соответствующих им выходных сигналов, в случае обучения без учителя в обучающей последовательности отсутствуют значения требуемых выходных сигналов.

Для создания программных моделей НС существуют различные библиотеки программ на разных языках, а также имеются инструментальные пакеты моделирования НС. Одной из удобных и эффективных сред проектирования НС является MATLAB и его библиотека функций для НС – Neural Network Toolbox (NNET) [27].

В состав библиотеки NNET входят:

- функции задания нового слоя и инициализации НС (для каждой нейросетевой парадигмы свои функции);
- функции обучения НС (для различных алгоритмов обучения);
- функции вычисления параметров сети на каждом шаге обучения для различных правил модификации весовых коэффициентов;
- функции вычисления и графического отображения изменения ошибок обучения;
- функции моделирования работы НС;
- вспомогательные функции.

Прежде чем рассматривать основные функции пакета, нужно описать используемые в них структуры данных.

### *4.5.1. Параметры, определяющие свойства НС (входы, слои, выходы и их взаимосвязи, целевые установки, весовые коэффициенты и смещения (пороги)).*

Для задания параметров НС в MATLAB-е предусмотрена структурная переменная net. В данном разделе представлено описание только основных полей этой структуры.

**net.numInputs** – число входов НС (число входных нейронов).

- net.inputs** – матрица ( $N_2 * 1$ ,  $N_2=net.numInputs$ ), элементы которой net.inputs{i} представляют собой структурные переменные и определяют свойства  $i$ -ого входа.
- net.inputs{i}** – содержит детальное описание каждого  $i$ -ого входа.
- net.inputs{i}.size** – размерность входного вектора у  $i$ -ого входного нейрона.
- net.inputs{i}.range** – матрица ( $R_i * 2$ ,  $R_i=net.inputs{i}.size$ ), каждый  $j$ -ая строка содержит минимальное и максимальное значения  $j$ -ого входа.
- net.numLayers** – число слоев НС, за исключением входного слоя.
- net.layers** – матрица ( $N_1 * 1$ ,  $N_1=net.numLayers$ ), элементы которой net.layers{i} – структурные переменные и определяют свойства  $i$ -ого слоя.
- net.layers{i}** – содержит детальное описание каждого  $i$ -ого слоя.
- net.layers{i}.size** – число нейронов в  $i$ -ом слое.
- net.layers{i}.initFcn** функция инициализации  $i$ -ого слоя.
- Возможные значения:
- initwb** – инициализация с вызовом функций, определенных для каждого слоя в отдельности (поля net.biases{i}.initFcn, net.inputWeights{i,j}.initFcn и net.layerWeights{i,j}.initFcn определяют функции инициализации для смещений, весов входов и слоев соответственно);
  - initnw** – инициализация по правилу Нгуена-Уидроу одинаковая для всех слоев .
- net.layers{i}.transferFcn** – передаточная функция нейронов  $i$ -ого слоя.
- Возможные варианты (см. рис. 4.6):
- hardlim** – шаговая несимметричная функция;
  - hardlims** – шаговая симметричная функция;
  - logsig** – сигмоидальная несимметричная функция;
  - tansig** – сигмоидальная симметрическая функция;
  - purelin** – линейная функция;
  - satlin** – линейная с зонами насыщения;
  - poslin** – линейная положительная без зон насыщения;
  - satlin** – линейная положительная с зонами насыщения;
  - tribas** – треугольная функция;
  - compet** – соревновательная функция;
  - и другие.
- net.biasConnect** – матрица ( $N_1 * 1$ ), определяющая наличие смещений у нейронов. Наличие (или отсутствие) смещения в  $i$  -ом слое нейронов отражается значением «1» (или «0») в net.biasConnect(i).
- net.biases** – матрица ( $N_1 * 1$ ), элементы которой net.biases { i } – структурные переменные и определяют свойства смещений нейронов  $i$ -ого слоя НС (или нулевая матрица, если в слое отсутствуют смещения).

**net.biases{i}.initFcn** – функция инициализации смещений нейронов  $i$ -ого слоя. Возможные варианты (если `net.layers{i}.initFcn=initwb`):  
**initzero** – функция присвоения нулевых начальных значений смещениям;  
**rands** – функция присвоения случайных начальных значений смещениям;  
и другие.

**net.biases{i}.learn** – параметр, определяющий будет ли  $i$ -ый вектор смещений изменяться в процессе обучения. Возможные значения «0» (изменяется) и «1» (не изменяется).

**net.biases{i}.learnFcn** – функция обучения для  $i$ -ого вектора смещений (для  $i$ -ого слоя нейронов). Вызывается на каждом шаге вычисления новых значений смещений в процессе обучения, если в качестве `net.trainFcn` задана функция `trainwb` (обучение с использованием функций обновления смещений, определенных в описаниях свойств смещений каждого слоя). Возможные варианты:

**learngd** – обучение с применением метода градиентного спуска;  
**learnwh** – обучение Уидроу-Хоффа (с применением дельта правила);  
и другие.

**net.inputConnect** – матрица ( $N_1 * N_2$ ) определяет на какие слои НС поступают сигналы с входного слоя. Значение «1» (или «0») в элементе матрицы `net.inputConnect(i,j)` показывает наличие (или отсутствие) связи  $j$ -ого входного нейрона с  $i$ -м слоем.

**net.inputWeights** – матрица ( $N_1 * N_2$ ), элементы которой `net.inputWeights{i,j}` – структурные переменные и определяют свойства весовых коэффициентов поступающих на  $i$ -й слой НС с  $j$ -ого входа НС (или нулевая матрица, если связи от  $j$ -ого входа к  $i$ -ому слою нет).

**net.inputWeights{i,j}.delays** – вектор положительных (или нулевых) значений, определяющих величину линии задержки между  $j$ -ым входом и  $i$ -ым слоем.

**net.inputWeights{i,j}.initFcn** – функция инициализации весовых коэффициентов, поступающих на  $i$ -й слой с  $j$ -ого входа.

Возможные варианты (если `net.layers{i}.initFcn=initwb`):

**initzero** – функция присвоения нулевых начальных значений весам;  
**rands** – функция присвоения случайных начальных значений весам;  
и другие.

**net.inputWeights{i,j}.learn** – параметр, определяющий будут ли весовые коэффициенты, поступающие на  $i$ -й слой с  $j$ -ого входа, изменяться в процессе обучения. Возможные значения «0» (изменяется) и «1» (не изменяется).

**net.inputWeights{i,j}.learnFcn** – функция обучения для весовых коэффициентов, связывающих  $i$ -й слой с  $j$ -ом входом.

Вызывается на каждом шаге вычисления новых значений весов в процессе обучения, если в качестве `net.trainFcn` задана функция `trainwb` (обучение с использованием функций обновления весов, определенных в описаниях свойств весовых коэффициентов каждого слоя). Возможные варианты:

`learngd` – обучение с применением метода градиентного спуска;

`learnh` – обучение с обновлением весов по правилу Хебба;

и другие.

**net.layerConnect** – матрица ( $N_1 * N_1$ ), показывающая наличие (отсутствие) связей с другими слоями. Значение “1” (или “0”) в элементе матрицы `net.layerConnect(i,j)` показывает наличие связи от  $j$ -ого к  $i$ -ому слою.

**net.layerWeights** – элементы этой матрицы ( $N_1 * N_1$ ) `net.layerWeights{i,j}` – структурные переменные и определяют свойства весовых коэффициентов связывающих  $i$ -й слой НС и  $j$ -й слой НС (или нулевая матрица, если связи от  $j$  ого к  $i$ -ому слою нет).

**net.layerWeights{i,j}.delays** – вектор положительных (или нулевых) значений, определяющих величину линии задержки сигналов, поступающих с  $j$ -ого слоя на  $i$ -ый слой.

**net.layerWeights{i,j}.initFcn** – функция инициализации весовых коэффициентов, поступающих на  $i$ -й слой с  $j$ -ого слоя.  
Вычисления начальных значений весовых коэффициентов в соответствии с заданными в этом поле функциями выполняются, если `net.layers{i}.initFnc=initwb` (инициализация с вызовом функций, определенных для каждого слоя в отдельности).

Возможные варианты:

`initzero` – функция присвоения нулевых начальных значений;

`rands` – функция присвоения случайных начальных значений;

и другие.

**net.layerWeights{i,j}.learn** – параметр, определяющий будут ли весовые коэффициенты, поступающие на  $i$ -й слой с  $j$ -ого слоя, изменяться в процессе обучения. Возможные значения «0» (изменяется) и «1» (не изменяется).

**net.layerWeights{i,j}.learnFcn** – функция обучения для весовых коэффициентов, связывающих  $i$ -й слой с  $j$ -м слоем.

Вызывается на каждом шаге вычисления новых значений весов в процессе обучения, если в качестве `net.trainFcn` задана функция `trainwb` (обучение с использованием функций обновления весов, определенных в описаниях свойств весовых коэффициентов каждого слоя). Возможные варианты такие же, как и для `net.inputWeights{i,j}.learnFcn`.

**net.outputConnect** – матрица ( $1 * N_1$ ), показывающая какие слои выдают выходные значения НС. Если  $i$ -й слой участвует (или не

- участвует) в выдаче выходного сигнала, то в элементе матрицы ***net.outputConnect(i)*** должна стоять «1» (или «0»).
- net.outputs** – элементы этой матрицы ( $N_1 \times 1$ ) ***net.outputs{i}*** – структурные переменные и определяют свойства выходов *i*-ого слоя НС (или нулевая матрица, если в слое отсутствуют элементы, участвующие в формировании выходного сигнала НС).
- net.targetConnect** – матрица ( $1 \times N_1$ ), показывающая для каких слоев заданы целевые установки. Если для *i*-ого слоя существует (или не существует) указание учителя, то в элементе матрицы ***net.targetConnect(i)*** должна стоять «1» (или «0»).
- net.targets** – элементы этой матрицы ( $N_1 \times 1$ ) ***net.targets{i}*** – структурные переменные и определяют свойства целевых установок, связанных с *i*-м слоем НС (или нулевая матрица, если в слое отсутствуют элементы, для которых определены указания учителя).

#### *4.5.2. Параметры, определяющие режимы работы функций и их аргументы*

- net.adaptFcn** – название используемого правила подстройки (в библиотеке MATLAB NNET предусмотрено правило подстройки смещений и весовых коэффициентов – ***adaptwb***).
- net.initFcn** – используемая функция инициализации НС (в библиотеке MATLAB NNET предусмотрена функция последовательной инициализации слоя за слоем - ***initlay***).
- net.performFcn** – имя используемого показателя оценки работы НС. Некоторые возможные значения:
- mae*** – средняя абсолютная ошибка;
  - mse*** – среднеквадратическая ошибка;
  - sse*** – суммарная среднеквадратическая ошибка;
  - и другие.
- net.trainFcn** – функции обучения НС. Некоторые возможные значения:
- trainbfg*** – обратное распространение ошибки на основе квазиньютоновского метода (метод Бройдена, Флетчера, Гольдфарба и Шенно);
  - traincgf*** – обратное распространение ошибки на основе метода сопряженных градиентов Флетчера-Паузлла;
  - traingd*** – обратное распространение ошибки на основе метода градиентного спуска;
  - traingdm*** – обратное распространение ошибки с моментами на основе метода градиентного спуска;
  - trainwb*** – обучение в соответствии с функциями, заданными для каждого слоя в отдельности (параметры полей ***net.inputWeights{i,j}.learnFcn*** и ***net.layerWeights{i,j}.learnFcn***);

**trainlm** – обратное распространение ошибки по методу Левенберга–Марквардта;

и другие.

**net.adaptParam** – параметры и значения для соответствующей функции подстройки (**net.adaptFcn**).

**net.initParam** – параметры и значения для соответствующей функции инициализации (**net.initFcn**).

**net.performParam** – параметры и значения для соответствующей функции оценки работы НС (**net.performFcn**).

**net.trainParam** – параметры и значения для соответствующей функции обучения (**net.trainFcn**).

#### *4.5.3. Переменные для хранения значений весовых коэффициентов и смещений НС*

**net.IW** – матрица весовых коэффициентов ( $N_1 * N_2$ ), элемент **net.IW{i,j}** – матрица, в которой хранятся значения веса по связи от  $j$ -ого входа к  $i$ -ому слою.

**net.LW** – матрица весовых коэффициентов ( $N_1 * N_1$ ), элемент **net.LW{i,j}** содержит матрицу весов по связи от  $j$ -ого слоя к  $i$ -ому слою.

**net.b** – вектор смещений ( $N_1 * 1$ ), элемент **net.b{i}** содержит вектор смещений для  $i$ -ого слоя.

#### *4.5.4. Основные функции моделирования НС*

В данном разделе рассматриваются основные функции для моделирования НС прямого распространения сигналов – функции создания слоев (**newff**, **newfftd**), функция обучения (**train**), функция моделирования (**sim**) и функция задания начального состояния сети (**init**). Однако, функции **init**, **sim**, **train** являются общими и могут быть использованы для инициализации, обучения и моделирования НС любого типа. Естественно НС должна быть предварительно создана (а следовательно и определена структурная переменная **net**, в которой хранится информация о параметрах сети) с помощью соответствующих функций создания слоев определенного типа.

При описании основных функций вызываемые внутри них другие функции здесь рассматриваются как параметры и детально не описываются (хотя они и могут быть использованы отдельно вне рассматриваемых функций).

##### init

Предназначена для инициализации НС (задания начальных параметров или возвращения НС в начальное состояние)

Вызов функции:

**net = init(net)**

В результате выполнения функция возвращает НС с весовыми коэффициентами и смещениями вычисленными в соответствии с функцией, заданной в **net.initFcn** и с параметрами, заданными в **net.initParam**.

### **newff**

Эта функция служит для создания сети с прямым распространением сигналов.

Вызов функции:

**net = newff(PR,[S1 S2...SN1],{TF1 TF2...TFN1},BTF,BLF,PF)**

Параметры функции:

**PR** – матрица (**R \* 2**) минимальных и максимальных значений для **R** входных элементов;

**Si** – число нейронов в **i**-ом слое, (**i=1... N1**; число слоев – **N1**);

**TFi** – передаточная функция нейронов **i**-ого слоя, по умолчанию – '**tansig**';

**BTF** – функция обучения НС, по умолчанию – '**traingdx**';

**BLF** – правило модификации весовых коэффициентов и смещений, по умолчанию – '**learngdm**';

**PF** – функция оценки работы НС, по умолчанию – '**mse**'.

В результате выполнения функции формируется входной слой с **R** нейронами и **N1** слоев НС, в каждом **Si** нейронов (**i=1...N1**). Последний слой является выходным слоем.

### **Пример 4.1.**

Пусть имеется набор входных значений, представленных вектором **P** и соответствующие им выходные значения представленные вектором **T**.

Необходимо построить НС, выдающую в ответ на входной сигнал соответствующий выходной сигнал.

**P = [0 1 2 3 4 5 6 7 8 9 10]; % задание вектора входных сигналов**

**T = [0 1 2 3 4 3 2 1 2 3 4]; % задание вектора целей (выходных сигналов)**

Создадим двухслойную НС с прямым распространением сигнала: в первом слое – пять нейронов с симметричной сигмоидальной передаточной функцией и во втором – один нейрон с линейной передаточной функцией. Диапазон входных значений лежит в интервале от 0 до 10. Для обучения НС будет использоваться функция **train** (с последующим вызовом функции **trainlm**, которая установлена по умолчанию в качестве параметра **trainFcn**).

**net = newff([0 10],[5 1],{'tansig' 'purelin'}); %создание двух слоев с прямым %распространением сигнала**

Можно посмотреть характеристики полученной НС:  
`disp(net)` %вызов функции отображающей переменную net  
В результате будет получена следующая информация о НС:

```
numInputs: 1
numLayers: 2
biasConnect: [1; 1]
inputConnect: [1; 0]
layerConnect: [0 0; 1 0]
outputConnect: [0 1]
targetConnect: [0 1]

numOutputs: 1 (read-only)
numTargets: 1 (read-only)
numInputDelays: 0 (read-only)
numLayerDelays: 0 (read-only)

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
targets: {1x2 cell} containing 1 target
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

adaptFcn: 'adaptwb'
initFcn: 'initlay'
performFcn: 'mse'
trainFcn: 'trainlm'

adaptParam: .passes
initParam: (none)
performParam: (none)
trainParam: .epochs, .goal, .max_fail, .mem_reduc,
            .min_grad, .mu, .mu_dec, .mu_inc,
            .mu_max, .show, .time

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors
```

Кроме этого дополнительную информацию о созданной НС можно получить с помощью функции `display(net)`.

Моделирование НС:

```
Y = sim(net,P); % вызов функции моделирования работы НС  
plot(P,T,P,Y,'o') % графическое представление начальных условий
```

```
net.trainParam.epochs = 50; % задание числа проходов обучения
```

```
net = train(net,P,T); % вызов функции обучения
```

```
Y = sim(net,P); % вызов функции моделирования работы НС
```

```
plot(P,T,P,Y,'o') % графическое представление результата обучения
```

### newfftd

Эта функция служит для создания сети с прямым распространением сигналов с задержками на входных сигналов.

Вызов функции:

```
net = newfftd(PR,IDL,[S1 S2...SN1],{TF1 TF2...TFN1},BTF,BLF,PF)
```

Параметры функции этой функции совпадают с параметрами функции **newff**, за исключением нового параметра **ID** – вектор входной задержки.

В результате выполнения функции получается **N1** слоев НС с линией задержки от входа до первого слоя.

### *Пример 4.2*

Пусть имеется набор входных значений, представленных вектором **P** и соответствующие им выходные значения представленные вектором **T**.

Необходимо построить НС, выдающую в ответ на входной сигнал соответствующий выходной сигнал.

Задача состоит в создании ответной реакции, представленной вектором **T**, на входное воздействие, представленное вектором **P**. Задача может быть решена с использованием НС прямого распространения сигнала с одиночной задержкой на входе.

```
P = {1 0 0 1 1 0 1 0 0 0 0 1 1 0 0 1};  
T = {1 -1 0 1 0 -1 1 -1 0 0 0 1 0 -1 0 1};
```

Создадим двухслойную НС с прямым распространением сигнала: в первом слое – пять нейронов с симметричной сигмоидальной передаточной функцией и во втором- один нейрон с линейной передаточной функцией. Диапазон входных значений лежит в интервале от 0 до 10. Входной сигнал поступает на нейроны первого слоя (значение задержки равно 0) и на блок единичной задержки (значение задержки равно 1), с которого сигнал поступает на нейроны первого слоя (см. рис. 4.9).

```
net = newfftd([0 1],[0 1],[5 1],{'tansig' 'purelin'});  
net.trainParam.epochs = 50;  
net = train(net,P,T);  
Y = sim(net,P)
```

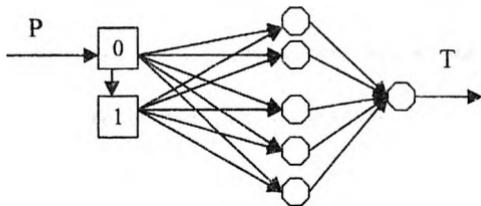


Рис. 4.9. Пример моделирования НС

### sim

Предназначена для моделирования работы НС.

Форматы вызова:

[Y,Pk,Ak] = sim(net,Pn,Ah)

[Y,Pk,Ak] = sim(net,{Q TS},Pn,Ah)

Параметры функции:

**P** – входные сигналы НС – матрица ( $N_i * TS$ ), каждый элемент которой  $P\{i,ts\}$  – матрица ( $R_i * Q$ );

**Pn** – начальные значения вектора входных задержек (по умолчанию установлены нулевые значения) – матрица ( $N_i * ID$ ), каждый элемент которой  $Pn\{i,k\}$  – матрица ( $R_i * Q$ );

**Ah** – начальные значения вектора задержек между слоями – матрица ( $N_i * LD$ ), каждый элемент которой  $Ah\{i,k\}$  – матрица ( $S_i * Q$ ) (по умолчанию установлены нулевые значения).

Возвращаемые значения:

**Y** – Выходные сигналы НС – матрица ( $No * TS$ ), каждый элемент которой  $Y\{i,ts\}$  – матрица ( $U_i * Q$ );

**Pk** – Полученные (конечные) значения вектора входных задержек – матрица ( $N_i * ID$ ), **каждый элемент которой  $Pk\{i,k\}$**  – матрица ( $R_i * Q$ );

**Ak** – Полученные (конечные) значения вектора задержек между слоями – матрица ( $N_i * LD$ ), **каждый элемент которой  $Ak\{i,k\}$**  – матрица ( $S_i * Q$ ).

**Ni** = net.numInputs

**Nl** = net.numLayers

**No** = net.numOutputs

**ID** = net.numInputDelays (глубина задержки по входу)

**LD** = net.numLayerDelays (глубина задержки между слоями)

**Ri** = net.inputs{i}.size

**Si** = net.layers{i}.size

**Ui** = net.outputs{i}.size

**TS** – число временных отсчетов, в течение которых проводится моделирование.

**Q** – число обучающих последовательностей входных сигналов

Параметры **Rn**, **An**, **Rk**, **Ak** – необязательные и используются, только если в НС необходимы задержки.

Столбцы матриц **Rn**, **An**, **Rk**, **Ak** организованы таким образом, что вначале следуют сигналы с самыми большими задержками, а затем – с самыми маленькими:

**Rn{i,k}** – i-ый входной сигнал в момент времени  $ts=k-ID$ .

**Rk{i,k}** – i-ый входной сигнал в момент времени  $ts=TS+k-ID$ .

**An{i,k}** – выходной сигнал i-ого слоя в момент времени  $ts=k-LD$ .

**Ak{i,k}** – выходной сигнал i-ого слоя в момент времени  $ts=TS+k-LD$ .

Вызов функции вида

**[Y,Pf,Af] = sim(net,{Q TS},Rn,An)**

используется в случае, если НС не имеет входных сигналов (как НС Хопфилда, у которой задаются только начальные условия функционирования).

### train

Функция для обучения НС.

Вызов функции:

**[net,tr] = train(net,P,T,Rn,An)**

Назначение параметров функции совпадает с одноименными параметрами функции **sim** и дополнительный параметр **T** – желаемые (целевые) значения выходных сигналов (по умолчанию установлены нулевые значения) – матрица ( $Nt * TS$ ,  $Nt = net.numTargets$ ), каждый элемент которой  $T{i,ts}$  – матрица ( $Vi * Q$ ,  $Vi = net.targets{i}.size$ ).

Возвращаемые функцией параметры:

**net** – обученная НС;

**tr** – шаги и характеристики процесса обучения.

Обязательными параметрами при вызове функции являются только **net** и **P**. При этом осуществляется вызов функции обучения, заданной в качестве параметра в поле **trainFcn**.

## 4.6. Диагностика многоканальной информационно-измерительной системы с использованием нейронной сети

Рассмотрим как можно решить задачу функциональной диагностики многоканальной информационно-измерительной системы с использованием НС.

Основной особенностью рассматриваемого объекта диагностирования (ОД) является наличие разнообразной избыточности (структурной, функциональной, временной, аналитической), позволяющей сформировать пространство контрольных соотношений (алгебраических инвариантов (АИ), являющихся стационарными случайными последовательностями (СП), в отличие от самих измеряемых сигналов) для принятия решения о наличии дефекта в ОД. Под дефектом в данном случае подразумевается увеличение уровня систематической или случайной составляющей погрешности измерений информационных каналов (ИК).

С учетом того, что задача обнаружения дефекта в этом случае формализуется как задача обнаружения изменения статистических свойств (разладки) СП, представленной последовательностью АИ, можно разработать нейронную сеть для обнаружения изменения свойств СП и использовать ее для принятия решения по каждому из АИ. Для построения НС необходимо выполнить рассмотренную выше последовательность шагов.

1. Пусть СП – процесс авторегрессии  $p$ -ого порядка AR( $p$ ) с математическим ожиданием  $m_i$ , и дисперсией  $\sigma_i$  ( $i=1$  до возникновения дефекта и  $i=2$  после возникновения разладки). Необходимо определить момент изменения свойств СП. Входной информацией для разрабатываемой НС будет наблюдаемая реализация СП. Результатом работы НС должен быть сигнал о наличии или отсутствии разладки.
2. Значения каждого из отсчетов СП представляют собой числа и их можно сразу использовать в качестве входной информации. При большом разбросе значений СП целесообразно выполнить нормировку либо относительно максимального значения, или в соответствии со следующим выражением [24]:

$$X_i = 2 * (x_i - \min) / (\max - \min) - 1,$$

где  $x_i$  – исходное значение,

$X_i$  – получаемый нормированный сигнал,

$\min$  и  $\max$  – соответственно минимальное и максимальное значения интервала входных параметров, подаваемых на  $i$ -ый нейрон.

Обучающая выборка будет представлять собой последовательности реализаций СП из  $N$  элементов (из них  $N_{\text{без разладки}}$  и  $N_1$  элементов с разладкой) и последовательности требуемых значений выходных сигналов, соответствующих заданной последовательности входных сигналов.

3. Так как рассматриваемая задача является задачей распознавания образов (образ 1 – СП без разладки, образ 2 – СП с разладкой), то для ее решения можно использовать простейшую модель НС прямого распространения сигнала со слоистой структурой с одним скрытым слоем (см. рис. 4.10), с нейронами с одинаковыми аддитивными функциями активации (см. соотношение 4.1) и сигмоидальными передаточными функциями. Число входов НС  $N_{\text{вх}}$  соответствует числу отсчетов (значений СП в предыдущие по отношению к рассматриваемому моменты времени), по которым будет

приниматься решение и может быть от одного до допустимого времени запаздывания в обнаружении разладки (если входов больше одного, то каждое значение СП проходит через блоки задержки для образования «временного» окна наблюдения), а  $N_{\text{вых}}=1$  (так как одного нейрона достаточно для разделения двух классов: значение на выходе равное 0 - отсутствие разладки, 1 – есть разладка). Начальное значение числа нейронов в скрытом слое можно взять равным  $(N_{\text{вх}}+1)/2$ , а затем уточнить на основе моделирования.

4. Обучение НС целесообразно проводить с учителем на основе алгоритма обратного распространения ошибки. Критерием останова обучения НС будет достижение заданного уровня ошибки обучения по всей обучающей последовательности. Скорость обучения  $\alpha=0.1-0.3$ . В процессе обучения нужно отслеживать скорость изменения ошибки обучения и при необходимости изменять начальные условия (начальные весовые коэффициенты, которые задаются случайным образом) или скорость обучения, или увеличивать число нейронов в скрытом слое.

Выходной сигнал сети должен быть максимально приближен к выходному требуемому, но учитывая, что выбранные функции преобразования нейронов могут не позволять формировать на выходе 0 или 1, а поэтому потребуется введение некоторого интервала (порога), в пределах которого значение будет отнесено к 1 или 0.

5. Для тестирования обученной НС необходимо создать проверочную последовательность. Показателями качества функционирования НС будут вероятность ложного срабатывания и вероятность обнаружения разладки[6]. Если полученные показатели не удовлетворяют требованиям, то необходимо в зависимости от вида процесса обучения изменить параметры сети или увеличить обучающую последовательность.

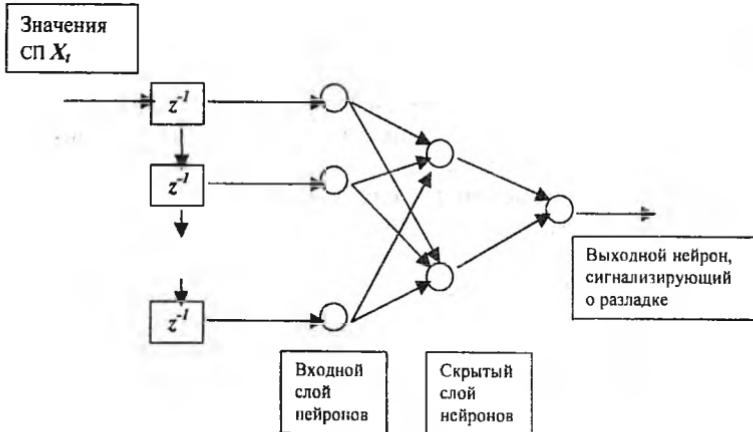


Рис.4.10. Нейронная сеть для обнаружения разладки

При таком подходе комплексирование частных решений по всем АИ в многоуровневой системе принятия решений АИ – параметр -ИК. для получения заключения о наличии дефекта параметра, а затем и канала, может быть выполнено как с использованием НС, так и без них.

Пусть кроме обнаружения момента возникновения дефекта для трехканального ОД требуется локализовать дефект (определить в каком канале присутствует дефект) и определить тип возникшего дефекта (что произошло изменение математического ожидания или изменение дисперсии). Для решения указанной задачи могут быть предложены различные структуры НС, некоторые представлены на рис. 4.11-4.13. На вход у любой из них подаются значения АИ.

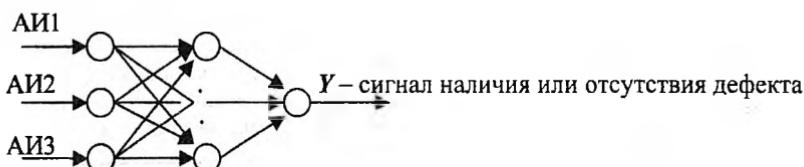


Рис. 4.11. Нейронная сеть для решения задач диагностирования:  
обнаружение дефекта в трехканальной системе по срезу АИ

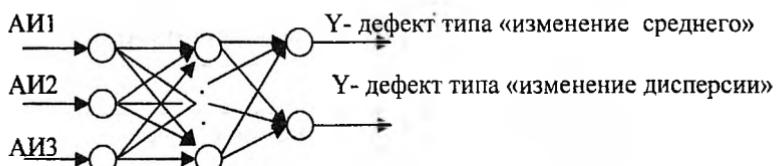


Рис. 4.12. Нейронная сеть для решения задач диагностирования:  
обнаружение дефекта и определение его типа

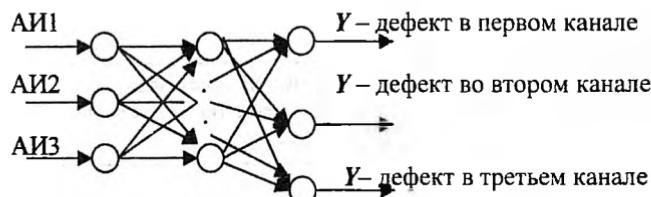


Рис. 4.13. Нейронная сеть для решения задач диагностирования:  
обнаружение и локализация дефекта

При этом во всех структурах НС рис. 4.11-4.13 учитываются взаимосвязи между АИ и решение формируется параллельно. Одновременно может быть предложена структура с последовательным вариантом принятия решений: вначале по каждому АИ в соответствии со схемой рис. 4.10, а затем в соответствии со схемой рис. 4.11. В этом случае структуру НС целесообразно сделать неполно связной (рис. 4.14).

Кроме оценки основных показателей качества работы НС (вероятности ложного срабатывания (оценивается при нормальном функционировании) и вероятности правильного решения (оценивается при функционировании с дефектом)) целесообразно отслеживать достигнутое значение ошибки обучения, а также количество проходов обучения и время обучения.

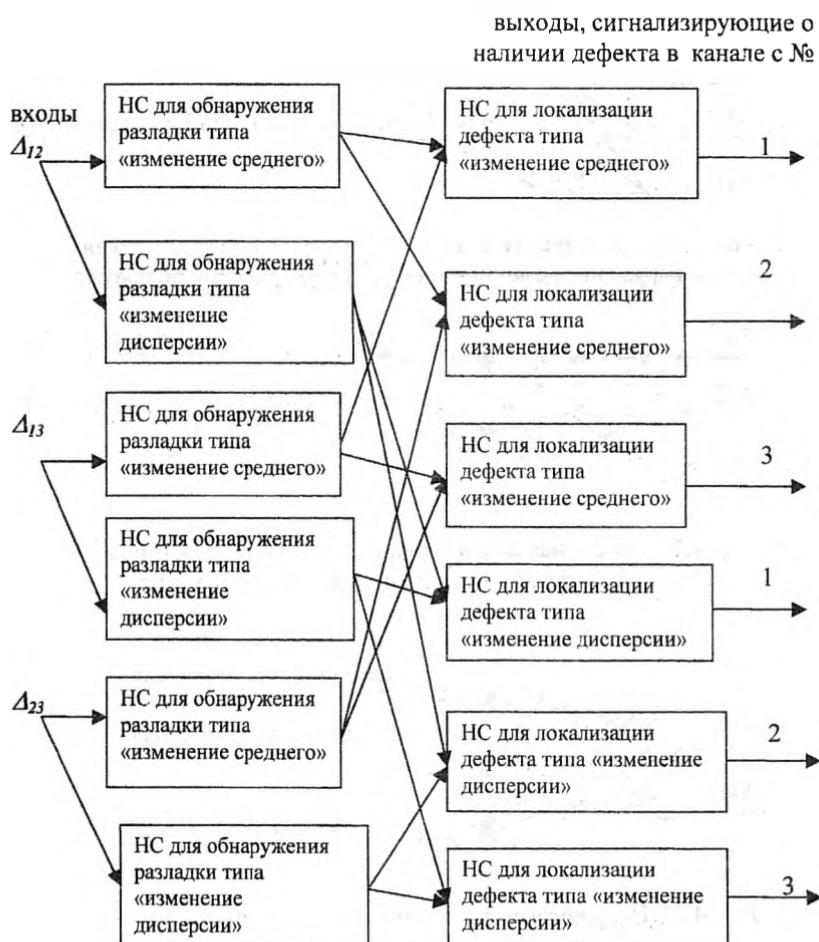


Рис. 4.14 Нейронная сеть для обнаружения и локализации дефектов

## 5. Содержание программы работ по изучению методов функционального диагностирования

### 5.1. Моделирование случайных факторов с использованием пакета MATLAB

#### 5.1.1. Генерация и анализ случайных величин

Для генерации случайных чисел в MATLAB имеется функция **rand**, причем результат ее выполнения зависит от типа и количества аргументов при обращении к ней.

1. Если **n** – целое число, **rand(n)** генерирует квадратную матрицу случайных чисел размером **n x n**, **rand(m, n)** создает **m x n** матрицу случайных чисел.
2. Если **B** – матрица, то **rand(size(B))** порождает матрицу такого же размера, что и **B**.
3. Если множество аргументов пусто, то обращение к **rand** без аргументов генерирует одно случайное число, которое изменяется при последующих вызовах.

По умолчанию генератор установлен на генерацию равномерно распределенных случайных чисел в интервале (0,1).

Установка **randn** переводит функцию **rand** на генерацию случайных чисел, распределенных по нормальному закону с нулевым средним и дисперсией равной единице.

4. Можно изменить порождающую константу генератора случайных чисел; выполнение **rand('state', n)** устанавливает ее равной **n**, **rand('state',0)** – возвращает установку генератора в начальное состояние.
5. Изменение числовых параметров распределения достигается с помощью линейного преобразования случайной последовательности. Для получения **n** равномерно распределенных чисел в интервале **[a, b]** необходимо выполнить:

$$x = a + (b - a) . * \text{rand}(n,1).$$

Для получения **n** случайных чисел с нормальным законом распределения при среднем значении **m** и дисперсии **D** нужно выполнить:

$$x = m + \text{sqrt}(D) . * \text{randn}(n,1) .$$

## Основные функции и команды для анализа последовательности случайных величин

1. **mean( x )** – вычисляет оценку среднего значения случайной величины, заданной вектором  $x$ . Если  $x$  – матрица, то **mean(x)** выдает средние значения по столбцам и возвращает вектор-строку, длина которой равна количеству столбцов матрицы.

2. **median(x)** – вычисляет значение серединного элемента случайной величины, заданной вектором  $x$ . Если  $x$  – матрица, то **median(x)** выдает значение срединных элементов по столбцам и возвращает вектор-строку, длина которой равна количеству столбцов матрицы.

3. **std( x )** – вычисляет стандартное отклонение, (девиацию) случайной величины, представленной вектором или матрицей. Если  $x$  – матрица, то девиация вычисляется для каждого столбца и результат имеет вид вектора-строки.

4. Команда **hist( x, n )** – строит гистограмму случайной величины  $x$ , где  $n$  – число градаций в общем объеме выборки.

5. **cov ( x )** – вычисляет дисперсию вектора  $x$ . Если  $x$  – матрица, размера  $n \times n$ , где каждый столбец соответствует переменной случайной величине, а каждая строка – наблюдению, то  $y = cov( x )$  будет матрицей ковариаций размером  $n \times n$ , диагональные элементы которой равны величине дисперсии для соответствующих столбцов матрицы  $x$ . Элементы матрицы  $x$  вычисляются по соответствующей формуле:

$$x_{ij} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_i)(x_j - \bar{x}_j),$$

где  $i, j = 1, 2, \dots, n$ ,

$\bar{x}_i, \bar{x}_j$  – средние значения по столбцам  $i$  и  $j$ .

Функция **cov ( x, y )**, где массивы  $x$  и  $y$  имеют одинаковое количество строк, равносильна функции **cov ([x y])**.

6. **corrcoef(x)** – вычисляет корреляционную матрицу для двумерного массива  $x$ , где каждый столбец рассматривается как переменная, а каждая строка как наблюдение. Элементы матрицы  $s = corrcoef(x)$  связаны с элементами матрицы  $c = cov(x)$  следующим соотношением:

$$s(i, j) = c(i, j) / \sqrt{D_i D_j},$$

Функция  $s = corrcoef(x)$ , где массивы  $x$  и  $y$  имеют одинаковое количество строк, равносильна функции **corrcoef ([x y])**.

6. Функция  $\text{erf}(x)$  определяется следующим образом:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

Программа моделирования последовательности независимых случайных величин приведена в приложении 5.1.

### 5.1.2. Генерация и анализ случайных векторов

Пусть задан вектор  $x$ ,  $x \in R^n$ . Полное описание задается совместной плотностью распределения  $f(x) = f(x_1, x_2, \dots, x_n)$ . В дальнейшем будем рассматривать генерацию вектора  $x$  в рамках корреляционной теории.

Обычно с этой позиции моделируется вектор, имеющий многомерное нормальное распределение  $f(x)$ :

$$f(x) = (2\pi)^{1/2} (\det M)^{1/2} \exp [-1/2 (x - m_x)^T M^{-1} (x - m_x)], \quad (5.1)$$

где  $m_x$  – математическое ожидание вектора  $x$ ,

$M = [m_{ij}]$  – корреляционная матрица,

$M = M [(x - m_x)(x - m_x)^T]$ .

В общем случае  $M \neq E$ , где  $E$  – единичная матрица.

Распределение (5.1) моделируется с помощью линейного преобразования:

$$X = A y + m_x, \quad y \sim N(O, E), \quad (5.2)$$

Матрица  $A = [a_{ij}]$  порядка  $n$  определяется условием:

$$M = A A^T, \quad (5.3)$$

где  $A$  – нижняя треугольная матрица.

Общий принцип моделирования распределения (5.1) сводится к следующему:

1. В соответствии с (5.3) вычисляется матрица  $A$ .
2. Генерируется вектор  $y$  с распределением  $N(O, E)$ .
3. Проводится линейное преобразование вектора  $y$  в вектор  $x$  в соответствии с (5.2).

Определение  $A$  в MATLAB проводится на основе разложения Холецкого:

$$M = A A^T,$$

справедливого для симметричных положительно определенных матриц. Оно выполняется с помощью функции  $R = \text{chol}(r)$ , дающей значение верхнетреугольной матрицы  $A$ .

Программа моделирования вектора с заданной корреляционной матрицей приведена в приложении 5.2.

### 5.1.3. Генерация и анализ случайных процессов

В дальнейшем пойдет речь о последовательности коррелированных случайных величин, представляющих собой стационарные временные ряды, задаваемые следующим уравнением:

$$x(n) = m_x + \sum_{i=1}^p a_i [x(n-i) - m_x] + \beta z(n), \quad (5.4)$$

где  $x(n) \in N(m_x, \sigma^2)$ ,  $z(n) \in N(0, 1)$ .

Модель типа (5.4) принято называть уравнением авторегрессии  $AP(p)$ , где  $p$  – порядок авторегрессии, а коэффициенты  $m_x$ ,  $a_i$ ,  $i = \overline{1, p}$ ,  $\beta$  являются параметрами модели [9].

#### *Основные типовые модели и их параметры*

Модель  $AP(0)$ :

$$x(n) = m_x + \sigma_x z(n), \quad (5.5)$$

соответствует случаю независимой последовательности.

Модель  $AP(1)$ :

$$\begin{aligned} x(n) &= m_x + a_1 [x(n-1) - m_x] + \beta z(n), \\ \text{где } a_1 &= \rho_1, \quad \rho_1 = \frac{M \{ (x_1 - m_x)(x_{1+1} - m_x) \}}{\sigma_x^2}, \\ \beta &= \sigma_x \sqrt{1 - a_1^2}, \end{aligned} \quad (5.6)$$

Процесс стационарен, если  $|a_1| < 1$ .

Модель  $AP(2)$ :

$$x(n) = m_x + a_1 [x(n-1) - m_x] + a_2 [x(n-2) - m_x] + \beta z(n), \quad (5.7)$$

$$\text{где } a_1 = \frac{\rho_1 (1 - \rho_2)}{1 - \rho_1^2}, \quad a_2 = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}; \quad (5.8)$$

$$\beta^2 = \sigma_x^2 \frac{(1 + a_2) \{ (1 - a_2)^2 - a_1^2 \}}{1 - a_2}; \quad (5.9)$$

$$\rho_k = \frac{M \{ (x_1 - m_x)(x_{1+k} - m_x) \}}{\sigma_x^2}, \quad k = 1, 2, \dots$$

Процесс стационарен, если:

$$\begin{aligned} a_1 + a_2 &< 1, \\ a_2 - a_1 &< 1, \\ -1 &< a_2 < 1. \end{aligned} \quad (5.10)$$

Подбирая параметры  $a_1$  и  $a_2$  можно в соответствии с (5.6) и (5.7) моделировать процессы с разными автокорреляционными функциями:

$$\begin{aligned} k_x(\tau) &= \sigma^2 e^{-\alpha/\tau}; \\ k_x(\tau) &= \sigma^2 e^{-\alpha/\tau} \cos(\beta \tau) \text{ и др.} \end{aligned}$$

Вычисление корреляционных функций случайных процессов в MATLAB производится с помощью функции `covf`, имеющей формат:

$$R = covf(z, M)$$

где  $z$  – матрица данных размером  $N \times n_z$ ,

$N$  – количество точек наблюдения,

$n_z$  – количество процессов, а целое число  $M$  указывает верхнюю границу изменения параметра корреляционных функций по отношению к интервалу наблюдений.

Результирующая матрица  $R$  имеет размерность  $n_z^2 \times M$ . Ее строка с номером  $i + (n_z - 1)j$  содержит взаимную корреляционную функцию столбцов  $i \neq j$  матрицы  $z$ , а при  $i = j$  автокорреляционную функцию столбцов матрицы  $z$ .

Программа моделирования случайного процесса, представленного моделью авторегрессии 1-го порядка  $AP(1)$ , приведена в приложении 5.3.

### Программа работы

1. Ознакомиться с содержанием алгоритма моделирования нормально-распределенных чисел и с основными операторами для анализа их характеристик (математического ожидания, дисперсии, закона распределения).

Выяснить зависимость оценок от объема выборки ( $n = 10, 20, 50, 100, 1000$ ) при доверительном уровне  $\alpha = 0,9; 0,95$ .

2. Ознакомиться с содержанием алгоритма генерации вектора нормально-распределенных чисел при размерности  $n = 2$  при  $m_x = 0$  для следующих значений корреляционных матриц:

$$R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \quad R = \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \end{pmatrix};$$

$$R = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}; \quad R = \begin{pmatrix} 1 & -0.8 \\ -0.8 & 1 \end{pmatrix}.$$

Построить оценки заданной корреляционной матрицы и выяснить зависимости оценок от объема выборки.

Построить доверительные интервалы для оценки коэффициента корреляции в зависимости от объема выборки  $n = 10, 20, 50, 100$ , при доверительном уровне  $\alpha = 0.9; 0.95$  [28].

3. Ознакомиться с содержанием алгоритма моделирования случайного процесса, представленного уравнением авторегрессии 1-го порядка.

Построить оценку параметра  $a_1$  для случаев  $a_1 = 0.1; 0.8$ .

Смоделировать процесс, заданный уравнением авторегрессии 2-го порядка при условии его стационарности и провести оценку параметров  $a_1, a_2$ .

## 5.2. Построение структурных инвариантов

Далее рассматривается два наиболее распространенных принципа построения структурных инвариантов (СИ):

1. СИ представляет собой разность одноименных сигналов.
2. СИ сформирован с использованием обобщенной статистики в виде оценки среднего по массиву значений однородных сигналов.

Общая методология проведения сравнительного анализа различных способов построения СИ основана на следующих предпосылках:

1. Любой выходной сигнал  $i$ -го канала представлен аддитивной моделью:

$$y_i(t) = \tilde{y}_i(t) + \xi_i(t), \quad i = \overline{1, 3},$$

где  $\tilde{y}_i(t)$  – полезная составляющая сигнала,  $\xi_i(t)$  – случайная составляющая, представляющая в дискретном времени независимую последовательность, имеющую нормальный закон распределения с параметрами  $\mu_i = \mu_l, \sigma_i = \sigma_l$ , где  $l = \overline{1, 2}; l = 1$  соответствует нормальному (правильному) режиму (НР) функционирования ОД, а  $l = 2$  – режиму при наличии дефекта (ФД).

2. Модель дефекта (Д) в любом из информационных каналов (ИК) отображается моделью скачкообразного изменения параметров  $\mu_l, \sigma_l$ , что соответствует на уровне структурных инвариантов разладке типа  $k$  или  $u$ , где

$$k = \frac{\mu_2 - \mu_1}{\sigma_1}; \quad u = \frac{\sigma_2}{\sigma_1}. \quad (5.11)$$

3. Все информационные каналы в случае двухканальной или трехканальной структурной избыточности в режиме НФ предполагаются однородными по статистическим характеристикам и являются однопараметрическими.

4. Принятие решения о наличии дефекта (разладки) производится на основе допускового алгоритма.

Проведение сравнительного анализа производится на основе характеристики обнаружения [29], позволяющей объединить два режима: режим НФ и режим ФД. Характеристика обнаружения может быть представлена одной из зависимостей:

$$\begin{aligned} P_{\text{обн}} &= F_1(P_{\text{ло}}, d), \\ \bar{N}_{\text{обн}} &= F_2(P_{\text{ло}}, d), \end{aligned} \quad (5.12)$$

где  $P_{\text{обн}}$  – вероятность правильного обнаружения дефекта или места возникновения дефекта  $d$ ,

$\bar{N}_{\text{обн}}$  – среднее время запаздывания в обнаружении дефекта  $d$ ,

$P_{\text{ло}}$  – вероятность ложного обнаружения дефекта в режиме НФ,

$P_{\text{ло}} = f_2(h)$ ,

$\bar{N}_{\text{обн}} = f_1(h, d)$ , где  $h$  – порог обнаружения.

Обнаружение и локализация дефекта в случае инвариантов СИ, построенных в соответствии с принципом 1:

$$\begin{aligned} \Delta_1(h) &= y_3(h) - y_2(h), \\ \Delta_2(h) &= y_3(h) - y_1(h), \\ \Delta_3(h) &= y_1(h) - y_2(h); \end{aligned} \quad (5.13)$$

производится согласно правилу 1:

$$|\Delta_1(n)| < h, \quad |\Delta_2(n)| > h, \quad |\Delta_3(n)| > h, \quad (5.14)$$

которое будем называть мажоритарным правилом формирования решения. Условно принято, что дефект возникает в первом ИК.

Обнаружение дефекта в случае СИ, построенных в соответствии с принципом 2, осуществляется согласно правилу 2:

$$\left| y_j(n) - \frac{1}{3} \sum_{j=1}^3 y_j(n) \right| \gtrless h; \quad j = \overline{1, 3}; \quad (5.15)$$

### Правило 1

В режиме НФ инварианты СИ  $\Delta_i$ , имеют характеристики:

$$M[\Delta_i] = 0, \quad D[\Delta_i] = 2\sigma_i^2, \quad i = \overline{1, 3}, \quad (5.16)$$

$$r_{\Delta_1, \Delta_2} = \frac{1}{2}; \quad r_{\Delta_2, \Delta_3} = -\frac{1}{2}, \quad r_{\Delta_3, \Delta_1} = -\frac{1}{2},$$

а в режиме ФД:

$$\begin{cases} M[\Delta_1] = 0; \quad M[\Delta_2] = -k\sigma_I; \quad M[\Delta_3] = k\sigma_I; \\ D[\Delta_1] = 2\sigma_I^2; \quad D[\Delta_2] = D[\Delta_3] = \sigma_I^2[1 + u^2]; \\ R_{\Delta_2, \Delta_3} = -\frac{u^2}{1+u^2}, \quad R_{\Delta_2, \Delta_3} = R_{\Delta_1, \Delta_3} = \frac{1}{\sqrt{2}\sqrt{1+u^2}} \end{cases} \quad (5.17)$$

Вычисление  $P_{lo} = f_2(h)$ ,  $P_{obn} = f_1(h, D)$  при условии  $\frac{h}{\sigma} \gg 2.5$  производится в соответствии с приближенным соотношением:

$$P(|\Delta_1(n)| < h, |\Delta_2(n)| > h, |\Delta_3(n)| > h) \approx P(|\Delta_2(n)| > h, |\Delta_3(n)| > h) \quad (5.18)$$

Вычисление (5.18) в силу зависимости инвариантов  $\Delta_2, \Delta_3$  проводится с использованием двумерного нормального закона распределения [30] с параметрами, определяемыми соотношениями (5.16), для режима НФ или соотношениями (5.17) для режима ФД.

### Правило 2

В режиме НФ инварианты СИ, построенные в соответствии с правилом (5.15), имеют характеристики:

$$\begin{aligned} M[\Delta_i] &= 0, \quad i = \overline{1,3}; \\ D[\Delta_i] &= \frac{2}{3}\sigma_I^2, \end{aligned} \quad (5.19)$$

а в режиме ФД для первого канала имеем:

$$\begin{aligned} M[\Delta_1] &= 0, \quad D[\Delta_1] = \frac{2}{3}(2u^2 + 1)\sigma_I^2, \\ M[\Delta_2] &= M[\Delta_3] = \frac{1}{3}k\sigma_I, \\ D[\Delta_2] &= D[\Delta_3] = \frac{(5+u^2)}{9}\sigma_I^2. \end{aligned} \quad (5.20)$$

Определение  $P_{lo} = f_2(h)$ ,  $P_{obn} = f_1(h, d)$  производится приближенно в соответствии с вычислением вероятности:

$$P\left(\left|x_I - \frac{1}{3}\sum_{j=1}^3 x_j\right| > h\right)$$

с параметрами (5.19), (5.20) в зависимости от рассматриваемого режима с использованием таблиц нормального распределения.

На основе предложенных выше подходов могут быть построены характеристики обнаружения:

$$P_{обн} = F_1(P_{ло}, d),$$

$$\bar{N}_{обн} = F_2(P_{ло}, d),$$

согласно каждому из рассматриваемых дефектов  $d$  типа “ $k$ ” и типа “ $u$ ”. Построенные характеристики обнаружения являются основой для определения наилучшего принципа построения инвариантов, а также области его эффективного использования.

### *Программа работы*

1. Смоделировать три случайных процесса  $y_i(t)$ ,  $i = \overline{1, 3}$ , где каждый процесс рассматривается как последовательность независимых случайных величин, имеющих нормальный закон распределения  $N(0,1)$ .
2. Сформировать инварианты в соответствии с принципом 1 – (5.13).
3. Определить статистические характеристики инвариантов по всем каналам и проверить их соответствие соотношениям (5.16) в режиме НФ.
4. Методом моделирования определить вероятность ложных срабатываний с учетом мажоритарного принципа выбора недостоверного канала. Допусковый порог  $h$  по каждому из инвариантов  $\Delta_i$  установить предварительно в соответствии с уровнем вероятности ложных срабатываний  $P_{ло} = \{0.1, 0.2\}$ , для чего использовать аналитическую модель, основанную на аппроксимации закона распределения значений инвариантов нормальным распределением.
5. Проверить достоверность результатов моделирования п.4 с использованием аналитических моделей по определению  $P_{ло}$ , полученных в (5.18), основанных на двумерном нормальном распределении.
6. Определить среднее время по обнаружению недостоверного канала методом моделирования, при условии, что дефект вводится по первому каналу, за основу принять мажоритарный принцип принятия решений.

Дефект моделируется двумя способами:

- как сдвиг (изменение среднего уровня систематической погрешности),

$$\text{уровень дефекта: } k = \frac{\mu_2 - \mu_1}{\sigma_I} = 2, 3, 4; \quad \mu_1 = 0;$$

– как изменение уровня шумов:  $u = \frac{\sigma_2}{\sigma_1} = 2, 3, 4$ .

Порог  $h$  по каждому из инвариантов зафиксирован на уровне, гарантирующем общую вероятность  $P_{ло}$  с учетом мажоритарного правила принятия решений близкой к  $10^{-2}$ .

7. Проверить достоверность результата моделирования, полученных в п.6 на основе аналитических моделей (5.18) по определению  $P_{обн}$ , основанных на двумерном нормальном законе распределения, учитывая, что  $N_{обн} \approx 1/P_{обн}$ .
8. Провести аналогичное исследование в соответствии с п. 4, 5, 6, 7 для других принципов формирования инвариантов и правила принятия решений в соответствии с соотношением (5.15).

### 5.3. Применение теории распознавания образов к диагностике состояния структурно-избыточных систем

В качестве диагностических признаков выбрана система структурных инвариантов в виде разности одноименных параметров:

$$\begin{aligned}\Delta_1 &= x_3 - x_2, \\ \Delta_2 &= x_3 - x_1, \\ \Delta_3 &= x_1 - x_2,\end{aligned}\tag{5.21}$$

где  $x_j$  – сигнал на выходе  $j$ -го канала.

При отсутствии дефекта:

$$x_j \in N(\mu_1, \sigma_1), \tag{5.22}$$

при наличии дефекта:

$$x_j \in N(\mu_2, \sigma_2), \tag{5.23}$$

где  $\mu_2 = \mu_1 + k \sigma_1$ ;  $\sigma_2 = u \sigma_1$ ;  $k = 1, 2, 3, \dots$ ;  $u = 2, 3, 4, \dots$ .

Параметры  $k$ ,  $u$  определяют уровни дефекта.

Минимальное диагностическое пространство может быть представлено совокупностью двух инвариантов:  $(\Delta_1, \Delta_2)$ , либо  $(\Delta_1, \Delta_3)$ , либо  $(\Delta_2, \Delta_3)$ .

В дальнейшем для определенности выберем систему диагностических признаков  $(\Delta_1, \Delta_2)$ . Определим классы объектов для распознавания в соответствии с возможными состояниями ОД с учетом только однократных дефектов.

1. *Класс 1* – состояние  $S_0$  – отсутствие дефектов.

2. *Классы 2, 4* – состояние  $S_i$  – возникновение дефекта в  $i$ -м канале  $i = 1, 3$ .

Соответствующие характеристики классов могут быть представлены следующим образом:

**Класс 1:**

$$M[\Delta_j] = 0; \quad D[\Delta_j] = 2\sigma_l^2; \quad (j = 1, 2);$$

$$r_{\Delta_1, \Delta_2} = \frac{1}{2}.$$

**Класс 2:**

$$M[\Delta_1] = 0; \quad M[\Delta_2] = -k\sigma_l;$$

$$D[\Delta_1] = 2\sigma_l^2; \quad D[\Delta_2] = \sigma_l^2(1+u^2);$$

$$r_{\Delta_1, \Delta_2} = \frac{1}{\sqrt{2}\sqrt{1+u^2}}.$$

**Класс 3:**

$$M[\Delta_1] = -k\sigma_l; \quad M[\Delta_2] = 0;$$

$$D[\Delta_1] = \sigma_l^2(1+u^2); \quad D[\Delta_2] = 2\sigma_l^2;$$

$$r_{\Delta_1, \Delta_2} = \frac{1}{\sqrt{2}\sqrt{1+u^2}}.$$

**Класс 4:**

$$M[\Delta_1] = k\sigma_l; \quad M[\Delta_2] = k\sigma_l;$$

$$D[\Delta_1] = \sigma_l^2(1+u^2); \quad D[\Delta_2] = \sigma_l^2(1+u^2);$$

$$r_{\Delta_1, \Delta_2} = \frac{u^2}{1+u^2}.$$

В соответствии с этим каждый класс объектов описывается двухмерным нормальным распределением, например, для состояния  $S_0$  (класс 1) имеем:

$$w_I = \frac{1}{(2\pi)^{\frac{I}{2}} \sqrt{\det M}} \exp \left[ -\frac{1}{2} (x - a^{(I)})^T M_I^{-1} (x - a^{(I)}) \right], \quad (5.24)$$

где  $\{a^{(I)}\}^T = \{0, 0\}$ ;

$$M_I = \begin{bmatrix} 2\sigma_I^2 & \sigma_I^2 \\ \sigma_I^2 & 2\sigma_I^2 \end{bmatrix}.$$

### *Программа работы*

1. Разработать обобщенный генератор объектов для указанных выше четырех классов.
2. Проверить достоверность генерации соответствующих классов, для чего по выборке объектов из каждого класса произвести оценку характеристик класса (вектора математических ожиданий  $a^{(k)}$  и корреляционной матрицы  $M_k$ , где  $k = 1, 4$ . Построить зависимости соответствующих оценок от объема выборки и от уровня дефектов ( $k = 1, 2, 3, \dots$ ;  $u = 2, 3, 4, \dots$ ).
3. Рассмотреть детерминированные схемы распознавания в соответствии с соотношениями (3.61), (3.62), (3.63). Для этого в каждом из классов сгенерировать две последовательности: обучающую  $N_0$  и проверочную  $N_m$ . Принять следующие значения параметров:  
 $\sigma_i = 1$ ;  $k = 1, 2, 3$ ;  $u = 2, 3$ ;  $N_0 = N_n = 25$ .
4. Провести эксперименты по распознаванию и заполнить матрицы ошибок  
 $P_{\text{ош}} = \{P_{ij}\}_n$ ,  
где  $P_{ij}$  – оценка вероятности отнесения объекта  $i$ -го класса к  $j$ -му классу.
5. Рассмотреть статистические схемы распознавания в соответствии с соотношениями (3.66), (3.68), используя только проверочную последовательность  $N_n$ , вычислить матрицу ошибок при тех же значениях параметров, что и в п.3.
6. Провести сравнительный анализ процедур распознавания как внутри класса детерминированных и статистических процедур, так и между собой.
7. При построении программ моделирования использовать примеры программ, приведенные в приложениях 5.4, 5.5.

## 5.4. Обнаружение дефектов с использованием метода избыточных переменных

### Программа работы

Структурная схема упрощенной системы управления представлена на рис 5.1.

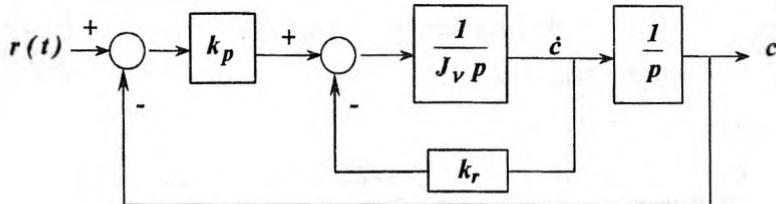


Рис. 5.1

Параметры непрерывной системы управления:

- коэффициент усиления датчика положения:  $k_p = 1.65 * 10^6$ .
- коэффициент усиления датчика скорости:  $k_v = 3.17 * 10^5$ .
- момент инерции:  $J_v = 41822$ .

Предполагается, что все единицы измерения приведены в соответствие друг с другом, так что в аналитических выражениях они умышленно опущены.

1. Построить передаточную функцию замкнутой системы:

$$W_3(p) = \frac{k_p}{J_vp^2 + k_r p + k_p} = \frac{39,453}{p^2 + 8,871p + 39,453}.$$

2. Преобразовать непрерывную передаточную функцию  $W_3(p)$  в модель в пространстве состоян트, используя MATLAB (библиотеку CONTROL). Для этого использовать оператор `tf2ss`.

$$[A, B, C, D] = \text{tf2ss}(\text{num}, \text{den}).$$

Для данного примера

$$\text{num} = [39.453];$$

$$\text{den} = [1 \ 8.871 \ 39.453];$$

$$[A, B, C, D] = \text{tf2ss}(\text{num}, \text{den}).$$

Результат:

$$A = \begin{bmatrix} -8.871 & -39.453 \\ 1.000 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 39.453 \end{bmatrix},$$

$$B = \begin{bmatrix} I \\ 0 \end{bmatrix}, \quad D = [0].$$

Полученный результат представляет собой модель ОД в виде соотношений:

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx, \end{cases} \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

3. Преобразовать непрерывную модель в пространстве состояний в дискретную, используя MATLAB. Для этого использовать оператор **c2d**

$$[\text{Phi}, \text{Gamma}] = \text{c2d}(A, B, T),$$

где  $T$  – период дискретизации

Модель  $\dot{x} = Ax + Bu$

переводится в модель  $x(n+1) = \text{Phi} \cdot x(n) + \text{Gamma} \cdot u(n)$  – (фиксатор нулевого порядка на входах).

4. Определить область устойчивости дискретной системы управления по параметру  $T$ .

Для рассмотренного примера дискретная модель выглядит следующим образом:

$$A = \begin{bmatrix} -8.871 & -39.453 \\ 1.000 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad T = 0.01.$$

$$[\text{Phi}, \text{Gamma}] = \text{c2d}(A, B, T).$$

Результат:

$$\text{Phi} = \begin{bmatrix} 0.9133 & -0.3773 \\ 0.0096 & 0.9981 \end{bmatrix}, \quad \text{Gamma} = \begin{bmatrix} 0.0096 \\ 0.0000 \end{bmatrix}.$$

5. Проверить достоверность проведенного преобразования, используя действие, обратное **c2d**. Для этого используем оператор **d2c**:

$$[A, B] = \text{d2c}(\text{Phi}, \text{Gamma}, T).$$

Пусть задано:

$$\text{Phi} = \begin{bmatrix} 0.9133 & -0.3773 \\ 0.0096 & 0.9981 \end{bmatrix}, \quad \text{Gamma} = \begin{bmatrix} 0.0096 \\ 0.0000 \end{bmatrix}, \quad T = 0.01;$$

тогда получим результат :

$$A = \begin{bmatrix} -8.8649 & -39.4527 \\ 1.0038 & 0.0022 \end{bmatrix}, \quad B = \begin{bmatrix} 1.0029 \\ -0.0049 \end{bmatrix}.$$

Полученный результат свидетельствует о правильности проведенного преобразования.

6. Построить устройство диагностирования (УД), используя модель п.2 непрерывную в пространстве состояний:

$$\text{ОД: } \begin{cases} \dot{x}_1 = a_{11}x_1 + a_{12}x_2 + bu \\ \dot{x}_2 = a_{21}x_1 + a_{22}x_2 + bu \end{cases},$$

$$\text{УД: } \Delta = x_1 + x_2 + z = 0.$$

Расширенная совместная модель ОД и УД в пространстве состояний включает уравнение:

$$z = -(a_{11} + a_{21})x_1 - (a_{12} + a_{22})x_2 - (b_1 + b_2)u$$

и выглядит следующим образом:

$$\begin{aligned} X &= \tilde{A}x + \tilde{B}u \\ Y &= \tilde{C}x + \tilde{D}u \end{aligned},$$

где

$$\begin{aligned} \tilde{A} &= \begin{bmatrix} -8.8710 & -39.453 & 0 \\ 1 & 0 & 0 \\ 7.78710 & 39.453 & 0 \end{bmatrix}; & \tilde{B} &= \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}; \\ \tilde{C} &= \begin{bmatrix} 0 & 39.453 & 0 \\ 0 & 0 & 1 \end{bmatrix}; & \tilde{D} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned}$$

7. Провести моделирование реакции совместного функционирования ОД и УД на входной сигнал в виде единичной ступеньки, используя оператор **step**:

- вводим матрицы  $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ ;
- ввести шаг моделирования  $t = 0 : 0.1 : 2$ ;
- на экран вывести  $y_1, y_2, z$ ;
- ввести дефекты как изменения коэффициентов матрицы  $A$  (только для ОД); убедиться в чувствительности координаты  $z$  к дефекту;
- изменить пространство наблюдаемых переменных, вводя вместо  $z$  значение  $\Delta$ ; тогда матрица  $C$  выглядит следующим образом:

$$\tilde{C} = \begin{bmatrix} 0 & 39.453 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

8. Провести моделирование реакции совместного функционирования ОД и УД на входной сигнал в виде ступеньки, используя оператор **dstep**:

- ввести матрицы  $\tilde{A}, \tilde{B}, T$ ;
- перейти к дискретной модели, используя оператор **c2d**:
- воспроизвести реакцию на ступенчатое воздействие:

$$y = \text{dstep} (\text{Ad}, \text{Bd}, c, d, 1, N),$$

$$A = \text{Phi}, \quad Bd = \text{Gamma},$$

где  $c, d$  – берутся из непрерывного представления,

$1$  – единичный вход,

$N$  – число точек,

$y_1$  – стремится к единице,

$y_2$  – избыточная переменная, которая удерживается близко к нулю.

- проверить чувствительность координаты  $y_2$  к дефекту (изменению коэффициентов матрицы  $A$  объекта диагностирования).

9. Провести моделирование ОД и УД с учетом непрерывной и дискретно-временной модели рассматриваемой системы управления, используя операторы MATLAB: **lsim**, **dlsim** для произвольного входного сигнала  $U(t)$ . При этом рассмотреть режимы:

- правильного функционирования объектов;
- скачкообразного возникновения дефектов в процессе функционирования (дефект моделируется изменением параметров матрицы  $A$ ).

## 5.5. Контроль с использованием фильтра Калмана

Ниже представлены основные соотношения одномерного фильтра; приведенные ранее в разделе 2.3.3.

Объект диагностирования представлен моделью:

$$\begin{aligned} x(n) &= ax(n-1) + g(n-1); \\ y(n) &= ax(n) + v(n). \end{aligned}$$

Уравнение фильтра выглядит следующим образом:

$$\begin{aligned} \hat{x}(n) &= a\hat{x}(n-1) + k(n)[y(n) - ac\hat{x}(n-1)]; \\ k(n) &= \frac{c[a^2 p \sigma^2(n-1) + \sigma_g^2]}{\sigma_v^2 + c^2 \sigma_g^2 + c^2 a^2 p(n-1)}; \end{aligned}$$

$$p(n) = \frac{1}{c} \sigma_v^2 k(n),$$

где  $\sigma_g^2 = \sigma^2(1 - a^2)$  – дисперсия нормального распределения шумов возмущений,

$\sigma_v^2$  – дисперсия нормального распределения шумов измерений.

Исходные данные:

$$c = 1, \quad a = 0.8, \quad (a = -0.8), \quad \sigma = 1, \quad \sigma_v = 2, \quad M[g] = 0, \quad M[v] = 0.$$

## *Программа работы*

1. Смоделировать ОД в виде модели  $AP(1)$  и провести его тестирование.
2. Построить программную модель фильтра и исследовать свойства и характеристики обновляющего процесса  $z(n) = y(n) - a c \hat{x}(n-1)$  (среднее, дисперсию, автокорреляционную функцию, закон распределения).
3. Исследовать чувствительность характеристик процесса  $z(n)$  на выделенном множестве дефектов  $d$ :
  - изменение  $c - \pm 20\%$ ;
  - изменение  $a - \pm 20\%$ ;
  - двукратное увеличение  $\sigma_v$  или  $\sigma$ ;
  - изменение математического ожидания шумов возмущений и измерений  $M[g] = [1; 2; 3]$ ,  $M[v] = [1; 2; 3]$ .
4. Использовать алгоритм сигнального отношения в соответствии с соотношениями (3.40) для обнаружения дефектов, предварительно проведя настройку параметров алгоритма, гарантирующую уровень ложных срабатываний  $P_{lo} = 10^{-2}$ .
5. Исследовать обнаруживающие свойства алгоритма на выделенном множестве дефектов, построив оценку среднего времени обнаружения  $\bar{T}_{obn}(d)$ .
6. Использовать алгоритм обнаружения разладки, основанный на принципе «невязок» в соответствии с соотношением (3.39), проведя настройку параметров алгоритма на уровень  $P_{lo} = 10^{-2}$ ; при этом целесообразно воспользоваться результатами таблицы 3.1.
7. Исследовать обнаруживающие свойства данного алгоритма на выделенном множестве дефектов, построив оценку среднего времени обнаружения  $\bar{T}_{obn}(d)$ .  
Провести сравнительный анализ алгоритмов по результатам моделирования, полученным в п.п. 5, 7 с использованием методов обработки статистики.
9. Исследовать возможности использования алгоритма сигнального отношения (3.40) для выделения последовательности многократных дефектов; при этом предположить, что дефект может по каким-либо неизвестным причинам исчезнуть и появиться в процессе функционирования через некоторое время. В качестве модели такого дефекта использовать появление систематической погрешности в канале измерений  $M[v] \neq 0$ .

При проведении указанной программы работ использовать примеры программ, приведенных в приложении 5.6.

## 5.6. Отбраковка аномальных измерений в потоках диагностической информации

При работе систем диагностирования в реальном времени часто встречается ситуация, когда одно или несколько измерений в потоке диагностической информации отличаются по величине от основного потока измерений. Причинами этого могут оказаться изменение внешних условий, помехи, неправильная работа измерительных средств и др. Такие измерения называются выбросами или аномальными измерениями.

Исследование аномальных измерений, полученных в процессе получения информации, преследует следующие цели:

- исключить аномальные измерения из процедур обработки и формирования управлений;
- выделить измерения, которые могут представлять особый интерес именно из-за их экстремальности и проанализировать возможные причины их появления.

Рассмотрим некоторые наиболее часто употребляемые способы отбраковки одиночных аномальных результатов [31].

1. Выборка независимых нормально распределенных чисел с известными параметрами  $m_x$ ,  $\sigma_x$ , размер выборки  $N$ . Расположим выборку в упорядоченный ряд  $x_1, x_2, \dots, x_n$ , где  $x_1$  – наименьшее значение,  $x_n$  – наибольшее значение. Аномальность крайних значений (например,  $x_n = x_{max}$ ) можно проверить с помощью  $u$ -статистики

$$u_i = \frac{x_i - m_x}{\sigma_x}; \quad i = \overline{1, N}. \quad (5.25)$$

где  $u_i$  представляет собой нормально распределенные случайные величины  $u_i \sim N(0,1)$ .

Таким образом, измерение считается аномальным, если значение статистики  $u_i$  превышает при выбранном уровне значимости квантиль стандартного распределения, т.е.

$$u_i > u_\beta \quad (5.26)$$

где  $u_\beta$  – квантиль нормального распределения. Значение  $u_\beta$  выбирается из условия согласия:

$$[P(x_i < m_x + u_\beta \sigma)] = (1 - \frac{q}{100})^{\frac{1}{N}}; \quad (5.27)$$

где  $q = 10\%; 5\%; 2.5\%; 1\%$  – уровень значимости для всей выборки.

Если вместо  $x_{max}$  в качестве подозрительного результата фигурирует  $x_{min} = x_1$ , то применяется та же процедура, но критерий  $u_i$  заменяется на

$$u_i = \frac{m_x - x_{min}}{\sigma}.$$

2. Имеем тот же случай, но известен только один параметр  $\sigma_x$ , вместо математического ожидания используется его оценка:

$$\hat{x} = \frac{1}{N} \sum_{i=1}^N x_i.$$

Нетрудно показать, что вместо (5.25) должна использоваться статистика

$$u_i = \frac{x_i - \hat{x}}{\sigma_x},$$

которая будет иметь распределение  $N\left(0, \sqrt{\frac{N-1}{N}}\right)$ .

Используя аналитический подход, можно считать крайний элемент  $x_{max}$  выборки аномальным, если выполняется неравенство

$$\frac{x_i - \hat{x}}{\sigma_x} > u_\beta \sqrt{\frac{N-1}{N}}, \quad (5.28)$$

где  $u_\beta$  – определяется из соотношения (5.27).

3. Выборка та же, но неизвестны оба параметра  $m_x$  и  $\sigma_x$ . В этом случае параметры  $m_x$  и  $\sigma_x$  заменяются соответствующими оценками:

$$\hat{x} = \frac{1}{N} \sum_{i=1}^N x_i;$$

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

Неравенство (5.28) заменяется соотношением:

$$\tau_i = \frac{x_i - \hat{x}}{S_x} > \tau_\beta, \quad i = 1, N, \quad (5.29)$$

где статистика  $\tau_i$  имеет специальное распределение, зависящее только от объема выборки  $N$  [31]. Соответствующие рассчитанные квантили при различных значениях доверительной вероятности приведены в табл.5.1.

Таблица 5.1

Объем выборки	Квантили $\tau_\beta$ при различных значениях $1 - q$			
	0.9	0.95	0.975	0.99
3	1.41	1.41	1.41	1.41
4	1.65	1.69	1.71	1.72
5	1.79	1.87	1.92	1.96
6	1.89	2.00	2.07	2.13
7	1.97	2.09	2.18	2.27
8	2.04	2.17	2.27	2.37
9	2.10	2.24	2.35	2.46
10	2.15	2.29	2.41	2.54
11	2.19	2.34	2.47	2.61
12	2.23	2.39	2.52	2.66
13	2.26	2.43	2.56	2.71
14	2.30	2.46	2.60	2.76
15	2.33	2.49	2.64	2.80
16	2.35	2.52	2.67	2.84
17	2.38	2.55	2.70	2.87
18	2.40	2.58	2.73	2.90
19	2.43	2.60	2.75	2.93
20	2.45	2.62	2.78	2.96

При  $N \rightarrow \infty$  статистика  $\tau_i$  стремится к  $u$ -статистике нормального распределения.

4. Выборка имеет произвольное одномерное распределение. Тогда в соответствии с неравенством Чебышева любое выборочное значение  $x_i$  отклоняется от математического ожидания не более, чем на величину

$$\sigma_x / \sqrt{1 - \beta},$$

где  $\beta$  – доверительная вероятность.

Таким образом, если известны математическое ожидание  $m_x$  и дисперсия  $\sigma_x^2$ , то отбраковка аномального измерения при произвольном законе распределения может производиться с помощью статистики

$$u_i = \frac{x_i - m_x}{\sigma_x}, \quad H = \frac{1}{\sqrt{1 - \beta}}. \quad (5.30)$$

где вместо квантиля  $u_\beta$  используется множитель  $1/\sqrt{1 - \beta}$ . Эта замена значительно расширяет область, в которой разброс выборочных значений

относительно среднего объясняется эффектом случайности и не считается аномальным. Так, например, при доверительной вероятности  $\beta = 0.95$  в случае

- нормального распределения  $H = 1.96$ ;
- произвольного распределения  $H = 4.28$ ;
- симметричного одновершинного распределения  $H = 2.82$ .

Все рассмотренные статистики позволяют построить простые вычислительные алгоритмы для отбраковки аномальных измерений в том случае, если исходный процесс является нормальным, а его выборочные значения статистически независимы.

5. Методы, позволяющие учесть корреляционные зависимости между измерениями. Один из подходов основан на вычислении текущих разностей между соседними измерениями и сравнением их с порогом.

$$\Delta x_i \geq \Delta\beta/2, \quad (5.31)$$

где  $\Delta x_i = x_i - x_{i-1}$ ,  $i = 1, N$ ,  $\beta$  – доверительная вероятность.

При выполнении условия (5.31) проверяемый результат измерения признается аномальным. Случайная величина  $\Delta x_i$ , выбранная из из стандартного случайного процесса, описываемого одномерным нормальным распределением тоже имеет нормальный закон распределения  $N(\theta, \sigma_{\Delta x_i}^2)$  где  $\sigma_{\Delta x_i}^2$  – связана с дисперсией  $\sigma_x^2$  исходного процесса и его нормированной корреляционной функцией и при использовании только корреляции между соседними отсчетами определяется следующим соотношением:

$$\sigma_{\Delta x_i}^2 = 2\sigma_x^2(1 - \rho_1). \quad (5.32)$$

Известно, что дисперсия возрастает с увеличением порядка разности и уменьшением корреляции между соседними отсчетами, поэтому применение разностей выше первого порядка нецелесообразно [31].

6. Использование порядковых статистик. Порядковые статистики формируются следующим образом. Величины выборки  $x_1, x_2, \dots, x_n$ , выстраиваются в порядке возрастания. В образованной таким образом ранжированной последовательности (вариационном ряду) величины обозначаются через  $x_{(i)}$ , причем

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(N)}.$$

Величина  $x_{(i)}$  называется  $i$ -ой порядковой статистикой и является  $i$ -ой в порядке возрастания величиной среди набора  $x_1, x_2, \dots, x_n$ . С точки зрения практических приложений при решении задач отбраковки наиболее

важной порядковой статистикой является выборочная медиана (средний член вариационного ряда).

Для трех измерений  $x_1, x_2, x_3$  выборочная медиана обозначается как  $\text{med}(x_1, x_2, x_3)$ . При наличии выбросов, аномальные измерения обычно оказываются на краях вариационного ряда и выборочная методика дает устойчивую оценку в отличие от среднего арифметического. Вследствие этого применение алгоритма выделения медианы при преобразовании последовательностей дискретных данных позволяет использовать для повышения надежности и точности не только структурную, но и временную избыточность. Например, при использовании алгоритма скользящей медианы по выборке из трех измерений выходной сигнал преобразования представляет собой медиану трех последних значений временной последовательности

$$z(j) = \text{med}(x_{(j)}, x_{(j-1)}, x_{(j-2)}).$$

Естественно, что данный способ позволяет исключить одиночные аномальные измерения, но вносит запаздывание на шаг выходного сигнала.

### Программа работы

- Смоделировать независимую случайную последовательность  $x_1, x_2, \dots, x_n$ , имеющую закон распределения:

$$f(x) = p f_1(x) + (1-p) f_2(x), \quad (5.33)$$

где  $f_1(x) \sim N(m_x, \sigma_x^2)$ ,

$$f_2(x) \sim N(\theta, r\sigma_x^2).$$

Принять значения параметров  $p = 0.9; r = 4; m_x = 0, \sigma_x = 1$ .

Распределение (5.33) моделирует процесс возникновения аномальных измерений с вероятностью  $p = 0.1$ .

- Построить зависимости оценок  $\hat{x}$  и  $S_x$  от объема выборки  $N$

- при  $p = 0.1$  – отсутствие аномальных измерений;
- при  $p = 0.9$  – отсутствие аномальных измерений;

- Использовать способ отбраковки № 4 при пороге  $H = 1.96$  и  $\beta = 0.95$ , построить зависимость  $\hat{x}$  и  $S_x$  от объема выборки  $N$  ( $N = 5, 10, 25, 50, 100$ ),  $p = 0.9$ . Сравнить результаты с п.2.
- Использовать способ отбраковки № 6 на основе алгоритма скользящей медианы при  $N = 1$ , построить зависимость  $\hat{x}$  и  $S_x$  от объема выборки  $N$ , условия те же самые, что и в п.1.

5. Использовать алгоритм выделения медианы при наличии трехканальной структурной избыточности для обнаружения недостоверного канала. Правило обнаружения выглядит следующим образом:

$$\begin{aligned} |\Delta_1| &= |x_1 - \text{med}(x_1, x_2, x_3)| \geq h, \\ |\Delta_2| &= |x_2 - \text{med}(x_1, x_2, x_3)| \geq h, \\ |\Delta_3| &= |x_3 - \text{med}(x_1, x_2, x_3)| \geq h. \end{aligned} \quad (5.34)$$

Все измерения  $x_1, x_2, x_3$  относятся к одному моменту времени.

В режиме НФ все измерения имеют  $N(\theta, \sigma_x)$ . В режиме ФД в качестве дефекта рассматривается наличие аномальных измерений в первом канале, т.е.  $x_1$  имеет распределение (5.33) с параметрами  $p = 0.9; r = 4$ .

Методом моделирования построить характеристики обнаружения  $P_{обн} = P(P_{ло}, D)$ . Диапазон изменения порога  $h$  для  $P_{ло}$  от  $10^{-1}$  до  $10^{-2}$ . При выборе порога в режиме НФ использовать известный результат: статистика  $\text{med}(x_1, x_2, x_3)$  при одинаковых свойствах каналов  $N(\theta, \sigma_x)$  имеем приближенно нормальное распределение  $N(\theta, \sigma)$ , где

$$\sigma^2 = \frac{\pi \sigma_x^2}{4 + \pi}. \quad (5.35)$$

При проведении исследований использовать примеры программ, приведенных в приложениях 5.7 и 5.8.

## 5.7. Использование аналитической зависимости для диагностирования двухканальных систем

Данный метод диагностирования подробно описан в [1]. Рассматривается система со структурной избыточностью, содержащая два параллельных канала с общим входом.

Структурная избыточность позволяет обнаружить наличие дефекта, а имеющаяся аналитическая избыточность позволяет определить номер недостоверного канала.

Пусть задана система из двух идентичных каналов с передаточной функцией (ПФ):

$$Q(p) = \frac{B(p)}{A(p)}, \quad (5.36)$$

где  $A(p)$  и  $B(p)$  – взаимно простые многочлены от  $p$  степеней  $n$  и  $m (m \leq n)$ .

Тогда проверка правильности функционирования может быть осуществлена путем сравнения сигнала невязки  $\Delta = y_1 - y_2$  с нулем.

Практически признаком правильной работы является выполнение неравенств для некоторой нормы невязки

$$\|\Delta(t)\| < \varepsilon, \quad (5.37)$$

где  $\varepsilon$  – заранее установленный порог.

Будем рассматривать дефекты, приводящие к константным изменениям коэффициентов числителя и знаменателя ПФ каналов. Назовем дефект, состоящий в искажении только одного коэффициента ПФ однократным и покажем, что анализируя сигналы  $y_1$ ,  $y_2$  и  $\Delta$  можно осуществить локализацию однократных дефектов.

Пусть

$$A^*(p) = A(p) + \Delta a_i p^i,$$

для одного из каналов (например, первого), где  $\Delta a_i$  – неизвестная постоянная величина. Тогда при нулевых начальных установках невязка определяется операторным уравнением:

$$\Delta = y_1 - y_2 = \frac{B(p)}{A^*(p)} x - \frac{B(p)}{A(p)} x = -\Delta a_i \frac{p^i B(p)}{A(p) A^*(p)} x. \quad (5.38)$$

Из (5.38) видно, что признаком однократного дефекта  $\Delta a_i$  является пропорциональность сигнала невязки  $\Delta$  выходному сигналу неисправного канала:

$$y_i^* = \frac{B(p)}{A^*(p)} x,$$

пропущенному через фильтр с ПФ  $p^i/A(p)$ .

Следовательно, сравнивая измеренный сигнал  $\Delta$  на пропорциональность с сигналами  $S_1 = y_1/A(p)$  и  $S_2 = y_2/A(p)$  и их производными, можно установить номер неисправного канала и индекс искаженного коэффициента знаменателе его ПФ. Величина ошибки однозначно определяется коэффициентом пропорциональности.

При искажении числителя ПФ того же канала:

$$B^*(p) = B(p) + \Delta b_j p^j$$

невязка выглядит следующим образом:

$$\Delta = y_1 - y_2 = \frac{B^*(p)}{A(p)}x - \frac{B(p)}{A(p)}x = \Delta b_j \frac{p^j}{B(p)} \frac{B(p)}{A(p)}x. \quad (5.39)$$

В данном случае сигнал  $\Delta(t)$  оказывается пропорциональным сигналу исправного канала  $y_2$ , пропущенного через фильтр с ПФ  $p_j/p(B)$ . Следовательно, сравнивая сигнал невязки  $\Delta(t)$  на пропорциональность с сигналами  $S_3 = y_1/B(p)$ ,  $S_4 = y_2/B(p)$  и их производными можно установить номер неисправного канала и индекс искаженного коэффициента в числителе ПФ.

### *Программа работы*

Пусть ПФ каналов  $Q$ :

$$Q_1(p) = Q_2(p) = Q(p) = \frac{k}{Tp + 1}, \quad (5.40)$$

где  $T = 1$ ,  $k = 1$ .

В качестве дефекта рассматривается константная неисправность  $T^* = 0.333$ .

1. Рассмотреть детерминированный случай, входной сигнал  $x(t)$  представить в виде единичной ступеньки  $I(t)$ .
2. Проанализировать чувствительность сигнала невязки  $\Delta(t)$  к факту возникновения указанного дефекта  $\Delta T$ .
3. Разработать алгоритм обнаружения и локализации дефекта на основе использования соотношения (5.38).
4. Рассмотреть статистический случай, входной сигнал  $x(t)$  представить в виде белого шума с единичной дисперсией.  
Проанализировать корреляционные функции процессов  $y_1(t)$ ,  $y_2(t)$  и  $\Delta(t)$ .
5. Разработать алгоритм обнаружения дефекта с использованием алгоритма сигнального отношения и алгоритм локализации дефекта с использованием соотношения (5.38).
6. Провести моделирование с оценкой показателей качества обнаружения и локализации выбранного дефекта.

## 5.8. Применение нейронных сетей в задачах диагностики

### 5.8.1. Решение задачи о разладке случайной последовательности с использованием аппарата нейронных сетей

#### Программа работы

- Сформировать два СП – процесса авторегрессии второго порядка с параметрами соответственно  $m_1 = 0, \sigma_1 = 1, a1_1 = 0.1, a2_1 = 0.1$  и  $m_2 = k, \sigma_2 = 0, a1_2 = 0.1, a2_2 = 0.1$  (взять  $k = 1, 5$ ).
- Смоделировать НС (рис. 4.10). Для этого при создании НС число входных элементов взять  $N_{\text{ex}} = 1, 10$ , а число нейронов в скрытом слое взять  $N = 4, 10$  (использовать функцию `newff`). Обучить НС, взяв в качестве обучающей последовательности  $N_0 + N_1$  реализаций  $AR(p)$  ( $N_0 = 50, 100, 300$  отсчетов СП без разладки и  $N_1 = 50, 100, 300$  отсчетов СП с разладкой). Использовать функцию `train`. Изменяя параметры НС добиться успешного окончания процесса обучения.
- Проверить работу обученной НС на новой (вновь сформированной) проверочной СП. Оценить показатели качества работы НС (вероятность ложного обнаружения, вероятность обнаружения, число циклов обучения, время обучения, достигнутую суммарную ошибку обучения) в зависимости от числа элементов задержки (при фиксированных остальных параметрах), от числа нейронов в скрытом слое, от длины обучающей последовательности и при различном соотношении длины участков СП с разладкой и без разладки. Проверить работоспособность НС, обученной на СП с уровнем дефекта большим, чем в проверочной последовательности. Результаты представить в виде таблиц и графиков.
- Сформировать два СП – процесса авторегрессии второго порядка с параметрами соответственно  $m_1 = 0, \sigma_1 = 1, a1_1 = 0.1, a2_1 = 0.1$  и  $m_2 = 0, \sigma_2 = u, a1_2 = 0.1, a2_2 = 0.1$  (взять  $u = 2, 6$ ).
- Построить НС аналогично п.2, но для обнаружения изменения дисперсии СП. Провести исследование полученной НС согласно п.3.
- Смоделировать три СП – процесса авторегрессии второго порядка с параметрами  $m_1 = 0, \sigma_1 = 1, a1_1 = 0.1, a2_1 = 0.1$  до возникновения разладки и с параметрами  $m_2 = k, \sigma_2 = u, a1_2 = 0.1, a2_2 = 0.1$  после возникновения разладки. В одном из СП смоделировать возникновение разладки (типа  $k$  или типа  $u$ ) в момент времени  $t$ . Сформировать из СП три структурных АИ, считая, что ОД – трехканальная однопараметрическая система, в которой в некоторый момент времени

вышел из строя один из каналов. В результате будут получены АИ, два из которых – СП с участком без разладки и с участком с разладкой.

7. Построить НС для реализации логической функции двухвходовой «И».
8. Реализовать модель НС для локализации и определения типа дефекта в ОД (рис. 4.14), используя в качестве составных элементов уже обученные НС (п.2, п.5, п.7), а в качестве входных сигналов использовать последовательности АИ, полученных в п.6.
9. Оценить показатели качества принятия решений при использовании созданной НС.

### **5.8.2. Обнаружение недостоверного канала в структурно-избыточных системах с использованием нейронной сети**

#### **Программа работы**

1. Смоделировать три СП – процесса авторегрессии второго порядка, состоящих из двух участков, один с параметрами  $m_1 = 0$ ,  $\sigma_1 = 1$ ,  $aI_1 = 0.1$ ,  $a2_1 = 0.1$ , второй с параметрами  $m_2 = k$ ,  $\sigma_2 = u$ ,  $aI_2 = 0.1$ ,  $a2_2 = 0.1$ , ( $k = 0, 1, 2, 6$ ;  $u = 1, 2, 6$ ).
2. В одном из СП смоделировать возникновение разладки (типа  $k$  или типа  $u$ ) в момент времени  $t$ . Сформировать из СП три структурных АИ, считая, что ОД – трехканальная однопараметрическая система, в которой в некоторый момент времени вышел из строя один из каналов. В результате будут получены АИ, два из которых – СП с участком без разладки и с участком с разладкой.
3. Смоделировать НС (рис. 4.11). Для этого при создании НС число входных элементов взять  $N_{\alpha} = 3, 10$ , а число нейронов в скрытом слое взять  $N=4, 10$  (использовать функцию `newff`). Обучить НС, взяв в качестве обучающей последовательности  $N_0 + N$  реализаций  $AR(p)$  ( $N_0 = 50, 100, 300$  отсчетов СП без разладки и  $N_1 = 50, 100, 300$  отсчетов СП с разладкой). Использовать функцию `train`. Изменяя параметры НС добиться успешного окончания процесса обучения.
4. Проверить работу обученной НС на новой (вновь сформированной) проверочной СП. Оценить показатели качества работы НС (вероятность ложного обнаружения, вероятность обнаружения, число циклов обучения, время обучения, достигнутую суммарную ошибку обучения) в зависимости от числа элементов задержки (при фиксированных остальных параметрах), от числа нейронов в скрытом слое, от длины обучающей последовательности и при различном соотношении длины участков СП с разладкой и без разладки. Проверить работоспособность

НС, обученной на СП с уровнем дефекта большим, чем в проверочной последовательности. Результаты представить в виде таблиц и графиков.

5. Смоделировать НС (рис. 4.12) аналогично п.3 и провести исследование согласно п.4.
6. Смоделировать НС (рис. 4.13) аналогично п.3 и провести исследование согласно п.4.

### % Приложение 5.1

```
clc, clear
disp('Программа моделирования случайной величины и
вычисление ее характеристик')
    % Задание исходных данных
disp('Задайте математическое ожидание и дисперсию для
моделирования')
disp('    случайной величины, распределенной по
нормальному закону')
mo = input('    mo = ');    sigm = input('    sigm = ');
disp(' Задайте количество реализаций (100 - 2000)'), 
n = input ('    n = ');
    % Моделирование
x=normrnd(mo,sigm,1,n);
    % Оценка параметров распределения
disp('Вычисленные МО и дисперсия случайной величины:')
mo_= mean(x), sigma_= std(x), pause
    % Построение графиков
subplot(211), plot(x,'r'), grid, hold on,
    mol=mo_*ones(1,n); plot(mol,'g'), hold off
    title(' Последовательность случайных чисел')
subplot(212), hist(x,100), grid,
    title('Гистограмма плотности распределения
случайной величины'), pause
end
```

### % Приложение 5.2

```
clc, clear, echo on
    % Программа моделирования случайных векторов
%Заданная корреляционная матрица генерируемого вектора:
    % echo off
r = [1 0.8; 0.8 1]
disp(' '),
n = input(' Введите число реализаций (рекомендуем
n>1000) n = ');
    % Моделирование
q1 = chol(r);
    % верхнетреугольную матрицу обращаем в нижнетреугольную:
q = q1';
```

```

z1 = zeros(n,2); z2=z1;

for i=1:n
    x1=randn(2,1); z1(i,:) = x1';
    x2 = q*x1;      z2(i,:) = x2';
end

% Проверка результатов моделирования:

disp(' ')
disp(' Вычисленная корреляционная матрица
генерированного вектора:')
r2 = cov(z2)

% Построение графиков:

subplot(221), plot(z1(1:200,1), z1(1:200,2),'ro'), grid
title('Некоррелированный случайный вектор')

subplot(222), plot(z2(1:200,1), z2(1:200,2),'bo'), grid
title('Коррелированный случайный вектор')
end

```

### % Приложение 5.3

```

clc, clear, clg, echo on

% Программа моделирования случайного процесса,
% представленного моделью AP1: x(n) = a * x(n-1) + g
% Исходные данные

n=1000; % объем выборки
a = 0.8;
sigm = 1;
sigmag = sigm * sqrt(1 - a^2);
echo off, pause
xm = zeros(1,n);

% Моделирование

x1 = 0;
for i = 1:n
    g = randn * sigmag;
    x = a * x1 + g;
    xm(1,i) = x; x1 = x;
end

```

```

% Анализ результатов

disp(' '),
disp(' Вычисленные статистические параметры
генерированного процесса:')
mo_= mean(xm), sigm_ = std(xm),

disp(' '), disp(' Вычисленная автокорреляционная
функция:')
r = covf2(xm', 10),
a_= r(2)

% Графическое представление результатов

subplot(211), plot(xm,'g'), grid
title('Случайный процесс авторегрессии 1-го порядка')
subplot(212), plot(r,'r'), grid,
title('Автокорреляционная функция этого процесса')
end

```

#### % Приложение 5.4

```

echo on, clear, clc
    % Программа распознавания образов,
    % основанная на геометрическом подходе

% Разделяемые классы представляют собой две выборки
% случайных двухэлементных векторов x1 и x2. Элементы
% каждого вектора интерпретируем как координаты точек
% на плоскости. Тогда разделяемые классы (будем называть
% их красным и зеленым) представляют собой множество
% точек в окрестности центров классов: точек mo1 и
% mo2 - математических ожиданий каждой из выборок.

```

#### % 1. Исходные данные

```

sigmal = 1; sigma2 = 1;
mo1 = [1 1];      % Центр класса красных
mo2 = [3 3];      % Центр класса зеленых

```

```

% Для прерывания паузы нажмите любую клавишу
% Рекомендуемая длина выборки 100 - 200.

```

```

echo off
n = input('    Введите длину выборок n = ');

```

#### % 2. Генерирование классов

```

x1 = zeros(n,2);    x2 = zeros(n,2);

```

```

for i=1:n
% элемент выборки красных:
    x01 = randn(1,2)*sigma1 + m01;
% элемент выборки зеленых:
    x02 = randn(1,2)*sigma2 + m02;
    x1(i,:) = x01; x2(i,:) = x02;
end

% 3. Графическое представление классов:

subplot(121), plot(x1(:,1), x1(:,2), 'ro', x2(:,1),
x2(:,2), 'go')
xlabel('x(i,1)'), ylabel('x(i,2)'),
title('Граф. предст. выборок, синяя линия соединяет их
центры'), hold on,
xmo = [m01(1,1) m02(1,1)]; ymo = [m01(1,2) m02(1,2)];
plot(xmo,ymo,'b-',xmo,ymo,'b*')
disp(' '), disp(' '), pause

% 4. Разделение классов и расчет ошибок разделения

% критерий разделения - минимальное расстояние точки
% до центра ее класса

rg=0; gr=0; xrg=zeros(n,2); xgr=zeros(n,2);

for i = 1:n
    % Проверяем красных:
    r1 = norm(x1(i,:)-m01); r2 = norm(x1(i,:)-m02);
    if (r2 - r1 >= 0), % правильно определена
красная точка
        else rg = rg+1; % ошибка: красная точка
принята за зеленую
        xrg(rg,:) = x1(i,:);
    end

    % Проверяем зеленых:
    g1 = norm(x2(i,:)-m02);
    g2 = norm(x2(i,:)-m01);
    if(g2-g1>=0) % правильно определена зеленая точка
    else gr=gr+1;%ошибка: зеленая точка принята за кр.
        xgr(gr,:) = x2(i,:);
    end
end

rgn=rg/n; grn=gr/n; disp(' ')

```

```

disp('          Оценка ошибок разделения классов:')
disp('          ~~~~~')
sprintf('Относительное число красных точек, принятых за
зеленых %g ',rgn),
sprintf('Относительное число зеленых точек, принятых за
красных %g ',grn),
% 5. Графическое представление ошибок разделения
subplot(122), plot(x1(:,1), x1(:,2),'ro',x2(:,1),
x2(:,2), 'go')
xlabel('x(i,1)'), ylabel('x(i,2)'),
hold on, xmo = [m01(1,1) m02(1,1)];
ymo = [m01(1,2) m02(1,2)];
plot(xmo,ymo,'b-',xmo,ymo,'b*:')
% отсекаем незаполненные части массивов:
xrg(rg+1:n,:)=[];
xgr(gr+1:n,:)=[];
plot(xrg(:,1),xrg(:,2),'y.', xgr(:,1),xgr(:,2), 'y.'),
title('Ошибки разделения классов помечены желтыми
точками'), end

```

## % Приложение 5.5

```

clear, echo on
% Программа распознавания образов,
% основанная на статистическом подходе

% 1. Исходные данные

m01=[1; 1];      % вектор матожидания "красной" выборки
m02=[2.5; 2.5]; % вектор матожидания "зеленой" выборки
r=[1 0; 0 1];    % корреляционная матрица
echo off

n=input('Введите размер выборок: ');
disp('Для прерывания паузы нажмите любую клавишу'),
disp(' '), pause

% 2. Генерирование двух выборок: векторов x1 и x2

[m,m] = size(r); l = chol(r); lt = l';
for i = 1:n
    x1 = lt*randn(m,1) + m01; x1m(i,:) = x1';
    x2 = lt*randn(m,1) + m02; x2m(i,:) = x2';
end

```

```

disp('Вычисленные корреляционные матрицы генерированных
выборок:');
disp(' ');
r1 = cov(x1), r2 = cov(x2),
% 3.Разделение классов и расчет ошибок разделения
xrg = zeros(n,2); xgr = zeros(n,2); rg=0; gr=0;
for i = 1 : n
    % Проверяем красных:
    d = x1m(i,:)*inv(r)*(mo2-mo1)-
        0.5*(mo1+mo2)'*inv(r)*(mo2-mo1);
    if (d <= 0), % красная точка определена правильно
    else rg=rg+1; xrg(rg,:)=x1m(i,:); % ошибка
    end
    % Проверяем зеленых:
    d = x2m(i,:)*inv(r)*(mo2-mo1)-
        0.5*(mo1+mo2)'*inv(r)*(mo2-mo1);
    if(d >= 0) % зеленая точка определена правильно
    else gr=gr+1; xgr(gr,:)=x2m(i,:); % ошибка
    end
end
% 4.Аналитическая оценка ошибок разделения классов
a = sqrt((mo2-mo1)'*inv(r)*(mo2-mo1)); sq = sqrt(2);
alfa = 0.5*[1- erf(a/(2*sqrt(2)))]; beta = alfa;
disp('Оценки ошибок разделения классов - аналитическая
и экспериментальная'), disp(' ');
rgn = rg/n;
disp('Относит. число красных принятых за зеленых:');
sprintf('    теория :    %g', alfa),
sprintf('    эксперимент: %g', rgn),
grn=gr/n;
disp('Относит.число зеленых принятых за красных:');
sprintf('    теория :    %g', beta),
sprintf('    эксперимент: %g', grn),

```

```

% 5.Графическое представление классов
xmo = [mo1(1,1) mo2(1,1)]; ymo = [mo1(2,1) mo2(2,1)];
subplot(121),
plot(x1m(:,1),x1m(:,2),'ro', x2m(:,1),x2m(:,2),'go',
xmo,ymo,'b',xmo,ymo,'b*')
xlabel('x(i,1)'), ylabel('x(i,2)'),
title('Классы объектов, синяя линия соед.их центры'),
pause

% 6.Графическое представление ошибок разделения
subplot(122),
plot(x1m(:,1),x1m(:,2),'ro', x2m(:,1),x2m(:,2),'go',
xmo,ymo,'b',xmo,ymo,'b*')
hold on, xlabel('x(i,1)'), ylabel('x(i,2)'),
xrg(rg+1:end,:)=[]; xgr(gr+1:end,:)=[]; % отсекаем
незаполненные части массивов
plot xrg(:,1), xrg(:,2),'y.', xgr(:,1), xgr(:,2), 'y.')
title(' Ошибки раздел. классов помечен желт.точками'),
hold off, end

```

### % Приложение 5.6

```

clc, clear, echo on
%% Программа моделирования скалярного фильтра Калмана
% с процедурой обнаружения дефекта

% Исходные данные:
% Параметры сигнала (x):
a=0.8; sigm=1; sigmx = sigm*sqrt(1-a^2);

% Параметры наблюдения (y):
c=1; sigmy=2;
% Начальные значения:
p=sigmy;
x1=0; % начальное наблюдение
x1_ = sigmx; % начальная оценка наблюдения
pause

% Параметры режима моделирования:
% Процесс моделирования будет представлять собой
% реализаций процесса фильтрации, каждая из которых
% моделирует r шагов фильтрации. На заданном шаге
% процесса моделирования (tdef) в каждой реализации

```

% может быть введен дефект математического ожидания или  
% дисперсии сигналов в канал наблюдения или в канал  
% измерения.

% Обозначения режимов работы:

% 0 - режим нормальной работы;  
% 1 - дефект первого рода (дефект постоянный);  
% 2 - дефект второго рода (дефект случайный);  
% 3 - дефект процесса в канале наблюдения;  
% 4 - дефект в канале измерения.

```
echo off
disp(' '), n=input('Введите число реализаций n = ');
disp(' '), r=input('Введите число шагов в реализации (~100) r = ');
disp(' '), nd=input('Введите номер режима работы (0-норм, 1-деф. МО., 2-деф.sigma.): ');
if nd>2, error('Неверный ввод номера режима работы'), end

if nd==0, tdef=r+1; def=0; defgconst=0; defgstoh=0;
defvconst=0; defvstoh=0;
else
    disp(' '), md=input('Введите место дефекта: (3-наблюдение, 4-измерение): ');
    if (md<3|md>4),
        error('Неверный ввод места дефекта'), end

    disp(' '), tdef=input('Введите номер шага возникновения дефекта(7<tdef<r): ');
    disp(' '), def=input('Введите величину дефекта, нормированную sigma (1-10): ')
end
echo on
```

% Параметры алгоритма обнаружения:

```
% В качестве процедуры обнаружения используется
% алгоритм сигнального отношения.
% Хорошее качество алгоритма можно получить при
% porog=0.8; 0.15<=alfa<=0.4.
echo off
porog=input('Введите величину порога обнаружения
(0<=porog<=1): ');
alfa=input('Введите параметр сглаживания (0<=alfa<=1): ');
xm=zeros(1,r); um=xm; tzm=zeros(1,n);
stz=0; sobn=0; slobn=0;
```

```

% Моделирование

disp(' '), disp(' Процесс моделирования'),
disp('Номера реализаций')
for n1=1:n; disp(n1) % выбор номера реализации

defxmo=0; defymo=0; defxsigma=1; defysigma=1;
kz=0; kz1=0; tz=0; obn=0; lobn=0; zm=zeros(1,r);
sk=0; rk=0; q=zeros(1,r);

%=====
% Моделирование каждой реализации:

for i=1:r
    if i>=tdef, % введение дефекта на i-м шаге:
        if (nd==1 & md==3), defxmo=def; end,
        if (nd==1 & md==4), defymo=def; end,
        if (nd==2 & md==3), defxsigma=def; end,
        if (nd==2 & md==4), defysigma=def; end,
    end

    % Моделирование состояний:

x = a*x1 + sigmx*(randn*defxsigma + defxmo);
xm(1,i)= x; x1=x;

    % Моделирование наблюдений:

y = c*x + sigmy*(randn*defysigma + defymo);
ym(1,i)= y;

    % Вычисление коэффициента усиления и
    % среднеквадратичной ошибки фильтрации:

k = c*(a^2*p+sigmx^2)/(sigmy^2+c^2+sigmx^2+ c^2*a^2*p);
km(1,i) = k; p = sigmy^2*k / c; pm(1,i) = p;

    % Обновляющий процесс и оценка вектора состояний:

y_=c*a*x1_; y_m(1,i)=y_;
z=y-y_; zm(1,i)=z;
x_=a*x1_+ k*z; x_m(1,i)=x_;
x1_=x_;

    % Процедура обнаружения:

sk = sk*(1-alfa)+zm(1,i)*alfa;
rk=rk*(1-alfa)+abs(zm(1,i))*alfa;
q(1,i) = sk/rk; % сигнальное отношение

```

```

if (i>7 & abs(q(1,i)) > porog),
    kz=kz+1;% счетчик превышений порога
    if (nd~=0 & i>tdef),
        kz1=kz1+1; %счетчик правильных обнаружений
        if kz1==1, obn=1; tz=i-tdef; end
    else lobn = lobn+1; %счетчик ложных обнаружений
    end
end
end % конец реализации
=====
% Накопление переменных для расчета
% показателей качества обнаружения:

sobn = sobn + obn; % количество правильных обнаружений
slobn = slobn + lobn; % количество ложных обнаружений
stz = stz + tz; tzm(1,n1) = tz; % время задержки
end % конец моделирования

disp('Результат работы процедуры обнаружения дефекта:')
disp('*****')

if nd~=0, pnobn=(n-sobn)/n; else pnobn=0, end
pnobnl=sprintf('Вероятность необнаружения= %g', pnobn),
plobn = slobn/(n*r);
pll=sprintf('Вероятность ложных обнаружений=%g',plobn),
if sobn>0, srtz=stz/sobn; else srtz=0; end
srtzl=sprintf('Среднее время задержки обн.= %g', srtz),
if sobn>0, dsptz=std(tzm); else dsptz=0; end
dsptzl=sprintf('Дисперсия времени задержки обнаружения
= %g', dsptz), pause

see=input('Представить процесс моделирования
графически? (1--0)');
if see==1,disp('На графиках представлена последняя
реализация процесса моделирования.')
subplot(221),plot(km,'r'),hold on,
            plot(pm,'b'), hold off, grid,
            title('Коэф.усил.(красн.) и дисп.ошибок фильтрации'),
subplot(222),plot(xm,'g'),hold on,
            plot(x_m,'r'),hold off, grid,
            title('Наблюдения (зелен.) и оценка наблюдений'),
subplot(223),plot(ym,'b'),hold on,

```

```

    plot(y_m,'m'), hold off, grid,
    title('Измерения (синий) и оценка измерений'),
    subplot(224), plot(zm,'c'), hold on, plot(q,'k'),
    w = ones(1,r)*porog;
    plot(w,'r'), plot(-w,'r'), hold off, grid,
    title('Обновл.процесс, пороги (красн.), сигн. отнож.
    (черн.)'),
    end

```

### % Приложение 5.7

```
clc, echo on
```

#### % Моделирование работы медианного фильтра

```
% Сравнение процессов: исходного x (первый столбец),
% процесса с выбросами y (второй) и процесса z (третий),
% полученного фильтрацией медианным фильтром процесса y.
```

```
n=40; x=zeros(n,1); y=x; xyz=zeros(n,3);
mx = 0; sigmax = 1; % параметры процесса без выбросов
my = 0; sigmay = 5; % параметры процесса выбросов
pl=0.2; % вероятность выброса
pause, echo off

x = mx + sigmax*randn(n,1);

for i = 1:n
    p=rand;
    if p>pl y(i,1) = x(i,1);
    else      y(i,1) = my + sigmay * randn(1);
    end
end
```

%процесс с выбросами пропускаем через медианный фильтр:

```
[z] = median(y);

xyz(:,1) = x; xyz(:,2) = y; xyz(:,3) = z';
disp(xyz), disp('Для просмотра графика нажмите любую
клавишу'), pause,
```

```
plot(x,'g'), hold on, plot(y,'b'), plot(z,'r'), grid,
hold off
title('зеленый-случ.процесс, синий-сл.пр. с выбросами,
красный - он же после медианного фильтра')
end
```

```

function[z] = median(x);
% медианный фильтр для одномерного случайного процесса

[a,b] = size(x); n = max(a,b);
z = zeros(1,n); c = zeros(1,3); y1 = c;

if a == n, x = x'; end

for i = 2 : (n - 1);
    c = [x(1,i-1), x(1,i), x(1,i+1)];
    y1 = sort(c); z(1,i) = y1(1,2);
end

```

### % Приложение 5.8

```

clc, clear, echo on
% Программа моделирования и исследования случайных
% величин при наличии и отсутствии аномальных измерений

mx = 0; sigmax = 1; % параметры процесса без выбросов
my = 0; sigmay = 5; % параметры процесса с выбросами

p1=0.2; % вероятность выброса
n=500; % число реализаций случайных величин
pause, echo off

for i = 1:n
    x(1,i) = mx + sigmax*randn;
    p=rand;

    if p>p1 y(1,i) = x(1,i);
    else      y(1,i) = my + sigmay * randn;
    end
end

%Пропускаем процесс y с выбросами через медиану. фильтр:
z = median(y);
echo on
% Результаты:

% Статистика процесса x:
mox = mean(x), sigmx = std(x),
% Статистика процесса y (с выбросами):
moy = mean(y), sigmy = std(y),
% Статистика процесса z=median(y):
moz = mean(z), sigmz = std(z),
echo off

```

```
% Графики процессов и их гистограммы:  
subplot(221), plot(x(1,1:30), 'b'), hold,  
    plot(y(1,1:30), 'r'), hold, grid,  
    title('Процессы x(исх.) и y(с выброс.)'),  
subplot(222), plot(y(1,1:30), 'r'), hold,  
    plot(z(1,1:30), 'g'), hold, grid,  
    title('Процессы y и z(после медиан. фильтра')  
subplot(223), hist(y), grid,  
    title('Гистограмма плотн. распр. y'),  
subplot(224), hist(z), grid,  
    title('Гистограмма плотн. распр. z')  
end
```

## Литература

1. Мироновский Л.А. Функциональная диагностика динамических систем. М.: 1998.
2. Мозгалевский А.В., Калявин В.П. Системы диагностирования судового оборудования. Л.: Судостроение, 1987. 221с.
3. Евланов А.Г. Контроль динамических систем. М.: Наука, 1979. 431с.
4. Левин Б.Р. Теоретические основы статистической радиотехники. М.: Советское радио, 1968. 502с.
5. Вапник В.Н., Червоненкис А.Я.. Теория распознавания образов. М.: Наука, 1974. 416с.
6. Никифоров И.В. Последовательное обнаружение свойств временных рядов. М.: Наука, 1983. 199с.
7. Мироновский Л.А. Инварианты математических моделей. Санкт-Петербург.: ЛИАП, 1991. 41с.
8. Игнатьев М.Б., Мироновский Л.А., Юдович В.С. Контроль и диагностика робототехнических систем. Л.: ЛИАП, 1985. 155с.
9. Плескунин В.И. Теоретические основы анализа выборочных данных в эксперименте. Л.: ЛГУ., 1979. 230с.
10. Гром Д. Методы идентификации систем. М.: Мир, 1979. 302с.
11. Корноушенко Е.К., Пылаев Н.К. Передаточные числа и диагностирование линейных систем. Докл. АН СССР, 1988. т.300, №3.
12. Гришин Ю.П. Казаринов Ю.М. Динамические системы, устойчивые к отказам. М.: Радио и связь, 1985. 173с.
13. Сейдж Э., Мэлс Дж. Теория оценивания и ее применение в связи и управлении. М.: Связь, 1976. 494с.
14. Адаптивные фильтры, Под ред. К.Ф. Коуэна. М.: Мир, 1988. 388с.
15. Меньшиков В.А., Пчелинцев Л.А., Якушин Н.И. Перспективные системы и методы диагностики. Л.: ВИККИ, 1992.110с.
16. Кудрявцев В.В., Белозеров В.А. Достоверность диагностирования технического состояния сложных систем. // Приборостроение, 1997. № 8. с. 38-48.

17. Клигенс Н.И., Тельскнис Л.Л. Методы обнаружения моментов изменения свойств случайных процессов. // Автоматика и телемеханика, 1983, с. 5-56.
18. Бородкин Л.И., Моттль В.В. Алгоритм обнаружения моментов изменения свойств параметров уравнения случайного процесса. // Автоматика и телемеханика, 1976. № 6. с. 23-33.
19. Калишев О.Н. Метод диагностирования измерительных каналов с учетом предыстории. // Автоматика и телемеханика, 1988. № 6. с. 135-144.
20. Кемени Дж., Снелл Дж. Конечные цепи Маркова. М.: Наука, 1970. 271с.
21. Гаджиев Ч.М. Прогнозирование технического состояния динамических систем по обновляющей последовательности фильтра Калмана. // Автоматика и телемеханика, 1993. №5. с. 183-167.
22. Казанцев В.С. Задачи классификации и их программное обеспечение. М.: Наука, 1990, 133с.
23. Горбань А.Н., Дунин-Барковский В.Л., Кардин А.Н. Нейроинформатика. Новосибирск: Наука, 1998.
24. Галушкин А.И. Синтез многослойных систем распознавания образов, М. Наука, 1974.
25. Горбань А.Н. Обучение нейронных сетей СП. ПараГраф. 1990.
26. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск.: Наука, 1996.
27. Howard Demuth, Mark Beale. 'Neural Network Toolbox for use with MATLAB' – The Math Works. Inc., 1997.
28. Колесников Д.Н., Сиднев А.Г., Юрганов А.А. Моделирование случайных факторов в задачах автоматики и вычислительной техники. Санкт-Петербург: СПбГТУ, 1994. 104с.
29. Бакут П.А. и др. Обнаружение движущихся объектов. М.: Советское радио, 1980. 287с.
30. Оуэн Д.Б. Сборник статистических таблиц. М.: Выч. Центр АН СССР, 1966 г.
31. Автоматизация измерений и обработки данных на прочность. М.: Машиностроение, 1991. 207с.
32. Колесников Д.Н., Душутина Е.В., Пахомова В.И. Введение в MATLAB с примерами решения задач оптимизации и моделирования. Санкт-Петербург: СПбГТУ, 1995. 110с.

Бендерская Е.Н.  
Колесников Д.Н.  
Пахомова В.И.

## **Функциональная диагностика систем управления**

Учебное пособие

Оригинал-макет подготовлен авторами

---

Подписано в печать 30.10.2000г Объем 9,0 усл. п. л.  
Тираж 100 экз. Заказ № 47

---

Отпечатано в секторе оперативной полиграфии Инновационно-  
инвестиционного комплекса СПбГТУ

195251, Санкт-Петербург, Политехническая ул., 29

