

**С.В. Шидловский**

**АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ.  
ПЕРЕСТРАИВАЕМЫЕ СТРУКТУРЫ**

Томск  
2006

УДК 681.5  
ББК 32.965; 32.815  
Ш 564



*Издание осуществлено при поддержке  
Российского фонда фундаментальных  
исследований по проекту 06-08-06040*

**Шидловский С.В.**

**Ш 564** Автоматическое управление. Перестраиваемые структуры. – Томск:  
Томский государственный университет, 2006. – 288 с.

**ISBN 5–94621–186–2**

Излагаются алгоритмы и способы создания управляющих систем, построенных на базе логических структур и обладающих высокой надежностью за счет изменения внутренней структуры.

На основе многофункциональных логических модулей решены проблемы вычисления систем булевых формул определенных классов, отражены вопросы автоматизированного синтеза и исследования функциональных свойств многофункциональных логических модулей и построенных на их основе изотропных и квазиизотропных сред с помощью разработанной логической системы имитационного моделирования Cell System. Подробно освещен вопрос об использовании логических устройств с перестраиваемой структурой в задачах автоматического регулирования.

Для специалистов в области систем управления и автоматизации технологических процессов и производств, может быть полезна для студентов, аспирантов, инженеров и научных работников, занимающихся вопросами автоматического управления.

**УДК 681.5**  
**ББК 32.965; 32.815**

**Рецензенты:** доктор технических наук, профессор, академик РАЕН  
**В.И. Сырымкин** (Томский государственный университет);

доктор технических наук, профессор, заслуженный  
деятель науки РФ, академик МАНВШ **В.С. Титов** (зав. кафедрой вычислительной техники Курского государственного  
технического университета)

**ISBN 5–94621–186–2**

© С.В. Шидловский, 2006

**S.V. Shidlovskiy**

**AUTOMATED CONTROL.  
RECONFIGURABLE STRUCTURES**

Tomsk  
2006



*This edition is realized with the support  
of Russian foundation for basic research  
on the project 06-08-06040*

**Shidlovskiy S.V.**

Automated control. Reconfigurable structures. – Tomsk: Tomsk state university, 2006. – 288 p.

**ISBN 5–94621–186–2**

This work describes algorithms and ways of creation of control systems constructed on the basis of logic structures. Their high reliability is provided with the changes in their internal structure.

On the basis of multifunctional logic modules the problems of calculation Boolean formulas systems of the certain classes are solved. The questions of the automated synthesis and investigation of functional properties of multifunctional logic modules and isotropic and quasiisotropic environments constructed by means of the developed logic system of imitating modelling Cell System. The question of using the logic devices with reconfigurable structure in problems of automatic control is covered in detail.

For experts in the field of control systems and automation of technological process and production and can be useful for students, post-graduate students, engineers and scientists engaged in questions of automatic control.

**Reviewers:** **V.I. Siryamkin**, PhD technical science, professor, academician of Russian Academy of Natural Science (Tomsk state university);

**V.S. Titov**, PhD technical science, professor, the «honored scientific researcher» of the Russian Federation, academician IHEAS (the head of the chair of computer science of Kursk state technical university)

ISBN 5–94621–186–2

© S.V. Shidlovskiy, 2006

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ</b> .....	13
<b>ГЛАВА 1. ПЕРЕСТРАИВАЕМЫЕ АВТОМАТЫ И ОДНОРОДНЫЕ СТРУКТУРЫ ДЛЯ ПОСТРОЕНИЯ УПРАВЛЯЮЩИХ УСТРОЙСТВ</b> .....	19
1.1. Вводные понятия .....	19
1.2. Способы задания функций алгебры логики .....	20
1.3. Классификация автоматов .....	20
1.4. Перестраиваемые автоматы .....	22
1.5. Однородные структуры .....	24
1.5.1. Синхронные и асинхронные структуры .....	26
1.5.2. Структуры с индивидуальным и коллективным выбором направления передачи сигналов .....	27
1.5.3. Направленность передачи сигнала и связи ячейки .....	28
1.5.4. Настройка структуры .....	28
1.5.5. Функциональная способность ячейки .....	30
1.5.6. Степень универсальности структур .....	30
1.6. Задание алгоритмов функционирования и переработки информации в автоматических системах управления .....	31
1.7. Аппаратная и программная реализация алгоритмов функционирования управляющих устройств .....	34
1.8. Выводы .....	37
Литература .....	37
<b>ГЛАВА 2. БУЛЕВА МОДЕЛЬ ЛОГИКИ ПЕРЕСТРАИВАЕМЫХ СТРУКТУР</b> .....	39
2.1. Вводные понятия .....	39
2.2. Классификация булевых формул .....	40
2.3. Булева модель логики перестраиваемых структур для определенных классов булевых формул .....	43
2.4. Многофункциональные логические модули. Основные понятия и определения .....	45
2.5. Вычисление булевых формул .....	49
2.5.1. Вычисление неповторных ДНФ и КНФ булевых формул .....	49

2.5.2. Вычисление неповторных булевых формул с пропусками аргументов .....	59
2.5.3. Вычисление неповторных упорядоченных булевых формул выше второго порядка.....	74
2.5.4. Вычисление неупорядоченных булевых формул.....	76
2.5.5. Метод декомпозиции для вычисления произвольных булевых формул .....	80
2.5.6. Однотактное вычисление неупорядоченных булевых формул.....	83
2.5.7. Вычисление систем булевых формул из классов неповторных упорядоченных и неупорядоченных формул.....	87
2.5.8. Вычисление повторных упорядоченных произвольных нормальных булевых формул из $h$ букв и систем булевых формул как с пропусками аргументов, так и без них .....	91
2.6. Выводы.....	98
Литература.....	98
<b>ГЛАВА 3. РЕАЛИЗАЦИЯ БУЛЕВОЙ МОДЕЛИ ЛОГИКИ ПЕРЕСТРАИВАЕМЫХ СТРУКТУР С ПРИМЕНЕНИЕМ ИЗОТРОПНЫХ СРЕД.....</b>	
3.1. Вводные понятия.....	101
3.2. Имитационная система Cell System.....	102
3.3. Имитационное моделирование изотропных сред .....	107
3.4. Синтез линейных изотропных сред.....	112
3.5. Декомпозиция линейных изотропных сред.....	117
3.6. Выводы.....	122
Литература.....	123
<b>ГЛАВА 4. ДИНАМИКА СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ.....</b>	
4.1. Вводные понятия.....	125
4.2. Основные принципы построения адаптивных систем управления.....	128
4.3. Технологические процессы как объекты управления.....	130
4.4. Математическое описание систем автоматического управления технологическими процессами и понятие фазового пространства.....	134

4.5. Классификация промышленных объектов управления.....	140
4.6. Идентификация объекта управления .....	143
4.7. Понятие систем с переменной структурой.....	150
4.8. Метод фазового пространства .....	154
4.9. Типы движения в системах с переменной структурой.....	158
4.10. Выводы .....	162
Литература.....	163
<b>ГЛАВА 5. СИНТЕЗ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ С ПЕРЕСТРАИВАЕМОЙ СТРУКТУРОЙ.....</b>	<b>165</b>
5.1. Постановка задачи управления.....	165
5.2. Типовая система регулирования .....	172
5.3. Адаптивная система автоматического регулирования.....	176
5.4. Системы автоматического регулирования со структурной адаптацией.....	186
5.5. Системы автоматического регулирования с перестраиваемой структурой .....	190
5.5.1. Формирование логического закона управления .....	190
5.5.2. Пример синтеза системы без запаздывания в контуре управления .....	195
5.5.3. Пример синтеза системы с запаздыванием в контуре управления .....	198
5.6. Интегральный регулятор с перестраиваемой структурой.....	200
5.7. Интегральный дискретный регулятор с перестраиваемой структурой.....	204
5.8. Регулятор качества переходного процесса.....	208
5.9. Выводы .....	214
Литература.....	215
<b>ГЛАВА 6. АСПЕКТЫ ТЕОРИИ НЕЧЕТКИХ МНОЖЕСТВ В ЗАДАЧАХ КАЧЕСТВЕННОГО АНАЛИЗА ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ.....</b>	<b>217</b>
6.1. Вводные понятия .....	217
6.2. Некоторые основные определения теории нечетких множеств.....	218
6.3. Нечеткая логика и приближенные выводы .....	220
6.4. Регулятор качества.....	227

6.5. Выводы.....	235
Литература.....	235
<b>ГЛАВА 7. ИНФОРМАЦИОННО-ПОИСКОВЫЕ АСПЕКТЫ ПРИ СИНТЕЗЕ СТРУКТУРНОГО АВТОМАТА .....</b>	<b>237</b>
7.1. Вводные понятия.....	237
7.2. Режимы поиска.....	237
7.3. Анализ критериев выдачи информационно-поисковых систем.....	238
7.4. Динамические режимы работы информационно- поисковых систем .....	240
7.5. Информационно-поисковый автомат.....	241
7.6. Выводы.....	249
Литература.....	250
<b>ГЛАВА 8. КОРПОРАТИВНЫЙ ПОРТАЛ НА ОСНОВЕ ПРИНЦИПА ПЕРЕСТРАИВАЕМЫХ СТРУКТУР .....</b>	<b>251</b>
8.1. Вводные понятия.....	251
8.2. XML как инструментарий создания гипермоделей однородных структур .....	252
8.3. Сходство процессов в системах автоматического управления и обучения .....	256
8.4. Обобщенная структура портала.....	261
8.5. Интеграция распределенных Web-серверов .....	266
8.6. Выводы.....	268
Литература.....	269
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>271</b>
<b>Приложение 1. Таблицы для формул S-структуры     и <math>\varphi</math>-импликант.....</b>	<b>273</b>
<b>Приложение 2. Структурная схема алгоритма, текст программы     и инструкция пользователя для расчета     оптимальных параметров ПИ-регулятора .....</b>	<b>276</b>
<b>Приложение 3. Структурная схема алгоритма, текст программы     и инструкция пользователя интегрального     регулятора с перестраиваемой структурой .....</b>	<b>280</b>
<b>Список основных сокращений.....</b>	<b>285</b>



## CONTENTS

<b>INTRODUCTION .....</b>	<b>13</b>
<b>CHAPTER 1. RECONFIGURABLE AUTOMATIC DEVICES AND HOMOGENEOUS STRUCTURES FOR CONSTRUCTION OF CONTROL DEVICES.....</b>	<b>19</b>
1.1. Introduction.....	19
1.2. Ways of specifying of functions of algebra of logic.....	20
1.3. Classification of automatic devices .....	20
1.4. Reconfigurable automatic devices .....	22
1.5. Homogeneous structures.....	24
1.5.1. Synchronous and asynchronous structures .....	26
1.5.2. Structures with individual and collective choice of the direction of signal transmission .....	27
1.5.3. The direction of signal transmission and cell communication.....	28
1.5.4. Adjustment of structure .....	28
1.5.5. Functional cell abilities.....	30
1.5.6. Degree of structure universality.....	30
1.6. Prescription of algorithms of functioning and processing of information in automated control systems.....	31
1.7. Hardware and software realization of algorithms of functioning control devices .....	34
1.8. Conclusion .....	37
Literature.....	37
<b>CHAPTER 2. BOOLEAN MODEL OF LOGIC OF RECONFIGURABLE STRUCTURES.....</b>	<b>39</b>
2.1. Introduction.....	39
2.2. Boolean formulas classification.....	40
2.3. Logic reconfigurable structures for the certain classes of Boolean formulas.....	43
2.4. Multifunctional logic modules. The basic concepts and definitions .....	45
2.5. Boolean formulas calculation .....	49

2.5.1. Repetition-free DNF and CNF Boolean formulas calculation .....	49
2.5.2. Repetition-free Boolean formulas calculation with missing arguments.....	59
2.5.3. Repetition-free Boolean formulas calculation higher then the second order.....	74
2.5.4. Disordered Boolean formulas calculation .....	76
2.5.5. Decomposition method for random Boolean formulas calculation .....	80
2.5.6. Single-cycle calculation of disordered Boolean formulas.....	83
2.5.7. Boolean formula systems calculation from repetition-free ordered and disordered formula classes.....	87
2.5.8. Repeated ordered random normal Boolean formulas calculation consisting of $h$ letters and Boolean formula systems with or without missing arguments.....	91
2.6. Conclusion .....	98
Literature.....	98
<b>CHAPTER 3. BOOLEAN MODEL REALIZATION OF RECONFIGURABLE STRUCTURES LOGIC WITH APPLICATION OF ISOTROPIC ENVIRONMENTS .....</b>	<b>101</b>
3.1. Introduction.....	101
3.2. Imitating system Cell System.....	102
3.3. Isotropic environments imitating modelling .....	107
3.4. Linear isotropic environments synthesis .....	112
3.5. Linear isotropic environments decomposition .....	117
3.6. Conclusion .....	122
Literature.....	123
<b>CHAPTER 4. DYNAMICS OF CONTROL SYSTEMS .....</b>	<b>125</b>
4.1. Introduction.....	125
4.2. The main principles of adaptive control systems construction ....	128
4.3. Technological process as objects of control.....	130
4.4. The mathematical description of control systems of technological process and concept of phase space.....	134
4.5. Classification of industrial objects of control.....	140
4.6. Identification of the object of control.....	143
4.7. Concept of systems with variable structure.....	150

4.8. Phase space method .....	154
4.9. Types of movement in systems with variable structure .....	158
4.10. Conclusion .....	162
Literature.....	163
<b>CHAPTER 5. SYNTHESIS OF CONTROL SYSTEMS WITH RECONFIGURABLE STRUCTURE .....</b>	<b>165</b>
5.1. Setting of control target problem .....	165
5.2. Typical regulation system .....	172
5.3. Adaptive system of automated control .....	176
5.4. Control systems with structural adaptation .....	186
5.5. Control systems with reconfigurable structure .....	190
5.5.1. Formation of the logic law of control .....	190
5.5.2. Example of synthesis of system without delay in a control contour .....	195
5.5.3. Example of synthesis of system with delay in a control contour .....	198
5.6. An integrated regulator with reconfigurable structure .....	200
5.7. An integrated discrete regulator with reconfigurable structure.....	204
5.8. Transient quality regulator .....	208
5.9. Conclusion .....	214
Literature.....	215
<b>CHAPTER 6. ASPECTS OF THE FUZZY SETS THEORY IN PROBLEMS OF THE QUALITATIVE ANALYSIS OF TECHNOLOGICAL PROCESS .....</b>	<b>217</b>
6.1. Introduction.....	217
6.2. Some basic definitions of fuzzy sets theory.....	218
6.3. Fuzzy logic and the approximate conclusions .....	220
6.4. Quality regulator .....	227
6.5. Conclusion .....	235
Literature.....	235
<b>CHAPTER 7. SYNTHESIS INFORMATION RETRIEVAL ASPECTS IN THE STRUCTURAL AUTOMATIC DEVICE .....</b>	<b>237</b>
7.1. Introduction.....	237
7.2. Search modes .....	237
7.3 Criteria analysis of information retrieval systems delivery .....	238
7.4. Dynamic operating modes of information retrieval systems .....	240

---

7.5. Information retrieval automatic device .....	241
7.6. Conclusion .....	249
Literature .....	250
<b>CHAPTER 8. THE CORPORATE PORTAL ON THE BASIS OF RECONFIGURABLE STRUCTURES</b>	
<b>PRINCIPLE</b> .....	251
8.1. Introduction .....	251
8.2. XML as a toolkit for creation of homogeneous structures hypermodels .....	252
8.3. The similarity of the process in automated control and training systems .....	256
8.4. Generalized structure of the portal .....	261
8.5. Integration of the distributed Web-servers .....	266
8.6. Conclusion .....	268
Literature .....	269
<b>THE CONCLUSION</b> .....	271
<b>The appendix 1</b> .....	273
<b>The appendix 2</b> .....	276
<b>The appendix 3</b> .....	280
<b>Abridgements</b> .....	285

---

---

## ВВЕДЕНИЕ

---

---

В книге рассматривается применение автоматного принципа обработки информации при построении цифровых управляющих устройств и их внутренней структуры для систем управления различного назначения.

Основные теоретические исследования в этой области были проведены М.А. Гавриловым, В.М. Глушковым, В.А. Горбатовым, С.В. Емельяновым, А.А. Красовским, Б.Н. Петровым, Я.З. Цыпкиным и др.

К обозначенной тематике автор был привлечен со студенческой поры в Томском политехническом университете (2000–2002 гг.). Нарботанные за годы учебы результаты вылились в дипломную работу и послужили хорошей научной базой для дальнейшей творческой работы в рамках аспирантуры (2002–2004 гг.).

Интенсификация современного производства и ускорение научно-технического прогресса в значительной мере определяются степенью автоматизации разнообразных технологических и производственных процессов на базе широкого применения новейших средств вычислительной техники и уровнем подготовки инженерных кадров страны.

Технологическая база современного производства в большинстве отраслей промышленности достигла такого уровня развития, при котором качество управления технологическим процессом является существенным фактором эффективности производства в целом. В этой связи первостепенное значение приобретает задача оптимального управления технологическими процессами, решение которой возможно только при развитой автоматизированной системе управления технологическими

процессами (АСУ ТП). Этапы же развития АСУ ТП, как правило, связаны с появлением новых технических средств.

При управлении современным промышленным объектом к нему надо подходить как к единому целому, а не набору различных независимых элементов.

Характерными тенденциями формирования промышленных объектов и систем управления являются:

- 1) возрастание единичной производительности агрегатов;
- 2) рост необходимой «мощности» применяемых систем контроля и управления;
- 3) повышение требований к качеству технологических процессов из-за переоценки взглядов на значение энергетических ресурсов, экономии топлива, роль человека в производстве и защиту окружающей среды;
- 4) вовлечение все новых и новых агрегатов и участков в сферу действия управления в связи с повышением степени автоматизации производства.

В подобных обстоятельствах возникла проблема автоматизации процесса принятия решения, которое потребовало привлечения современных математических методов и новых технических средств.

В настоящее время при построении различного рода управляющих устройств широко используются булевы модели, однако некоторые разработчики избегают прямого применения функций алгебры логики. Объясняется это тем, что разработка эффективных алгоритмов вычисления сложных функций алгебры логики многих аргументов (десятки, сотни) сопряжена со значительными трудностями и представляет собой самостоятельную задачу, требующую глубоких исследований для ее решения.

Растущий интерес к функциям алгебры логики и проблеме их вычисления привел к созданию теории однородных структур. С практической точки зрения однородные (изотропные) среды как специализированные устройства, ориентированные на вычисления функций алгебры логики, обладают следующими достоинствами:

1. Применение изотропных сред в специализированных управляющих устройствах для реализации законов управления, представленных функцией алгебры логики, позволяет достичь такого быстродействия, когда нахождение значения функции алгебры логики на заданном наборе значений аргумента осуществляется за один период тактового генератора независимо от ее сложности и числа аргументов.

2. Имитационное моделирование изотропной среды с фиксированным кодом настройки обеспечивает временные затраты на вычисление функции алгебры логики, пропорциональные числу аргументов независимо от ее сложности.

3. Использование изотропной среды в специализированном многотактном автомате, реализующем в каждый период тактового генератора одну или несколько ячеек среды, обеспечивает заданную продолжительность вычисления функции алгебры логики независимо от ее сложности.

Таким образом, теория изотропных сред в принципе может быть использована при создании специализированных управляющих устройств. Однако успешному практическому применению этой теории препятствуют ориентация ее на свободное использование как информационных, так и настроечных входов, в зависимости от вида реализуемых булевых формул, и необходимость применения коммутирующих сред для установления соответствия между логическими аргументами и ячейками изотропной среды, в то время как требуются среды, обеспечивающие реализацию заданных классов булевых формул, где каждой ячейке среды поставлены в постоянное соответствие вполне определенные логические аргументы.

Глава 1 настоящей работы содержит основные понятия и определения, необходимые при рассмотрении перестраиваемых структур.

В главе 2 предложена булева модель логики перестраиваемых структур для определенного класса булевых формул, а также классификация булевых формул. Решаются проблемы вычисления систем булевых формул из классов неповторных упорядоченных и неупорядоченных, а также класса повторных упорядоченных произвольных нормаль-

ных булевых формул из  $h$  букв и систем булевых формул как с пропусками аргументов, так и без них.

Глава 3 посвящена теоретическим и практическим вопросам построения изотропных сред. Описывается созданная логическая система имитационного моделирования Cell System для рассматриваемого класса многофункциональных логических модулей, позволяющая синтезировать и исследовать многофункциональные логические модули, построенные на их основе изотропные и квазиизотропные среды и их взаимодействие с системами, имеющими практическую направленность в той или иной области науки и техники.

В главе 4 освещены вопросы математического описания динамики систем автоматического управления, отражены методы их получения по экспериментальным характеристикам, приведена классификация промышленных объектов управления.

Глава 5 содержит сведения о практическом применении булевой модели логики перестраиваемых структур, в том числе с использованием изотропных сред, для систем автоматического управления технологическими процессами.

В главе 6 рассматривается применение теории нечетких множеств для качественного анализа процессов, протекающих в системах автоматического регулирования; приведен вариант регулятора качества переходного процесса, функционирующего на базе теории нечетких множеств.

Глава 7 посвящена практическим вопросам применения перестраиваемых структур в области построения информационно-поисковых систем.

В главе 8 отражены основные предпосылки к формированию и созданию корпоративных порталов, ориентированных на повышение качества обучения, способных функционировать в гиперпространстве и обеспечивать качественный автоматический контроль знаний обучаемого. Выявлены сходства между процессами в системах автоматического управления и обучения, предложены методы оценки качества обучения и применения их в качестве составного модуля корпоративного портала.



Основные результаты получены в процессе работы на кафедре информационно-измерительной техники Томского государственного университета систем управления и радиоэлектроники под непосредственным руководством доктора технических наук, профессора А.А. Светлакова, которому автор искренне признателен за постоянное внимание к своей научной деятельности.

Базовые знания автор получил на кафедре автоматизации теплоэнергетических процессов Томского политехнического университета, за что искренне признателен ее преподавательскому составу и персонально В.С. Адныку, А.В. Волошенко, А.А. Татарникову за благожелательный интерес к настоящим исследованиям.

Основное влияние на научное мировоззрение автора оказали выдающиеся труды российских ученых Э.В. Евреинова, С.В. Емельянова, А.В. Каляева, С.К. Коровина, В.Б. Кудрявцева, И.В. Прангишвили, Е.И. Пупырева, В.И. Уткина, Я.И. Фета, А.А. Шалыто, Э.А. Якубайтиса.

Особо признателен за оказанное внимание к работам автора заведующему кафедрой «Технологии программирования» Санкт-Петербургского государственного университета информационных технологий, механики и оптики профессору А.А. Шалыто. Открытость его научных исследований, огромный педагогический опыт, несомненный величайший вклад в науку всегда служили и служат неисчерпаемым источником познания и вдохновения для творческой деятельности автора. Монографии и научные материалы А.А. Шалыто являются настольными наставлениями и основой для научных исследований автора.

Автор искренне признателен коллективу кафедры автоматики Новосибирского государственного технического университета в лице А.С. Вострикова, А.А. Воеводы, В.Д. Юркевича, О.Я. Шпилиной за ряд ценных замечаний, способствующих качественному улучшению научного материала автора и особо Г.А. Французовой за создание совместного информационного творческого поля.

Автор благодарен А.М. Корикову, А.И. Рубану, В.А. Соловьеву, О.М. Раводину, Ю.А. Андрееву за постоянное внимание к его научным исследованиям.

Автор хотел бы выразить благодарность профессору Ю.П. Шевелеву, его значительный опыт и ценные комментарии оказались весьма полезными.

Автор считает приятным долгом выразить искреннюю благодарность руководству Томского государственного университета систем управления и радиоэлектроники за систематическую помощь и поддержку.

Автор признателен всем сотрудникам Фонда содействия науке и образованию – Томского регионального инкубатора технологий (ФСНО – ТРИТ), способствующим появлению данной работы.

Работа выполнена в рамках программы «Ползуновские гранты» (2005 г.) Фонда содействия развитию малых форм предприятий в научно-технической сфере и при поддержке Государственного контракта с Федеральным агентством по науке и инновациям №02.442.11.7498.

Конечно монография не лишена недостатков, ответственность за которые несет только автор. Все полезные замечания и предложения, улучшающие техническую сторону монографии, будут приняты автором с благодарностью и учтены в последующих изданиях.

С автором можно связаться по адресу [stas@iit.tusur.ru](mailto:stas@iit.tusur.ru)

# ПЕРЕСТРАИВАЕМЫЕ АВТОМАТЫ И ОДНОРОДНЫЕ СТРУКТУРЫ ДЛЯ ПОСТРОЕНИЯ УПРАВЛЯЮЩИХ УСТРОЙСТВ

---

---

## 1.1. ВВОДНЫЕ ПОНЯТИЯ

В настоящее время существуют различные подходы к определению понятия конечного автомата, которые могут быть разбиты на группы *макроподхода* и *микроподхода*. При *макроподходе* интересуются внешним поведением устройства, тем, как оно осуществляет переработку входной информации в выходную информацию и последовательность состояний, отвлекаясь от внутреннего его состояния. На этом пути приходят к *понятию абстрактного конечного автомата*. Тем самым абстрактный конечный автомат может быть задан с помощью набора отображений, описывающих его «внешнее» функционирование. При *микроподходе* учитывается структура устройства, функционирование и связь между собой его частей. На этом пути приходят к *понятию структурного конечного автомата*, называемого также автоматной схемой, логической сетью либо многофункциональным логическим модулем (МЛМ). Структурный конечный автомат задается конечным множеством абстрактных автоматов, конечной схемой их соединения и указанием влияния частей схемы друг на друга. Понятия конечного абстрактного и конечного структурного автоматов можно считать составляющими понятия конечного автомата.

Обобщение понятия конечного автомата получается путем обобщения понятий конечных абстрактного и структурного автоматов.

Абстрактный автомат получается в результате выбора произвольных (необязательно конечных) множеств входных и выходных сигналов, множеств состояний, а также при расширении понятия зависимости состояния и выходного сигнала от входного сигнала и состояния. Структурный автомат получается в результате выбора произвольных множеств автоматов и схем их соединений.

## 1.2. СПОСОБЫ ЗАДАНИЯ ФУНКЦИЙ АЛГЕБРЫ ЛОГИКИ

Функция алгебры логики  $f(x_1, \dots, x_n)$  полностью определяется заданием ее значений на всех наборах аргументов. Область определения в любой функции алгебры логики конечна, поскольку число аргументов и число значений каждого аргумента также конечны. Функции алгебры логики могут задаваться рядом способов [6]:

- 1) табличным, когда функция задается в виде таблицы истинности;
- 2) графическим, когда функция задается в виде  $n$ -мерного единичного куба;
- 3) координатным, когда функция задается в виде координатной карты состояний (карты Карно);
- 4) числовым, когда функция задается в виде десятичных или восьмеричных эквивалентов наборов тех аргументов, на которых она принимает значение единицы;
- 5) аналитическим, когда функция задается в виде аналитического выражения, получаемого путем совершения каких-либо логических операций с переменными алгебры логики, т.е. в виде формул, например:

$$f = x_1 \bar{x}_2 \vee x_3.$$

Здесь и далее символ  $\vee$  обозначает операцию «дизъюнкция», а операцию «конъюнкция», которая обозначается символом  $\&$ , будем заменять точкой или опускать какой-либо знак между аргументами.

## 1.3. КЛАССИФИКАЦИЯ АВТОМАТОВ

По количеству выполняемых преобразований информации все множество автоматов можно разделить на два класса: неперестраиваемые, или монофункциональные, автоматы; перестраиваемые, или многофункциональные, автоматы (M-автоматы).

Монофункциональный автомат имеет жесткую структуру и всегда реализует одно и то же преобразование  $\{X\} \rightarrow \{Y\}$ . Для такого автомата величина его функциональности  $L = 1$ .

К M-автоматам относятся все управляемые автоматы, реализующие путем их настройки некоторое множество  $D_A = \{A_i\}, i = \overline{1, L}$ , автоматных преобразований  $\{X_i\} \rightarrow \{Y_i\}$ , где  $L > 1$ .

Если функции переходов  $\delta$  и выходов  $\lambda$  задаются на конечных множествах  $D_\delta \subseteq \{X\} \times \{Z\}$  и  $D_\lambda \subseteq \{X\} \times \{Z\}$  и М-автомат реализует любые автоматные преобразования (путем его настройки), определяемые парами функций  $\delta^{(i)}$  и  $\lambda^{(i)}$  и задаваемые на конечных множествах  $D_\delta$  и  $D_\lambda$ , то такой автомат будем называть универсальным, или У-автоматом.

Выполнение в М-автомате требуемого автоматного преобразования  $A_i \in D_A$  осуществляется соответствующей его настройкой. По принципу настройки М-автоматы делятся на три класса: с функциональной, структурной и программной настройкой.

Под *функциональной* настройкой М-автомата будем понимать такую, при которой код настройки  $Z_i$  не изменяется в течение всего времени выполнения преобразования  $A_i$ . При этом связи между функциональными элементами структурной схемы М-автомата могут оставаться прежними или изменяться в зависимости от реализуемого преобразования. В последнем случае изменение связей в структурной схеме М-автомата по существу эквивалентно преобразованию самой его структуры. Настройку автомата, в результате которой его структурная схема претерпевает соответствующие изменения (за счет изменения связей), будем называть *структурной* настройкой.

В *программно* настраиваемом автомате преобразования  $A_i$  выполняются за некоторое число  $n_{t_i}$  шагов, при этом код настройки на каждом шаге преобразования меняется.

Функциональная настройка является наиболее простым видом настройки и характерна для простейших М-автоматов (Мили, Мура, тривиальных и др.). Структурная настройка является более сложной и характерна для автоматов с более сложной структурной организацией, представляющей собой композицию некоторых функциональных элементов (автоматов) и элементов (автоматов) коммутации или связи. Код настройки такого М-автомата состоит из двух частей: кодов настроек функциональных автоматов и кодов настроек коммутационных автоматов. При этом сами функциональные и коммутационные автоматы могут быть функционально или структурно перестраиваемыми.

#### 1.4. ПЕРЕСТРАИВАЕМЫЕ АВТОМАТЫ

Под перестраиваемым автоматом будем понимать модель широкого класса дискретных устройств, называемых иногда многофункциональными и универсальными логическими модулями, настраиваемыми логическими устройствами и т.п. [8].

Конечные детерминированные автоматы, называемые ниже просто автоматами, представляют собой объекты, имеющие конечное число внутренних состояний  $c_k$  ( $k = 1, \dots, q$ ), конечное число входных сигналов  $x_i$  ( $i = 1, \dots, n$ ) и конечное число выходных сигналов  $f_j$  ( $j = 1, \dots, p$ ).

Автомат работает в дискретном времени  $t_v$  ( $v = 1, 2, \dots$ ) и в каждый момент времени может находиться только в одном состоянии [8].

Пусть существует конечное число так называемых внутренних сигналов  $y_r$  ( $r = 1, \dots, m$ ); сопоставим с каждым состоянием  $c_k$  набор значений сигналов  $y_r$  так, чтобы различным состояниям соответствовали различные наборы значений  $y_r$ .

Тогда в каждый момент времени значения внутренних и выходных сигналов можно задать как функции от входных и внутренних сигналов:

$$\begin{cases} y_1(v) = \alpha_1(x_1(v), \dots, x_n(v), y_1(v-1), \dots, y_m(v-1)), \\ \dots \\ y_m(v) = \alpha_m(x_1(v), \dots, x_n(v), y_1(v-1), \dots, y_m(v-1)); \end{cases} \quad (1.1)$$

$$\begin{cases} f_1(v) = \varphi_1(x_1(v), \dots, x_n(v), y_1(v-1), \dots, y_m(v-1)), \\ \dots \\ f_p(v) = \varphi_p(x_1(v), \dots, x_n(v), y_1(v-1), \dots, y_m(v-1)). \end{cases} \quad (1.2)$$

Описываемая модель автомата называется *структурным автоматом*, и, соответственно, системы (1.1) и (1.2) называются структурными функциями переходов и выходов.

Рассмотрим, например, комбинационный автомат, представленный на рис. 1.1. Обозначим сигналы, подаваемые на входы 1, 2, 3, 4, соответственно переменными  $x_1$ ,  $x_2$ ,  $z_1$ ,  $z_2$ . При этом если сигнал на некотором входе равен 1, то значение соответствующей переменной равно 1, а если сигнал равен 0, то и значение соответствующей

переменной равно 0. Тогда функция  $f(x_1, x_2, z_1, z_2)$ , реализуемая на выходе автомата, имеет вид

$$f = \bar{x}_2 \bar{z}_1 z_2 \vee x_1 x_2 z_1 \vee \bar{x}_1 z_1 z_2.$$

Пусть в некоторый момент времени значения переменных  $z_1$  и  $z_2$  фиксированы и не изменяются. Нас интересует функция от переменных  $x_1$  и  $x_2$ , реализуемая на выходе автомата. Тогда автомат может реализовывать любую из четырех формул от двух переменных  $x_1$  и  $x_2$ :

при  $z_1 = 0, z_2 = 0$

$$f = 0; \quad (1.3)$$

при  $z_1 = 0, z_2 = 1$

$$f = \bar{x}_2; \quad (1.4)$$

при  $z_1 = 1, z_2 = 0$

$$f = x_1 x_2; \quad (1.5)$$

при  $z_1 = 1, z_2 = 1$

$$f = x_1 x_2 \vee \bar{x}_1. \quad (1.6)$$

Фиксацию сигналов на входах 3, 4 назовем *настройкой*, а сами входы 3, 4 и соответствующие им переменные  $z_1$  и  $z_2$  – *настроечными*. В интегральной технологии различают два этапа настройки: *жесткую* (схемная, технологическая, постоянная) настройку, выполняемую путем травления, выжигания при изготовлении интегральной схемы, и *мягкую* (программная, оперативная, переменная, гибкая), выполняемую многократно в процессе использования схемы. Здесь и в дальнейшем рассматривается настройка второго этапа, называемая просто *настройкой*. Входы 1, 2 назовем *информационными*. Выражения (1.3) – (1.6) задают множество реализуемых автоматом формул и алгоритм настройки на любую из них.

Таким образом, любой автомат путем фиксации значений сигналов на части его входов может реализовывать некоторое множество автоматных отображений наборов значений выходных сигналов, поступающих на остальные входы, в выходные сигналы.

*Перестраиваемым* называется автомат, для которого задано множество реализуемых им автоматных отображений и определен алгоритм настройки на реализацию каждого из этих автоматных отображений [8].

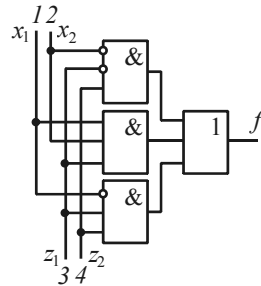


Рис. 1.1. Пример перестраиваемого автомата

Перестраиваемый автомат можно представить как множество автоматов с одними и теми же выходами, причем настройка определяет тот автомат, выходы которого считаются при этой настройке выходами всего перестраиваемого автомата. Поэтому перестраиваемый автомат ничего, кроме автоматных отображений, реализовать не может.

В случае, если перестраиваемый автомат при любых настройках формирует значение сигналов на его выходах, зависящих только от значений сигналов на его входах, он называется *комбинационным перестраиваемым автоматом*, или *многофункциональным логическим модулем*.

Если же для некоторых настроек значения сигналов на выходах автомата зависят от его состояния, то он называется *перестраиваемым автоматом с памятью*.

В основе построения автоматов с перестраиваемой структурой лежат три принципа.

1. *Параллельность*. Повышение быстродействия вычислительных средств за счет повышения тактовой частоты и параллельного (одновременного) выполнения большого числа операций, причем по мере повышения требований к быстродействию вклад от параллельного выполнения операций увеличивается [3].

2. *Перестраиваемость*. Надежность, гибкость и структурная универсальность (возможность создания для каждой задачи соответствующей структуры) управляющих вычислительных средств обеспечиваются схемно за счет программного изменения связей между элементами и автоматных функций самих элементов.

3. *Однородность*. Простота технологии изготовления вычислительных средств обеспечивается благодаря использованию одинаковых элементов и однотипных связей между ними.

## 1.5. ОДНОРОДНЫЕ СТРУКТУРЫ

Однородные структуры относятся к одному из важнейших видов управляющих систем. Они представляют собой дискретную математическую модель широкого класса реальных систем вместе с протекающими в них процессами.

*Однородными структурами* будем называть структуры [7], состоящие из функциональных ячеек и соединений между ними, которые отвечают следующим основным требованиям:



1) структура представляет собой геометрически правильную решетку, имеющую не менее двух осей симметрии, в узлах которой расположены функциональные ячейки (автоматы);

2) все функциональные ячейки структуры однотипны;

3) каждая функциональная ячейка соединена геометрически одинаковым образом с соседними, причем каждую из ячеек можно условно считать центром симметрии относительно связей ее с окружающими ячейками структуры;

4) функциональная ячейка структуры может быть настроена на выполнение в данный момент любой одной функции хотя бы одного полного базиса и функции канала передачи сигналов в заданном направлении, т.е. ячейка должна обладать функциональной и соединительной полнотой;

5) сигнал от любой ячейки  $A$  может быть передан к любой ячейке  $B$  структуры (хотя бы при помощи других ячеек);

6) изотропность структуры, вытекающая из предыдущих требований, позволяет реализовать любую заданную функцию на любом участке или области структуры.

Перечисленным признакам может отвечать большое количество разнообразных структур, отличающихся друг от друга геометрией решетки, функциональной схемой ячейки, количеством и характером связей, способом настройки и т.п.

Если в определенный момент времени настроить функциональные ячейки структуры на внутреннее состояние с учетом их информационных связей, то в однородной среде начнутся изменения состояний ячеек, определяемые законами их функционирования и связями между ними. Таким образом, развитие процесса в однородных структурах осуществляется во времени и пространстве с учетом причинно-следственных связей внутри них и, кроме того, в параллельном режиме.

Структуры, характеризующиеся не всеми указанными признаками, а лишь частью их, будем считать не полностью однородными.

Очевидно, что однородные структуры могут быть условно классифицированы по многим признакам, отражающим их особенность [7].

Так, например, однородные структуры могут быть классифицированы по:

1) метрике пространства: одно-, двух-, и трехмерные;

- 2) временным параметрам изменения состояний структуры: синхронные, асинхронные и смешанные;
- 3) характеру выбора направления передачи сигналов: на структуры с индивидуальным выбором и структуры с коллективным выбором;
- 4) числу возможных направлений передачи сигналов и числу связей с окружающими ячейками: структуры с двумя, тремя, четырьмя и большим числом связей и направлений передачи сигналов;
- 5) способам настройки ячеек структуры: сигналами извне (непосредственный, координатный, пошаговый, адресный способ и др.) или сигналами от соседних ячеек;
- 6) способам подачи входных сигналов и отбора выходных сигналов: по краям или изнутри структуры;
- 7) функциональной способности ячейки структуры (набор функций, реализуемых ячейкой);
- 8) степени универсальности структур: универсальные и специализированные.

### 1.5.1. Синхронные и асинхронные структуры

Однородные структуры называются *синхронными*, если изменения логических уравнений сигналов ячеек синхронизованы общей тактовой частотой. В синхронных структурах можно предсказать состояние всякой ячейки в момент времени, соответствующий любому синхронизирующему импульсу, если известны входные переменные и управляющие сигналы, а также свойства и состояния структуры.

В *асинхронных* структурах, в которых работа ячеек не синхронизована общей тактовой частотой, предсказание состояний ячеек невозможно, поскольку каждая операция осуществляется со скоростью, определяемой параметрами этих ячеек. В таких структурах при завершении операции образуется сигнал, который управляет началом следующей операции.

Асинхронные структуры в общем случае позволяют обеспечить большее быстроедействие реализуемых в них устройств, чем синхронные, за счет более быстрого распространения сигнала, в то время как в синхронных структурах скорость передачи сигнала ограничена тактовой частотой.

В некоторых частных случаях может быть достигнуто большее быстроедействие синхронных структур, чем асинхронных.

Реализация какого-либо устройства в асинхронных структурах в общем случае требует меньшего числа ячеек, чем реализация того же устройства в синхронной структуре.

Наряду чисто с асинхронными и синхронными структурами существуют структуры смешанного типа.

Общеизвестно [7], что реализация комбинационных схем более эффективна в асинхронных структурах, поэтому наиболее перспективными оказываются структуры, в которых при решении комбинационных задач структура выступает как асинхронная, а при решении последовательных задач допускается использование тактовой частоты.

### 1.5.2. Структуры с индивидуальным и коллективным выбором направления передачи сигналов

Реализация в однородных структурах практически любых логических или последовательных схем требует, чтобы любая ячейка была способна выполнять набор функций хотя бы одного полного базиса, причем в каждый момент времени она может выполнять только одну функцию из полного набора (например, ячейка может быть способна выполнять функцию ИЛИ-НЕ или И-НЕ, или полный набор функций «запрет +1» и т.п., причем в последнем случае в любой момент ячейка выполняет только запрет или единичную функцию).

В связи с этим классификация структур с точки зрения функциональной способности их ячеек не является целесообразной.

Особым отличительным признаком с точки зрения поведения ячеек в структуре является их способность к образованию каналов передачи сигналов в заданном направлении из всех возможных направлений передачи сигналов в структуре.

Если при передаче сигнала через данную ячейку структуры в любом заданном направлении требуется настройка только этой ячейки, то такие структуры условно будем называть структурами с *индивидуальным выбором направления* передачи сигналов.

Если для выбора направления передачи сигнала через данную ячейку требуется настройка совокупности нескольких ячеек, включая и данную ячейку, будем называть такие структуры структурами с *коллективным выбором направления* передачи сигналов.

### 1.5.3. Направленность передачи сигнала и связи ячейки

Одним из важных конструктивных признаков однородной структуры является количество возможных направлений передачи сигнала. Будем называть структуру *N-направленной*, если от любой функциональной ячейки сигнал может быть передан в любом из  $N$  направлений. Практически целесообразными являются структуры, в которых  $N \leq 8$  [7].

Количество возможных направлений передачи сигнала  $N$ , определяющее количество связей ячейки с окружающими, в значительной степени определяет и гибкость структуры с точки зрения реализации в ней различных функциональных схем. С увеличением числа направлений  $N$  гибкость структуры возрастает, однако вместе с тем усложняется конструкция структуры, так как при этом, естественно, возрастает число связей ячейки с окружающими ячейками.

В общем случае число связей ячейки в каждом из возможных направлений не ограничивается одной связью с соседней ячейкой, так как ячейка может быть симметрично связана не только с соседними ячейками.

### 1.5.4. Настройка структуры

В начальном состоянии однородная структура является недетерминированной относительно реализации в ней заданной функции. Для реализации заданной функции в структуру должна быть введена некоторая информация, однозначно определяющая реализацию требуемой функции.

Таким образом, настройка есть введение в структуру информации, которая обеспечивает перекоммутацию ячеек структуры для реализации схемы, реализующей заданную функцию.

Однородная среда называется *настраиваемой*, если функциональные ячейки среды могут быть настроены на реализацию той или иной функции только один раз, например путем выжигания отдельных соединений. Если однородная среда построена на основе многократно настраиваемых функциональных ячеек, она называется *перестраиваемой* [5]. В дальнейшем будем рассматривать только перестраиваемые однородные структуры.

Здесь предполагается, что программа настройки ячеек структуры, содержащая информацию о состояниях, в которые должны быть переведены каждая из настраиваемых ячеек, имеется вне данного участка структуры или всей структуры.

Существуют различные способы подачи управляющих сигналов на ячейки структуры.

Одним из наиболее простых является *координатный* способ выборки ячеек структуры, сопровождающийся передачей  $n$ -разрядного кода настройки, где в частном случае  $n = 1$ . В этом способе допускается произвольный порядок выборки настраиваемых ячеек. Выбор и настройка требуемой ячейки реализуются путем подачи определенной совокупности сигналов по соответствующим координатным шинам. В каждый момент времени в общем случае настраивается только одна из ячеек структуры, однако координатный способ допускает и одновременную выборку нескольких настраиваемых ячеек.

При *непосредственном* способе настройки ячеек структуры необходим индивидуальный внешний вывод от каждой ячейки, и применение его целесообразно при относительно небольшом числе ячеек.

*Адресный* способ настройки предполагает, что код настройки, сопровождаемый адресным кодом, посылается в общую систему связи, соединяющую все ячейки структуры. При этом каждая ячейка декодирует и анализирует адресный код и при совпадении его с собственным кодом записывает соответствующий код настройки.

Во многих однородных структурах отдельно используются управляющие и рабочие сигналы, которые передаются по независимым каналам.

Обычно управляющие сигналы служат для выборки требуемых ячеек (из всего множества ячеек структуры) и настройки их на выполнение заданных функций, а рабочие – являются входными аргументами реализуемых ячейками функций.

Однако в некоторых разновидностях однородных структур одни и те же сигналы являются как управляющими, так и рабочими, в зависимости от состояния ячейки.

Избранный способ настройки влияет на выбор конструктивных решений подачи рабочих сигналов (входных переменных) на структуру, а также на выбор способов отбора входных сигналов подачи и отбора по крайним ячейкам структуры или по любой ячейке структуры.

Наиболее целесообразным способом реализации однородных структур на микроэлектронных схемах является подача входных и отбор выходных сигналов по крайним ячейкам, так как не требует непосредственного доступа ко всем ячейкам структуры.

Иной способ требует осуществления внешнего вывода для каждой ячейки структуры, но при этом уменьшается количество ячеек, участвующих в реализации заданной функции, за счет сокращения длины соединительных каналов.

### 1.5.5. Функциональная способность ячейки

При реализации в однородной структуре любой логической или временной функции ячейка должна быть способна реализовать хотя бы один полный базис, а также выполнять функции канала связи между входными и выходными полюсами в заданном направлении, поэтому в общем случае ячейка структуры должна содержать в себе элементы для выполнения функций выбранного полного набора и элементы для образования канала связи. Выполняемая ячейкой в данный момент функция задается кодом настройки.

Тем не менее могут быть созданы структуры, в которых не имеет места разделение функциональной схемы ячейки; в таких структурах выполнение ячейкой функций базиса или функции канала связи осуществляется на одном и том же функциональном элементе в зависимости от сигналов настройки — извне или от окружающих ячеек.

### 1.5.6. Степень универсальности структур

Вместе с универсальными структурами, обеспечивающими решение задач классов почти с равной эффективностью, могут быть созданы структуры, в которых с наибольшей эффективностью реализуются функции некоторого класса  $R$ ; такие структуры можно считать специализированными в классе  $R$ .

Однородная среда называется *универсальной*, если в ней может быть реализован любой алгоритм функционирования устройства управления. Однородная среда называется *специализированной*, если в ней может быть реализован алгоритм из некоторого класса алгоритмов.

В частности, можно предположить, что в однородных структурах с адаптивными свойствами самоорганизующиеся и самонастраивающиеся

ся устройства могут быть реализованы более эффективно, чем детерминированные устройства с жесткой программой функционирования.

В то же время могут быть созданы структуры для реализации в них, например, только комбинационных схем или для выполнения последовательных функций определенного класса. Такие структуры также являются специализированными.

### **1.6. ЗАДАНИЕ АЛГОРИТМОВ ФУНКЦИОНИРОВАНИЯ И ПЕРЕРАБОТКИ ИНФОРМАЦИИ В АВТОМАТИЧЕСКИХ СИСТЕМАХ УПРАВЛЕНИЯ**

Большинство устройств логического управления построено непосредственно по словесному описанию их работы без использования формализованных методов (автоматные таблицы, графы переходов и т.п.) и применяются лишь в некоторых наиболее сложных случаях, как правило, для построения и минимизации отдельных небольших подсистем.

Сложность описания при использовании упомянутых формализованных подходов определяется числом входов и состояний синтезируемого автомата и почти не зависит от сложности алгоритма и соответственно от сложности предстоящей схемной реализации. Так, размер таблицы состояний автомата, имеющего  $n$  и содержащего  $s$  триггеров, будет  $2^n \cdot 2^s$ , причем он одинаков как для самого простого, так и для самого сложного устройства.

Идеальным представляется случай, когда сложность описания алгоритма пропорциональна сложности его предстоящей реализации. Отметим, что логические формулы обладают именно таким свойством: число букв в них линейно связано с числом элементов, требующихся для реализации этих формул.

Высокая стоимость разработки топологии интегральных схем диктует необходимость применения повторяющихся модульных структур, т.е. небольшого числа разных типов простых элементов. Задача состоит в определении набора элементарных ячеек (МЛМ), реализуемых наиболее эффективно и позволяющих оптимально использовать потенциальные возможности технологии изготовления новых устройств.

На основании проведенных исследований [1] установлено, что настраиваемые логические модули для построения логических устройств, исходя из специфики булевых формул, описывающих алгоритмы их

функционирования, должны быть способны реализовывать путем настройки произвольные функции  $n$  переменных, а лишь только те из них, для которых булевы формулы неповторны или обладают малой повторностью переменных, что обеспечит построение модулей с малой элементной сложностью и малым числом внешних выходов.

С конструированием матричных однородных структур связан целый ряд важнейших проблем, в частности проблемы соединений и их модульной организации.

Изотропные среды из многофункциональных логических модулей можно организовывать таким образом, чтобы обеспечить выполнение различных видов обработки информации.

Возможности и ограничения при организации соединений в изотропной среде существенным образом сказываются на быстродействии и оперативной гибкости данного устройства.

Разные алгоритмы требуют разной степени связи. Чтобы эффективно использовать специфические свойства способов организации соединений, математические операции, необходимые для обработки информации и выполнения вычислений, должны быть представлены в форме соответствующих алгоритмов.

При выполнении точечных операций каждая точка одномерной или двумерной матрицы должна обрабатываться совершенно независимо. Операции, при которых одномерный входной сигнал представляет собой временную последовательность, относятся к операциям «без запоминания». Все точки входной матрицы могут обрабатываться одинаковым образом или же для каждой из них имеется свой независимый набор правил обработки. В обоих случаях очевидно, что после ввода матрицы входных данных в матрицу изотропной среды каждый элемент может выполнять свою собственную задачу обработки независимо от остальных элементов среды. Поэтому необходимая для таких операций связь минимальна и проявляется только при вводе данных в изотропную среду или выводе из нее. Следовательно, такие операции могут выполняться параллельно.

Многие алгоритмы обработки информации могут быть выражены через матричные операции. Одной из основных операций матричной алгебры является умножение двух матриц (частным случаем такой операции можно считать умножение вектора на матрицу). Математически такое умножение описывается следующим образом:



$$C_{ij} = \sum_{k=1}^N A_{ik} B_{kj}, \quad i, j = 1, \dots, N, \quad (1.7)$$

где для удобства через  $A$ ,  $B$  и  $C$  обозначим квадратные матрицы  $N \times N$ . Результирующую матрицу можно иначе определить как сумму  $N$  матриц внешнего произведения, полученных умножением вектор-столбцов матрицы  $A$  на соответствующие вектор-строки матрицы  $B$ :

$$C = \sum_{k=0}^N C^{(k)}, \quad C_{ij}^{(k)} = A_{ik} B_{kj}, \quad i, j = 1, \dots, N, \quad (1.8)$$

где вторая строка – это внешнее произведение  $k$ -го столбца матрицы  $A$  на  $k$ -ю строку матрицы  $B$ . Из вышеприведенных уравнений очевидно, что при данной операции высока степень связанности элементов входных матриц и результирующей матрицы. Так, все элементы данной строки матрицы  $A$  и данного столбца матрицы  $B$  дадут вклад в один элемент результирующей матрицы  $C$  и наоборот, один элемент матрицы  $A$  (или  $B$ ) дает вклад во все элементы соответствующей строки (или столбца) матрицы  $C$ .

Регулярный характер связи, задаваемый уравнениями (1.7) и (1.8), означает, что операции можно выполнять также посредством рекуррентных и локально связанных алгоритмов, реализуемых с помощью изотропных сред из многофункциональных логических модулей [10].

Две важные операции обработки информации, требующие простых связей между всеми элементами матрицы входных данных, – это фурье-преобразование и сортировка. Иными словами, при этих операциях на каждый элемент выходной матрицы влияют все элементы входной матрицы и каждый элемент входной матрицы влияет на все элементы выходной матрицы. Вычисление при фурье-преобразовании сводится к умножению и сложению комплексных чисел, тогда как сортировка является операцией сравнения.

Операция сортировки числовой последовательности включает в себя элементарные операции сравнения двух чисел и расположение их на выходе в убывающем порядке. Одним из эффективных алгоритмов сортировки является сортирующий алгоритм Батчера [9]. Основной принцип, положенный в основу этого алгоритма, состоит в применении метода «разделяй и властвуй», т.е. крупная задача делится на несколько задач меньшего масштаба, и комбинация их решений на выходе позволяет получить решение общей задачи.

В настоящее время можно определить три направления, по которым идет развитие элементной базы большой степени интеграции для построения разного рода управляющих устройств [11].

В рамках *первого направления* ведется разработка микропроцессоров и микропроцессорных наборов на базе одного кристалла. При этом вопрос соответствия структурных и функциональных особенностей микропроцессоров структуре ЭВМ по существу не был учтен. Это вызвало необходимость использования мультипроцессорной организации цифровых систем на основе микропроцессоров и разработки соответствующих принципов обработки информации и управления.

*Второе направление* предполагает достижение высокой эффективности элементной базы большой степени интеграции благодаря максимальному соответствию структуры интегральных схем структуре проектируемых управляющих устройств. Элементами реализации этого направления являются матричные БИС, программируемые логические матрицы и перепрограммируемые запоминающие устройства.

*Третье направление* базируется на автоматном принципе обработки информации, это определяет структурное сближение модулей обработки информации, управления и электронной части модулей внешних устройств.

Таким образом, предполагается модульное построение систем управления на основе принципов многофункциональности и регулярности. Отличительной особенностью последнего направления от предыдущих является строгая формализация процесса его синтеза, который базируется на основе теории многофункциональных автоматов.

## 1.7. АППАРАТНАЯ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ ФУНКЦИОНИРОВАНИЯ УПРАВЛЯЮЩИХ УСТРОЙСТВ

Существует два основных принципа реализации алгоритма функционирования устройств управления (УУ) – *аппаратурная (схемная) реализация* в том или ином элементном базисе и *программная реализация* [2]. При аппаратурной реализации, когда по заданному алгоритму функционирования строится структурная схема УУ в каком-либо базисе элементов, говорят, что алгоритмическое описание переводится в *структурное описание*, представляющее собой структурную модель заданного алгоритма функционирования УУ.

Переход от алгоритмического описания к структурному в базе однородной среды для программируемого УУ означает, что по заданному алгоритму функционирования УУ осуществляется настройка многофункциональных элементов однородной среды.

При настройке многофункциональных элементов на реализацию одной из функций могут быть использованы различные способы, основными из которых являются *технологический*, когда настройка осуществляется удалением (например, выжиганием) и созданием (например, напылением пленок) отдельных соединений, и *функциональный*, когда настройка осуществляется подачей на какие-либо входы определенных сигналов.

Дискретные устройства с итеративной структурой его функциональной схемы будем называть однородной средой, если в качестве элементов функциональной схемы устройства использованы МЛМ.

Такое дискретное устройство путем определенной настройки его МЛМ, очевидно, может быть настроено на реализацию алгоритмов функционирования любого УУ.

Альтернативой структурному моделированию алгоритма функционирования УУ в виде аппаратурной реализации является программное моделирование. При программном моделировании вместо структуры УУ в том или ином элементном базисе алгоритм функционирования представляется в виде последовательности так называемых команд, каждая из которых определяет операцию над множеством исходных или промежуточных данных (операндов) и номер следующей команды.

Упорядоченная совокупность команд, однозначно описывающая заданный алгоритм функционирования, называется *программой*, а представление алгоритма функционирования в виде программы – *программным моделированием алгоритма*.

В отличие от аппаратурной реализации, при структурном моделировании алгоритма функционирования программа, реализующая тот же алгоритм функционирования при программном моделировании, является пассивной, т.е. представляет собой некоторую запись алгоритма функционирования. Для чтения этой записи и выполнения команд при программном моделировании можно использовать *микропроцессор*.

Таким образом, при структурном моделировании аппаратурно реализуется весь заданный алгоритм функционирования УУ. При программном моделировании микропроцессор, представляющий собой

универсальное устройство, настраиваемое на аппаратную реализацию одновременно только одной операции, обеспечивает выполнение всего алгоритма функционирования путем *последовательного* выполнения четко разграниченных отдельных операций (*актов алгоритма*), определяемых программой, хранящейся в запоминающем устройстве.

Легко понять, что при структурном моделировании изменение алгоритма функционирования требует изменения структурной схемы УУ, тогда как при программном моделировании необходимо изменить лишь хранящуюся в запоминающем устройстве программу.

**Особенности реализации алгоритмов функционирования УУ в однородной среде.** Если каждый элемент однородной среды (ОС) соединен непосредственно только со своими ближайшими соседями, то речь идет об ОС с близкодействием. Достоинством таких ОС является возможность локализации и исключения из числа используемых элементов неисправных путем соответствующей настройки соседних с ними элементов ОС. Однако наличие связей только между соседними элементами ОС с близкодействием приводит к тому, что для организации более протяженных связей приходится нерационально использовать большое число элементов среды. При этом возникает значительная избыточность. Кроме того, элементы таких ОС, выполняющие большое число различных операций, оказываются достаточно сложными, и требуется большой объем информации для их настройки.

Если ОС представляет собой прямоугольную итеративную решетку из  $H$  горизонтальных и  $V$  вертикальных шин, то будем называть ее матричной ОС. Разработка различных типов ОС привела к необходимости создания методов синтеза управляющих устройств в таких средах. Существующие методы синтеза можно разделить на две принципиально отличные группы:

- 1) моделирование в ОС алгоритма функционирования УУ;
- 2) размещение в ОС функциональной схемы, реализующей алгоритм функционирования УУ.

При использовании первого из указанных подходов осуществляется настройка ОС на реализацию исходного описания автомата.

Методы синтеза УУ в ОС, относящиеся ко второй группе, состоят в том, что по исходному описанию УУ строится функциональная схема в заранее выбранном логическом базисе и размещается в используемой для синтеза ОС. При этом говорят также о «натягивании» заданной

функциональной схемы, или логической, сети на выбранную ОС или о программировании ОС для реализации в ней заданной схемы [3].

Необходимо отметить, что реализация синтезируемого устройства ОС не заканчивается процессом размещения в ней логической схемы. В соответствии с выбранным размещением необходимо получить упорядоченный массив кодов настройки, который затем передается в настройечную память элементов ОС.

Таким образом, перенастройка программируемого УУ на реализацию другого алгоритма функционирования при программной его реализации состоит в замене программы в запоминающем устройстве, а при аппаратной реализации с помощью ОС – в перенастройке МЛМ.

## **1.8. ВЫВОДЫ**

Основные результаты, полученные в данной главе, сводятся к следующему.

1. Даны общие понятия синтеза структурных конечных автоматов.
2. Приведены способы задания функции алгебры логики и классификация автоматов.
3. Описаны основные принципы построения перестраиваемых автоматов.
4. Рассмотрены основные понятия однородных структур.
5. Раскрыты способы задания алгоритмов функционирования и их аппаратная и программная реализация, а также особенности их реализации в однородных структурах.

## **ЛИТЕРАТУРА**

1. *Артюхов В.Л., Копейкин Г.А., Шалыто А.А.* Настраиваемые модули для управляющих логических устройств. – Л.: Энергоатомиздат, 1981. – 168 с.
2. *Будинский Я.* Логические цепи в цифровой технике. – М.: Связь, 1977. – 392 с.
3. *Евреинов Э.В.* Однородные вычислительные системы, структуры и среды. – М.: Радио и связь, 1981. – 208 с.
4. *Ершова Э.Б. и др.* Основы дискретной автоматики в электросвязи. – М.: Связь, 1980. – 232 с.
5. *Лазарев В.Г., Пийль Е.И., Турута Е.Н.* Построение программируемых управляющих устройств. – М.: Энергоатомиздат, 1984. – 192 с.

6. *Поспелов Д.А.* Логические методы анализа и синтеза схем. – М.: Энергия, 1964. – 320 с.
7. *Прангшвили И.В., Абрамова Н.А., Бабичева Е.В., Игнатущенко В.В.* Микроэлектроника и однородные структуры для построения логических и вычислительных устройств. – М.: Наука, 1967. – 228 с.
8. *Пупырев Е.И.* Перестраиваемые автоматы и микропроцессорные системы. – М.: Наука, 1984. – 192 с.
9. *Фет Я.И.* Массовая обработка информации в специализированных однородных процессорах. – Новосибирск: Наука, 1976. – 200 с.
10. *Шальто А.А.* Логическое управление. Методы аппаратной и программной реализации алгоритмов. – СПб.: Наука, 2000. – 780 с.
11. *Шидловский С.В.* Задание алгоритмов управления и переработки информации в проблемно-ориентированных комплексах // Научная сессия ТУСУР: Материалы докл. межрегиональной науч.-техн. конф. 14–16 мая 2002 г. Томск, Россия. – Томск: ТУСУР, 2002. – Ч. 2. – С. 321–333.

## **БУЛЕВА МОДЕЛЬ ЛОГИКИ ПЕРЕСТРАИВАЕМЫХ СТРУКТУР**

---

---

### **2.1. ВВОДНЫЕ ПОНЯТИЯ**

Построение систем логического управления разнообразными объектами промышленной автоматики, обеспечивающих их эффективную согласованную работу в соответствии с заданным алгоритмом функционирования, всегда считалось одной из важнейших задач автоматизации производственных процессов.

За последние годы элементная база систем логического управления быстро изменялась, пройдя путь от релейно-контактных элементов, микросхем малого и среднего уровней интеграции к программным логическим устройствам, а затем к большим и сверхбольшим интегральным схемам, частным случаем которых являются микропроцессоры и микроконтроллеры, используемые в качестве вычислительного ядра, в том числе и в промышленных (управляющих) компьютерах, и в программируемых логических контроллерах.

Особенности построения систем логического управления на основе многофункциональных логических модулей состоят в том, что средством реализации алгоритмов функционирования проектируемых систем являются функциональные модули, которые могут настраиваться на реализацию конкретного логического алгоритма из заданного класса.

Настраивая один (или несколько) однотипный элемент, на основе которого создается логическое устройство управления, на реализацию алгоритма функционирования последнего, можно построить заданное устройство управления в базе лишь одного универсального элемента [9, 15]. Практически удобнее иметь набор различных элементов (универсальный базис) не на основе одного типа перестраиваемого элемента, а в виде набора небольшого числа (обычно не более пяти-шести) типов перестраиваемых элементов [4].

Разработки же формальных методов аппаратной и программной реализации алгоритмов логического управления были актуальны на протяжении многих лет и являются актуальными в настоящее время.

В логическом проектировании кроме задач, непосредственно возникающих из потребностей практики, имеется также уже ставшая традиционной область теоретических задач, связанных с исследованиями как класса всех булевых функций, являющихся одним из важнейших объектов дискретной математики, так и его подклассов.

Таким образом, большой интерес представляют исследования булевых формул одной из реализаций булевых функций, первичным заданием которых являются таблицы истинности. При этом особую важность имеют простейшие из булевых формул – бесповторные и повторные.

## 2.2. КЛАССИФИКАЦИЯ БУЛЕВЫХ ФОРМУЛ

Функция, у которой ее значения и значения аргументов двоичны, называется булевой [6].

Булевы функции (БФу) задаются в виде таблиц истинности (ТИ). Аналогичная запись БФу в виде формулы, использующей логические операции (базис формулы), называется булевой формулой (БФ).

В настоящей работе рассматриваются формулы в базисе И, ИЛИ, НЕ. Зафиксируем базис логических операций. При этом основной метрикой БФ является число  $h$  символов переменных и их инверсий в правой ее части, которые будем называть буквами.

В настоящее время БФ в фиксированном базисе из  $h$  букв могут быть разделены на два класса: бесповторные, для которых  $h = n$ , и повторные, для которых  $h > n$  (рис. 2.1). В настоящей работе, если это не оговаривается особо, рассматриваются одиночные, полностью определенные БФу  $n$  переменных, заданных на  $2^n$  входных наборах. Основной метрикой БФу является число переменных  $n$ .

Каждый логический элемент, реализующий отрицание и дизъюнкцию либо отрицание и конъюнкцию, в определенном смысле универсален, ибо, выбирая нужное число элементов и схему их соединения, можно построить любой логический автомат. Согласно [1] основную (93–99,7%) долю реализуемых в устройствах управления составляют бесповторные БФ или близкие к ним.



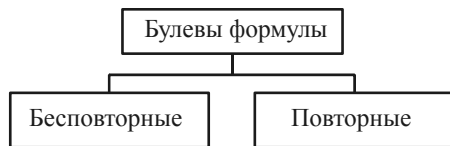


Рис. 2.1. Классификация булевых формул

Формулу будем называть бесповторной, если каждый аргумент, взятый в прямой или инверсной форме, входит в нее не более одного раза. Во всех остальных случаях формула является повторной.

Бесповторные БФ подвергнем дальнейшей классификации (рис. 2.2) [11]. Повторные БФ имеют аналогичную классификацию.

Если в записи бесповторной булевой формулы  $f(x_i)$ , где  $i = 1, 2, \dots, n$ ,  $n$  – число аргументов, индекс  $i$  при логических аргументах возрастает слева направо, то будем считать, что эта формула упорядочена. Упорядоченной будем считать формулу и в том случае, если существуют тождественные преобразования, в результате которых получается запись формулы с возрастающими слева направо индексами аргументов. Если в записи бесповторной булевой формулы аргументы с теми или иными индексами отсутствуют, будем считать, что эта формула содержит пропуски соответствующих аргументов, и назовем ее формулой с пропусками аргументов. Формулами с пропусками аргументов будем считать и такие бесповторные булевы формулы, для которых  $\mu < n$ , где  $\mu$  – максимальный индекс при логическом аргументе.

**Пример 1.** Формула  $f_1 = A_1 \vee A_2 \bar{A}_3 \vee A_4$  зависит от аргументов  $A_1, A_2, A_3, A_4$ , является упорядоченной, представлена в дизъюнктивно нормальной форме (ДНФ) и не содержит пропусков аргументов.

**Пример 2.** Формула  $f_2 = (\bar{A}_1 \vee A_2)(\bar{A}_3 \vee A_4)$  зависит от аргументов  $A_1, A_2, A_3, A_4$ , является упорядоченной, представлена в конъюнктивно нормальной форме (КНФ) и не имеет пропусков аргументов.

**Пример 3.** Формула  $f_3 = [(A_1 \vee A_2)A_3 \vee A_4]A_5$  зависит от аргументов  $A_1, A_2, A_3, A_4, A_5$ , представлена в форме четвертого порядка, является упорядоченной и не содержит пропусков аргументов.

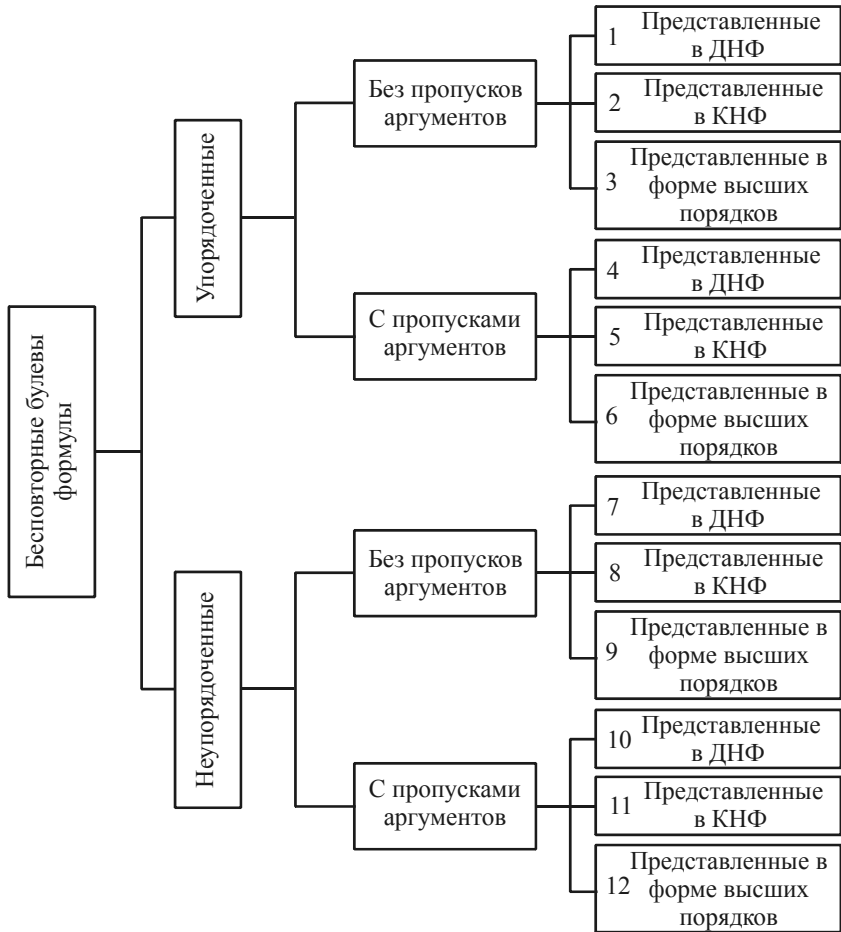


Рис. 2.2. Классификация бесповторных булевых формул

Пример 4. Формула  $f_4 = A_1 \vee A_3 A_4$  зависит от аргументов  $A_1, A_2, A_3, A_4$ , упорядочена, представлена в ДНФ с пропуском аргумента  $A_2$ .

Пример 5. Формула  $f_5 = (A_1 \bar{A}_2 \vee A_3 \bar{A}_4) A_5$  зависит от аргументов  $A_1, A_2, A_3, A_4, \dots, A_8$ , является упорядоченной, представлена в КНФ с пропуском аргументов  $A_6, A_7, A_8$ .

Пример 6. Формула  $f_6 = A_1 \bar{A}_3 \vee A_2 \bar{A}_4$  зависит от аргументов  $A_1, A_2, A_3, A_4$ , является неупорядоченной, представлена в ДНФ и не содержит пропусков аргументов.

В дальнейшем будем пользоваться приведенной классификацией.

Таким образом, в соответствии с приведенной классификацией (см. рис. 2.2) множество булевых формул распадается на попарно непересекающиеся классы – множества однотипных формул. Каждую формулу определенного класса можно выбрать в качестве представителя этого класса. Булевы формулы, принадлежащие одному классу, реализуются физически одинаковыми схемами, поэтому для каждого класса достаточно реализовать лишь одну схему, структура которой описывается формулой представителя класса.

### 2.3. БУЛЕВА МОДЕЛЬ ЛОГИКИ ПЕРЕСТРАИВАЕМЫХ СТРУКТУР ДЛЯ ОПРЕДЕЛЕННЫХ КЛАССОВ БУЛЕВЫХ ФОРМУЛ

Разработка сложных дискретных управляющих устройств требует обычно больших трудозатрат, поэтому является экономически оправданным создание логических управляющих устройств, которые способны при соответствующей перестройке, не затрагивающей их структуры, управлять работой других объектов.

Любой перестраиваемый автомат может рассматриваться как множество автоматов с одними и теми же выходами, причем настройка определяет тот автомат, выходы которого считаются при этой настройке выходами всего перестраиваемого автомата.

Многофункциональные логические модули, настраиваемые на реализацию любой из  $2^{2^n}$  булевых формул от  $n$  входных переменных, на-

зываются универсальными логическими модулями (УЛМ) от  $n$  переменных и описываются, согласно [7], следующей булевой формулой:

$$U = a_1 z_1 \vee a_2 z_2 \vee \dots \vee a_{2^n} z_{2^n},$$

где  $a_1, \dots, a_{2^n}$  – все возможные конъюнкции от  $n$  переменных и их отрицаний. Приравнявая некоторые  $z$  нулю, а остальные – единице, получим формулу, зависящую только от  $x_1, \dots, x_n$  и заданную в совершенной дизъюнктивной нормальной форме (СДНФ). Меняя подстановку 0 и 1 на множестве  $z_i$  ( $i=1, \dots, 2^n$ ), можно получать различные СДНФ.

На практике часто необходимо создавать перестраиваемый логический модуль, реализующий не все возможные булевы формулы, а любую формулу из заранее заданного множества, т.е. перестраиваемый автомат, универсальный на множестве булевых формул. Перестраиваемые модули, универсальные на множествах формул, по сравнению с УЛМ имеют меньшее число настроечных входов и меньшую сложность, поэтому разработка моделей логики перестраиваемых структур имеет большое значение для практики. Ниже приведем одну из них, позволяющую получить перестраиваемые автоматы с простой настройкой (настройка, при которой используется два сигнала: 0 и 1).

Суть заключается в следующем:

1. Пусть задано множество функций алгебры логики, каждая из которых может быть реализована на выходе перестраиваемого автомата:  $f_1(x_1, \dots, x_n), f_2(x_1, \dots, x_n), \dots, f_R(x_1, \dots, x_n)$ .

2. Введем  $t = \log_2 R$  дополнительных переменных  $z_1, \dots, z_t$ .

3. Поставим каждой из исходных функций алгебры логики  $f_i$  ( $i=1, \dots, t$ ) конъюнкцию  $a_i = z_1^{\sigma_{i,1}} \dots z_t^{\sigma_{i,t}}$ , где  $\sigma_{i,j} \in \{0,1\}$  ( $j=1, \dots, t$ ),  $z_j^0 = \bar{z}_j$ ,  $z_j^1 = z_j$ , так, чтобы разным функциям алгебры логики сопоставлялись различные конъюнкции.

4. Образует формулу

$$U = a_1 f_1(x_1, \dots, x_n) \vee a_2 f_2(x_1, \dots, x_n) \vee \dots \vee a_R f_R(x_1, \dots, x_n).$$

Логическая схема, реализующая заданную формулу, имеет информационные входы  $x_1, \dots, x_n$  и дополнительные входы  $z_1, \dots, z_n$ , исполь-

зубые для настройки. При этом если, например, нужно настроить полученный перестраиваемый автомат на реализацию функции алгебры логики  $f_i$ , которой сопоставлена конъюнкция  $z_1^{\sigma_{i1}} \dots z_i^{\sigma_{ii}}$ , то на настроечные входы  $z_1, \dots, z_i$  подаются соответственно сигналы-константы  $\sigma_{i1}, \dots, \sigma_{ii}$ .

Таким образом, используя булеву модель логики перестраиваемых структур, можно создать логическое устройство управления с перестраиваемой структурой, реализующее алгоритм управления из заранее заданного множества булевых формул.

## 2.4. МНОГОФУНКЦИОНАЛЬНЫЕ ЛОГИЧЕСКИЕ МОДУЛИ. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Напомним, что под МЛМ будем понимать перестраиваемый автомат, для любых настроек которого значение на его выходах зависит от значений сигналов на его входах.

Сигналы, подаваемые на входы МЛМ, разделяются на две группы. Сигналы, несущие информацию, предназначенную для обработки МЛМ, назовем *информационными* и обозначим  $A_1, A_2, \dots, A_n$ , где  $n$  – число аргументов. Эти сигналы включаются на  $n$  входов МЛМ. На остальные  $\eta = (h - n)$  входов подаются сигналы, которые назовем *настроечными*, где  $h$  – общее количество входов МЛМ.

Настройку, при которой используется два сигнала (0, 1), назовем *простой*, а настройку, которая производится  $n + 2$  сигналами  $(0, 1, z_1, z_2, \dots, z_n)$ , будем называть *нормальной*. Нормальная настройка по сравнению с простой зачастую позволяет увеличивать число функций, реализуемых МЛМ. Однако использование для настройки сигналов  $z_1, z_2, \dots, z_i$  приводит к необходимости подавать сигнал  $z_i$ , где  $i \in \{1, 2, \dots, n\}$ , не на один, а сразу на несколько входов.

При использовании для настройки всех  $2n + 2$  сигналов, а именно  $(0, 1, z_1, z_2, \dots, z_n, \bar{z}_1, \bar{z}_2, \dots, \bar{z}_n)$ , настройка называется *сложной*. По отношению к нормальной сложная настройка позволяет увеличить число функций, реализуемых МЛМ. Однако ее использование оказывается возможным лишь при наличии информационных сигналов и их отрицаний.

В отношении выбора входов МЛМ, на которые подаются сигналы настройки, существуют две возможности. Если для подачи  $\eta$  сигналов настройки выделяется  $\eta$  определенных входов МЛМ, то говорят, что он имеет *фиксированные* входы настройки. Если же для настройки могут быть использованы любые  $\eta$  из  $h$  входов МЛМ, то будем считать, что он имеет *произвольные* входы настройки [20].

Также будем считать, что имеется некоторый алфавит  $N$ ,  $N = (n_1, n_2, \dots, n_k)$ , который используется для кодирования букв алфавита  $X, Y, Z$ .

Пусть  $X = \{x_1, x_2, \dots, x_n\}$ ,  $Y = \{y_1, y_2, \dots, y_m\}$ ,  $Z = \{z_1, z_2, \dots, z_p\}$  – конечные алфавиты. Функции  $f$ , определенные на  $X^*$  и принимающие значения  $Y^*$  в зависимости от значений  $Z^*$ , могут быть использованы для описания поведения МЛМ в дискретные моменты времени  $t = 1, 2, \dots$ , где  $X^*$ ,  $Y^*$  и  $Z^*$  – множество слов в алфавитах  $X, Y, Z$ . При этом слово  $x$ ,  $x \in X^*$ , интерпретируется как последовательность входных сигналов,  $f(x) \in Y^*$  – как последовательность сигналов, возникающих на выходе МЛМ, в зависимости от содержания слов  $z$ ,  $z \in Z^*$ .

Множества слов алфавитов  $X$  и  $Z$  могут быть составными частями некоторого множества  $M$ . Введем следующие понятия.

1. Если настроенный код подается на настроенные входы МЛМ, то будем говорить, что речь идет о прямом варианте реализации функции  $N_1$ .

2. Если же настроенный код подается на место входных сигналов МЛМ, а входные сигналы – на настроенные входы, то условимся называть это обратным вариантом реализации функции ( $N_2$ ), т.е. слова из алфавитов  $X$  и  $Z$  меняются местами. Тогда имеем

$$M_1 = \bar{N}_1 \cap N_2 \neq 0;$$

$$M_2 = N_1 \cap \bar{N}_2 \neq 0;$$

$$M_3 = N_1 \cap N_2 \neq 0.$$

Функции из множеств  $M_2$  и  $M_3$  соответствуют прямой реализации, а функции из множеств  $M_1$  могут быть реализованы лишь в обратном варианте.

В МЛМ с фиксированными входами настройки сигналы подаются только на входы, выделенные для этой цели. Важной особенностью элементов с фиксированными входами настройки (по сравнению с элементами, имеющими произвольные выходы настройки) является возможность уменьшения числа внешних вводов. Правда, это достигается за счет усложнения логической структуры элемента и принятия  $n = \text{const}$ . Кроме того, при использовании элементов с фиксированной настройкой нет необходимости в переключении информационных сигналов, что позволяет строить логические автоматы, в которых функциональные возможности элементов изменяются во время переработки информации.

Таким образом, фиксация входов ячейки (МЛМ) позволяет сделать внутренними все входы настройки и подавать на них сигналы. При этом заметим, что появляются новые (дополнительные) внешние входы настройки.

Будем различать МЛМ с внешней и внутренней настройкой. Структурная схема ячейки с внешней настройкой (рис. 2.3), использующая многопозиционный код, может быть описана выражением [20]:

$$Y_j = \bigvee_{i=1}^{\gamma} f_i A_i, \quad (2.1)$$

где  $f_i$  – базовая формула, т.е. формула, принадлежащая произвольному множеству формул  $f_1, f_2, \dots, f_\gamma$ ;  $A_i$  – сигнал настройки,  $j \in \{1, 2, \dots, m\}$ ;  $\gamma$  – число базовых формул.

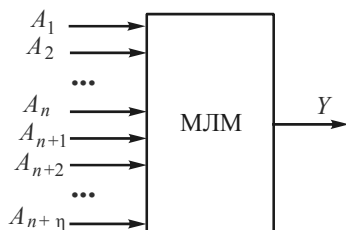


Рис. 2.3. Структурная схема элемента с внешней настройкой

Структура МЛМ с внутренней настройкой представлена на рис. 2.4. Основной логический блок описывается в зависимости от выбранного

метода кодирования уравнением (2.1) либо уравнением вида

$$Y_j = \bigvee_{i=1}^{\gamma} f_i A_1^i \dots A_n^i \text{ при использовании однопозиционного кода.}$$

На входы  $A_1, A_2, \dots, A_n$  подаются информационные сигналы, на входы  $A_{n+1}, A_{n+2}, \dots, A_{n+\eta}$ , являющиеся внутренними, подаются сигналы настройки  $I_1, I_2, \dots, I_\eta$ , генерируемые блоками настройки  $E_1, E_2, \dots, E_\eta$ .

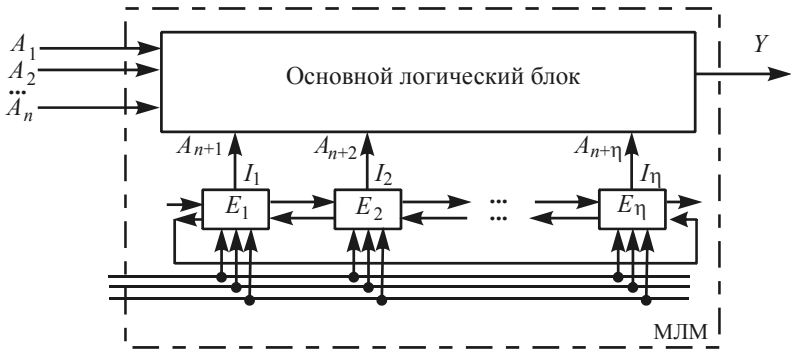


Рис. 2.4. Структура элемента с внутренней настройкой

С помощью рассматриваемых в данной работе МЛМ можно строить различные однородные среды. Напомним, что под *однородной* средой в дальнейшем будем понимать дискретное устройство с итеративной структурой его функциональной схемы, если в качестве элементов этой схемы используются *однотипные* МЛМ. Если же дискретное устройство построено на основе *разнотипных* МЛМ, то подобные среды будем называть *квазиоднородными*. Квазиоднородные среды позволяют, с одной стороны, с помощью применения нескольких типов МЛМ существенно сократить избыточность, которая неизбежна при применении одного типа МЛМ, а с другой – сохранить все основные достоинства однородных сред.

Таким образом, используя различные МЛМ, можно создавать электронные устройства с перестраиваемой структурой, которые за счет перестройки в процессе или перед началом работы могут решать различные задачи.



## 2.5. ВЫЧИСЛЕНИЕ БУЛЕВЫХ ФОРМУЛ

### 2.5.1. Вычисление неповторных ДНФ и КНФ булевых формул

#### 2.5.1.1. Вычисление конъюнктивной нормальной формы формул на структуре для дизъюнктивной нормальной формы

В работах [16, 18] представлен МЛМ, обеспечивающий вычисление неповторных ДНФ функций алгебры логики. МЛМ (рис. 2.5, а) описывается формулами вида

$$\varepsilon_{i-1} = A_{i-1}\varepsilon_{i-2} \vee T_{i-1}; \quad (2.2)$$

$$\varphi_i = \varphi_{i-1} \vee \varepsilon_{i-1}A_iT_i, \quad (2.3)$$

где  $A_i$  – логические аргументы, обозначающие информационные входы ячейки;  $T_i$  – логические аргументы, обозначающие настроечные входы ячейки;  $i = 1, 2, \dots, n$ ,  $n$  – число ячеек линейной среды (см. рис. 2.5, б).

Рассматриваемая структура (которую в дальнейшем будем называть  $F$ -структурой) предназначена для реализаций формул, представленных в ДНФ, но для реализации формул, представленных в КНФ, нет необходимости в использовании отдельной схемы, так как схема, предназначенная для ДНФ, может быть использована и для КНФ [3, 5].

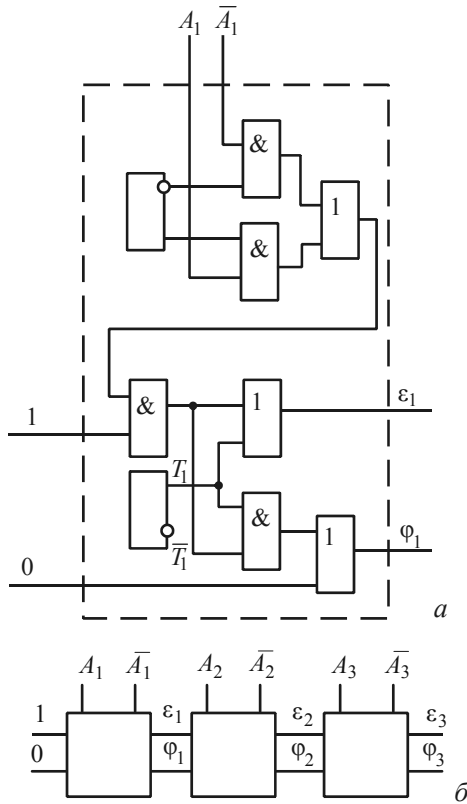
Поставим в соответствие каждому логическому аргументу неповторной упорядоченной формулы двоичный разряд и единицей условимся обозначать последний справа аргумент каждой конъюнкции. Все остальные аргументы будем обозначать нулями. В результате получим  $n$ -разрядное двоичное число, где  $n$  – число аргументов формулы. Условимся называть такое число  $\alpha$ -кодом. Для примера рассмотрим формулу (примем  $n = 7$ ):

$$f = A_1 \vee A_2A_3 \vee A_4A_5A_6.$$

Ее  $\alpha$ -код имеет вид

$$\alpha = 1010010.$$

Очевидно, что всякой булевой формуле, представленной в ДНФ и имеющей не более  $n$  вхождений аргументов, можно поставить в соответствие некоторый  $\alpha$ -код. При этом одной и той же формуле могут соответствовать различные  $\alpha$ -коды в зависимости от расположения в этой формуле конъюнкций различной длины [7].

Рис. 2.5. Ячейка  $F$ -структуры

Чтобы отличать инверсные аргументы от неинверсных, каждому разряду  $\alpha$ -кода поставим в соответствие еще один двоичный разряд. Если аргумент находится в неинверсной форме, то условимся его обозначать нулем, если в инверсной – единицей. Полученное при этом  $n$ -разрядное двоичное число будем называть  $\beta$ -кодом. Например, для формулы ( $n = 7$ )

$$f = A_1 A_2 \vee \bar{A}_3 A_4 \vee \bar{A}_5 \bar{A}_6$$

$\beta$ -код имеет вид

$$\beta = 0010110.$$

Два кода ( $\alpha$  и  $\beta$ ) полностью определяют структуру неповторной упорядоченной булевой формулы, представленной в ДНФ и содержащей инверсные и неинверсные аргументы.

Для реализации конъюнктивных форм формул достаточно записать  $\gamma$ -код точно так же, как и  $\alpha$ -код, с той лишь разницей, что группа нулей с примыкающей к ним справа единицей будет соответствовать дизъюнкции, записанной в скобках. Например, для формулы

$$f = (A_1 \vee A_2 \vee \bar{A}_3 \vee \bar{A}_4)(A_5 \vee \bar{A}_6 \vee A_7)(\bar{A}_8 \vee A_9)A_{10}$$

$\gamma$ -код представится в виде

$$\gamma = 0001001011.$$

Если этот код ввести в  $F$ -структуру, то будет реализована формула

$$f_1 = A_1A_2A_3A_4 \vee A_5A_6A_7 \vee A_8A_9 \vee A_{10},$$

в которой все аргументы не содержат инверсий. Чтобы учесть инверсии, достаточно записать  $\beta$ -код и проинвертировать его:

$$\bar{\beta} = 1100101011.$$

С учетом  $\bar{\beta}$ -кода  $F$ -структура обеспечит реализацию формулы

$$f_2 = \bar{A}_1\bar{A}_2A_3A_4 \vee \bar{A}_5A_6\bar{A}_7 \vee A_8\bar{A}_9 \vee \bar{A}_{10}.$$

Очевидно, что  $f_2 = \bar{f}$ .

Таким образом, чтобы с помощью  $F$ -структуры реализовать конъюнктивную форму формулы, достаточно:

- 1) найти  $\gamma$ -код;
- 2) найти  $\bar{\beta}$ -код;
- 3) проинвертировать сигнал, снимаемый с выхода  $F$ -структуры.

В справедливости этого утверждения нетрудно убедиться, если учесть, что в случае конъюнктивных форм определение  $\gamma$ -кода соответствует операции замены знаков дизъюнкции на знаки конъюнкции и наоборот, а инвертирование  $\beta$ -кода соответствует замене инверсных аргументов неинверсными и наоборот. Этими двумя операциями реализуются правила инвертирования формул по теореме де Моргана.

Таким образом, с помощью одной и той же ячеистой структуры можно реализовать и дизъюнктивные, и конъюнктивные нормальные формы неповторных упорядоченных булевых формул.

## 2.5.1.2. Анализ работы модуля

Работа структуры рассматривается в предположении равной доступности прямых и инверсных выходов источников информации при помощи  $\beta$ -кода.

Исследуем возможности  $F$ -структуры в предположении, что настроечными являются входы  $A_i$ , а информационными – входы  $T_i$  ( $i = 1, 2, \dots, n$ ), где  $n$  – число ячеек изотропной структуры. Подставим формулу (2.2) в (2.3) и получим

$$\varphi_i = \varphi_{i-1} \vee (A_{i-1} \varepsilon_{i-2} \vee T_{i-1}) A_i T_i = \varphi_{i-1} \vee A_{i-1} T_i A_i \varepsilon_{i-2} \vee T_{i-1} T_i A_i.$$

Здесь  $\varepsilon_{i-2}$  состоит из конъюнкций, в каждую из которых  $T$ -аргументы входят не более одного раза. Следовательно, в ДНФ выражения  $\varphi_i$ , записанного через  $A$ - и  $T$ -аргументы, нет ни одной конъюнкции, содержащей более двух  $T$ -аргументов. Представим формулу (2.3) в ДНФ и выразим ее через  $A$ - и  $T$ -аргументы, например для  $n = 8$ :

$$\begin{aligned} \varphi_i = & A_1 T_1 \vee A_1 A_2 T_2 \vee A_2 T_1 T_2 \vee A_1 A_2 A_3 T_3 \vee A_1 A_2 T_1 T_3 \vee \dots \vee T_i \bigwedge_{i=1}^n A_i \vee \\ & \vee T_1 T_8 \bigwedge_{i=2}^n A_i \vee T_2 T_8 \bigwedge_{i=3}^n A_i \vee \dots \vee T_7 T_8 A_8. \end{aligned}$$

Эта формула состоит из дизъюнкции конъюнкций двух видов:

а) конъюнкций, содержащих только один  $T$ -аргумент, вида

$$T_h \bigwedge_{i=1}^h A_i, \quad h = 1, 2, \dots, n.$$

Всего их существует  $n$ ;

б) конъюнкций, содержащих только два различных  $T$ -аргумента; всего их  $C_n^2$ . При этом каждая конъюнкция, содержащая  $\lambda$   $A$ -аргументов, имеет вид

$$T_h T_{h+\lambda} \bigwedge_{i=1}^{h+\lambda} A_{h+i},$$

т.е. индексы  $A$ -аргументов образуют натуральную последовательность, начиная с  $h+1$  и заканчивая  $h+\lambda$ .

Таким образом, если  $A$ -аргументы использовать в качестве настроечных, а  $T$ -аргументы считать логическими независимыми переменными, то с помощью  $F$ -структуры невозможно реализовать формулы, ми-

нимальные ДНФ которых содержат хотя бы одну конъюнкцию, охватывающую более двух аргументов.

Выясним, как по настроечному коду, поданному на  $A$ -входы, определить вид формулы, реализуемой  $F$ -структурой. Для этого сначала введем понятие полной группы единиц настроечного кода. Группу единиц в настроечном коде будем называть полной в следующих случаях:

а) если группа единиц, между которыми нет нулей, ограничена с обеих сторон нулями;

б) если группа единиц, между которыми нет нулей, находится либо в начале кода, либо в его конце.

Пр и м е р. Настроечный код вида

1 1 1 0 0 0 1 0 0 1 1 0 0 1 1 1 1

содержит четыре полные группы единиц. Первая из них содержит три единицы и находится в начале кода, вторая состоит из одной единицы, третья – из двух и четвертая – из расположенных в конце кода четырех единиц.

Длиной полной группы единиц условимся называть число образующих ее единиц. Для вышеприведенного кода длины полных групп есть соответственно числа 3, 1, 2, 4.

**Т е о р е м а 2.1.** Если настроечный код, подаваемый на  $A$ -входы  $F$ -структуры, состоит из  $r$  групп двоичных знаков, каждая из которых содержит полную группу единиц и один примыкающий к ним слева нуль, то  $F$ -структура реализует формулу

$$\begin{aligned} f_i &= S_{2,3,\dots,n_1}(T \in N_1) \vee S_{2,3,\dots,n_2}(T \in N_2) \vee \dots \vee S_{2,3,\dots,n_r}(T \in N_r) = \\ &= \bigvee_{i=1}^r S_{2,3,\dots,r_i}(T \in N_i), \end{aligned} \quad (2.4)$$

где  $n_i$  – число двоичных знаков, образованных полной группой единиц и примыкающим к ним слева нулем;  $N_i$  – множество  $T$ -аргументов, соответствующих  $i$ -й полной группе единиц и примыкающему к ней слева нулю,  $S_{2,3,\dots,r_i}$  – симметрическая реализация БФу [6] соответствующих аргументов.

**Д о к а з а т е л ь с т в о.** Пусть дан некоторый настроечный код. Выделим в нем какую-либо полную группу единиц с примыкающим к ней слева нулем. Будем считать, что этому нулю соответствует аргу-

мент  $A_h$ , а единицам – аргументы  $A_{h+1}, A_{h+2}, \dots, A_{h+\lambda}$ , где  $\lambda$  – длина полной группы ( $h, h + \lambda \in \{1, 2, \dots, n\}$ ). Тогда  $A_h = 0$ ;

$$A_{h+1} = A_{h+2} = \dots = A_{h+\lambda} = 1.$$

Подставим значения этих аргументов в формулу  $\varphi_i$ . Тогда конъюнкции формулы  $\varphi_i$ , содержащие одиночные  $A$ -аргументы (кроме  $T$ -аргументов), дадут  $\lambda$  различных двухсимвольных конъюнкций  $T$ -аргументов. Эти  $\lambda$  конъюнкции войдут в искомую формулу  $\varphi_i$ . Кроме того, в формуле  $\varphi_i$  существует  $C_\lambda^2$  конъюнкций, включающих в себя различные комбинации  $A$ -аргументов группы  $\lambda$ . При подстановке единиц вместо этих аргументов получим еще  $C_\lambda^2$  различных двухсимвольных конъюнкций  $T$ -аргументов, которые также войдут в формулу  $f_i$ . Всего данной полной группе единиц настроечного кода соответствует  $P$  двухсимвольных конъюнкций  $T$ -аргументов, входящих в искомую формулу  $f_i$ , где

$$P = \lambda \vee C_\lambda^2.$$

Все эти  $P$  конъюнкций являются различными и образуют их  $\lambda + 1$  аргументов  $T_h, T_{h+1}, \dots, T_{h+\lambda}$ .

Из [1] известно, что число выражений вида

$$S_{2,3,\dots,\lambda+1}(T_h, T_{h+1}, \dots, T_{h+\lambda})$$

равно  $Q$  где  $Q = C_{\lambda+1}^2$ .

Поскольку  $P = Q$ , то множество двухсимвольных конъюнкций, полученных на основе полной группы настроечного кода, равно множеству всех конъюнкций симметрической реализации БФу тех же аргументов, представленной в минимальной ДНФ. Так как и  $A$ -, и  $T$ -аргументы, входящие в различные полные группы единиц настроечного кода, не повторяются, то каждой полной группе единиц соответствует определенная симметрическая реализация БФу соответствующих аргументов. Пусть  $\lambda_i$  – длина  $i$ -й полной группы единиц ( $i = 1, 2, \dots, r$ ), тогда

$$n_i = \lambda_i + 1$$

и формула, которая реализуется  $F$ -структурой при помощи настроечного кода, содержащего  $r$  полных групп единиц, представится в виде

$$\begin{aligned} f_i &= S_{2,3,\dots,n_1}(T \in N_1) \vee S_{2,3,\dots,n_2}(T \in N_2) \vee \dots \vee S_{2,3,\dots,n_r}(T \in N_r) = \\ &= \bigvee_{i=1}^r S_{2,3,\dots,r_i}(T \in N_i), \end{aligned}$$

что и требовалось доказать.

**С л е д с т в и е 1.** Если полные группы единиц отделены одна от другой  $K$  нулями, то в минимальной ДНФ формулы  $f_i$  отсутствуют  $K - 1$   $T$ -аргументов. Справедливость этого следствия вытекает из того, что  $T$ -аргументы, образующие каждую симметрическую реализацию БФу выражения (2.4), определяются только единицами настроечного кода и одиночными нулями, примыкающими слева к каждой полной группе единиц.  $T$ -аргументы, соответствующие остальным нулям, в формулу  $f_i$  не входят.

**С л е д с т в и е 2.** Если настроечный код оканчивается  $m$  нулями, то последние  $m$   $T$ -аргументов в формулу  $f_i$  не войдут.

**С л е д с т в и е 3.** Если настроечный код начинается с единицы, то полной первой группе, насчитывающей  $\lambda$  единиц, соответствует дизъюнкция  $T$ -аргументов, входящая в формулу  $f_i$ :

$$f_i = \bigvee_{i=1}^{\lambda} T_i \vee \xi,$$

где  $\xi$  – дизъюнкция симметрических реализаций БФу  $T$ -аргументов, соответствующих остальным полным группам единиц настроечного кода.

Для доказательства этого следствия достаточно заметить, что если настроечный код начинается с единицы, то

$$A_1 = A_2 = \dots = A_{\lambda},$$

где  $\lambda$  – длина первой полной группы единиц. Тогда

$$f_i = T_1 \vee T_2 \vee \dots \vee T_{\lambda} \vee \xi = \bigvee_{i=1}^{\lambda} T_i \vee \xi,$$

что и требовалось доказать.

**С л е д с т в и е 4.** Если  $K_1$  и  $K_2$  – различные настроечные коды, то соответствующие им формулы, реализуемые  $F$ -структурой, на  $A$ -входы которой подаются коды  $K_1$  и  $K_2$ , не совпадают, что следует из выражения (2.4), если учесть, что

$$N_i \cap N_j = 0,$$

где  $i \neq j$ ;  $i, j = 1, 2, \dots, r$ .

**П р и м е р.** Найти формулу, которую реализует  $F$ -структура на основе кода 0 1 0 0 1 1 1 0, поданного на  $A$ -входы этой структуры.

Согласно записи кода в нем имеется две полные группы единиц. Первая состоит из одной единицы. Добавив к ней нуль, расположенный слева, находим, что в искомую формулу  $f_8$  войдет:

$$S_2(T_1T_2) = T_1T_2.$$

Вторая полная группа состоит из трех единиц, которым соответствуют аргументы  $T_5, T_6, T_7$ . С учетом нуля, расположенного левее полной группы, находим, что в формулу  $f_8$  войдет:

$$S_{2,3,4}(T_4, T_5, T_6, T_7) = T_4T_5 \vee T_4T_6 \vee T_4T_7 \vee T_5T_6 \vee T_5T_7 \vee T_6T_7.$$

Других групп единиц в настроечном коде нет, следовательно, искомая формула  $f_8$  примет вид

$$f_8 = T_1T_2 \vee T_4T_5 \vee T_4T_6 \vee T_4T_7 \vee T_5T_6 \vee T_5T_7 \vee T_6T_7.$$

Полные группы единиц отделены двумя нулями. Согласно следствию 1 один  $T$ -аргумент в формуле  $f_8$  отсутствует. Это аргумент  $T_3$ , соответствующий второму нулю, расположенному слева от второй полной группы единиц. Настроечный код оканчивается нулем. Согласно следствию 2 в формуле  $f_8$  отсутствует аргумент  $T_8$ .

Таким образом, теорема 2.1 и следствия позволяют по настроечному коду найти формулу, которую реализует  $F$ -структура, если на ее  $A$ -входы подать этот настроечный код. При этом, согласно следствию 4, всякому настроечному коду соответствует единственная  $f$  формула.

### 2.5.1.3. Нахождение настроечных кодов по виду булевой формулы

С практической точки зрения наибольший интерес представляет вопрос о том, как найти настроечный код по заданной булевой формуле.



Очевидно, что если формула  $f$  представлена в ДНФ, не содержит повторяющихся аргументов и пропусков аргументов, то она реализуется  $F$ -структурой. Если настроечный код подается на  $T$ -входы, то будем говорить, что речь идет о прямом варианте реализации формулы. Если же настроечный код подается на  $A$ -входы, то условимся называть это обратным вариантом реализации.

Относительно всякой формулы имеют место следующие вопросы.

1. Если формула  $f$  не входит в множество  $N_1$ , рассмотренное в разделе 2.4, то входит ли она в множество  $M_1$ ?

2. Если на вопрос 1 дан положительный ответ, то как найти настроечный код?

Для ответа на первый вопрос воспользуемся доказанной теоремой 2.1. Непосредственно из формулы (2.4) следует, что всякая формула реализуется  $F$ -структурой в обратном варианте, если эта формула представима в виде дизъюнкции симметрических реализаций БФу, минимальные ДНФ которых содержат конъюнкции не более чем по два аргумента, и если множества аргументов, входящих в различные симметрические реализации БФу, не пересекаются.

Пример 1. Формула

$$f = A \vee B \vee CD \vee CE \vee DE \quad (2.5)$$

реализуется в обратном варианте (и не реализуется в прямом, поскольку в ней есть повторяющиеся аргументы), так как можно записать

$$f = A \vee B \vee S_{2,3}(C, D, E) = S_{1,2}(A, B) \vee S_{2,3}(C, D, E).$$

Пример 2. Формула

$$f = A \vee B \vee CD \vee CE \vee CF \vee DE \vee DF$$

не реализуется  $F$ -структурой ни в прямом, ни в обратном вариантах, поскольку она не сводится к дизъюнкции симметрических реализаций БФу, множества аргументов которых не пересекаются.

Ответ на второй вопрос не является однозначным. Если известно, что формула реализуема в обратном варианте, то можно найти несколько различных настроечных кодов, каждый из которых можно использовать для настройки  $F$ -структуры на заданную формулу. Эта неодно-

значность нахождения кодов обусловлена следствием 1, т.е. в зависимости от того, какому входу  $F$ -структуры поставлен в соответствие тот или иной  $T$ -аргумент, настроечный код примет тот или иной вид. Если соответствие между входами  $F$ -структуры и аргументами заданной формулы зафиксировано, то настроечный код находится однозначно. Рассмотрим, например, формулу (2.5). Пусть соответствие между ее аргументами имеет вид

$$\begin{array}{ccccc} A & B & C & D & E \\ 1 & 2 & 3 & 4 & 5, \end{array}$$

где цифры являются номерами  $T$ -входов. Тогда для 8-входовой  $F$ -структуры получаем настроечный код 11011000. Выберем другое соответствие:

$$\begin{array}{ccccc} A & B & C & D & E \\ 1 & 2 & 5 & 6 & 7, \end{array}$$

получим настроечный код 11000110.

Рассмотрим еще пример. Найти настроечный код для формулы

$$f = AB \vee CD \vee FK \vee LM$$

при условии, что аргумент  $E$  является пропущенным.

Хотя формула представлена в ДНФ и является неповторной, реализовать ее в прямом варианте невозможно из-за пропуска аргумента  $E$ . Установим соответствие для 10-входовой  $F$ -структуры:

$$\begin{array}{cccccccccc} A & B & C & D & E & F & K & L & M & N \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10. \end{array}$$

Настроечный код имеет вид 01010001010, которому соответствует формула, выраженная через  $T$ -аргументы:

$$f_{10} = T_1 T_2 \vee T_3 T_4 \vee T_6 T_7 \vee T_8 T_9,$$

где отсутствует аргумент  $T_5$ .

Таким образом, благодаря обратному варианту реализации формул возможности  $F$ -структуры значительно расширились.

**2.5.2. Вычисление неповторных булевых формул с пропусками аргументов**

*2.5.2.1. Операция удаления аргументов из булевой формулы*

Рассмотренный в разделе 2.5.1 МЛМ не обеспечивает реализацию неповторных упорядоченных булевых формул классов 3, 4, 5, 6 из приведенной классификации.

Рассмотрим цепь ячеек, приведенных на рис. 2.6, которая представляет собой линейную изотропную (однородную) среду. Так как за каждой ячейкой изотропной среды закреплен вполне определенный логический аргумент, возникает вопрос, что значит не учитывать тот или иной аргумент при вычислении заданной формулы?

Например, формула

$$f = A_1 \vee A_3 A_4$$

не зависит от аргумента  $A_2$ . Схемно это будем осуществлять путем электрического соединения входов и выходов ячейки  $A_2$ , как показано пунктиром на рис. 2.6.

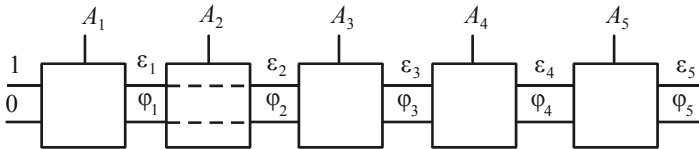


Рис. 2.6. Линейная изотропная среда

Математически такое соединение ячеек соответствует некоторой операции, которая в случае неповторной формулы сводится к подстановке в нее нуля или единицы вместо удаляемого аргумента. Пусть, например, при  $n = 5$   $F$ -структура настроена на формулу

$$f_1 = A_1 A_2 \vee A_3 A_4 \vee A_5.$$

Если принять  $A_3 = 1$ , то получим новую формулу

$$f'_1 = A_1 A_2 \vee A_4 \vee A_5,$$

на  $F$ -структуре ее можно реализовать лишь в том случае, если в нее ввести код 01011 и удалить ячейку 3.

Рассмотрим другую формулу:

$$f_2 = A_1 A_2 \vee A_3 \vee A_4 A_5.$$

Если по аналогии с формулой  $f_1$  в формулу  $f_2$  подставить  $A_3 = 1$ , чтобы удалить аргумент  $A_3$ , то получим

$$f_2' = A_1 A_2 \vee 1 \vee A_4 A_5 \equiv 1$$

вместо ожидаемого результата

$$f_2'' = A_1 A_2 \vee A_4 A_5,$$

который получился бы при вводе в  $F$ -структуру кода 01101 с последующим удалением ячейки 3.

Отсюда следует, что для неповторных упорядоченных формул операции удаления аргумента соответствует подстановка вместо него единицы, если он входит в конъюнкцию с другими аргументами, и нуля, если он входит в дизъюнкцию. Однако это справедливо лишь для формул, не содержащих инверсных аргументов. В последнем случае, прежде чем выполнять подстановку, необходимо выяснить, не является ли инверсным удаляемый аргумент [19]. Например, чтобы удалить аргумент  $A_2$  в формуле

$$f = A_1 A_2 \vee A_3,$$

необходимо выполнить подстановку  $A_2 = 1$ . Чтобы удалить тот же аргумент в формуле

$$f = A_1 \bar{A}_2 \vee A_3,$$

необходима подстановка  $A_2 = 0$ .

В общем случае, когда формула содержит повторные аргументы, принцип подстановки не применим. Поясним это на примере формулы

$$f = A_1 A_2 \vee A_2 A_3. \quad (2.6)$$

Чтобы из (2.6) получить формулу

$$f = A_1 A_2 \vee A_3, \quad (2.7)$$

необходимо вместо второго вхождения аргумента  $A_2$  подставить единицу, а первое вхождение оставить без изменения. Очевидно, что

подстановкой в формулу (2.6)  $A_2 = 1$  выражение (2.7) получить невозможно. При  $A_2 = 0$  получаем  $f \equiv 0$ .

В дальнейшем будем полагать, что операции удаления аргумента соответствует не подстановка вместо него какого-либо значения, а его *вычеркивание* без изменения всех логических операций, связывающих аргументы функции, при этом инверсные аргументы удаляются вместе со знаками отрицания. Если аргумент входит в формулу многократно, то вычеркиванию подлежит не сам аргумент, а его вхождение в формулу.

Например, формула

$$f = A_1 A_2 A_3 \vee A_2 A_4 \vee A_2 A_5$$

имеет семь вхождений аргументов. Чтобы из нее получить формулу

$$f = A_1 A_2 A_3 \vee A_4 \vee A_5,$$

необходимо удалить второе и третье вхождение аргумента  $A_2$ , а первое оставить без изменения.

В случае неповторных упорядоченных формул аргумент и его вхождение неразличимы.

Таким образом, операцию вычеркивания аргумента реализует принцип удаления ячейки, основанный на создании полной прямой проводимости между входами удаляемой ячейки и соответствующими ее выходами.

### 2.5.2.2. Реализация операции удаления аргумента из формулы

Структура рассматриваемого МЛМ (рис. 2.7) описывается системой вида [10]:

$$\begin{cases} \varepsilon_i = (A_i \varepsilon_{i-1} \vee T_i) \bar{S}_i \vee \varepsilon_{i-1} S_i, \\ \varphi_i = A_i \varepsilon_{i-1} T_i \bar{S}_i \vee \varphi_{i-1}. \end{cases} \quad (2.8)$$

Если на ее основе построить линейную среду ( $S$ -структуру), то получим возможность вычислять неповторные булевы формулы, аргументы которой упорядочены, но некоторые из них отсутствуют, например:

$$f_1 = A_1 A_2 \vee A_4 A_6.$$

Здесь аргументы  $A_3$  и  $A_5$  удалены, однако в  $\alpha$ -коде их необходимо учесть так, как будто они входят в формулу, т.е.

$$\alpha = \begin{array}{cccccc} 0 & 1 & 0 & 0 & 0 & 1 \\ A_1 & A_2 & A_3 & A_4 & A_5 & A_6. \end{array}$$

Код, содержащий информацию об удаленных аргументах и задаваемый с помощью  $S$ -триггеров, будем называть  $S$ -кодом. В данном случае  $S$ -код имеет вид

$$S = \begin{array}{cccccc} 0 & 0 & 1 & 0 & 1 & 0 \\ A_1 & A_2 & A_3 & A_4 & A_5 & A_6, \end{array}$$

где единицами обозначены отсутствующие аргументы.

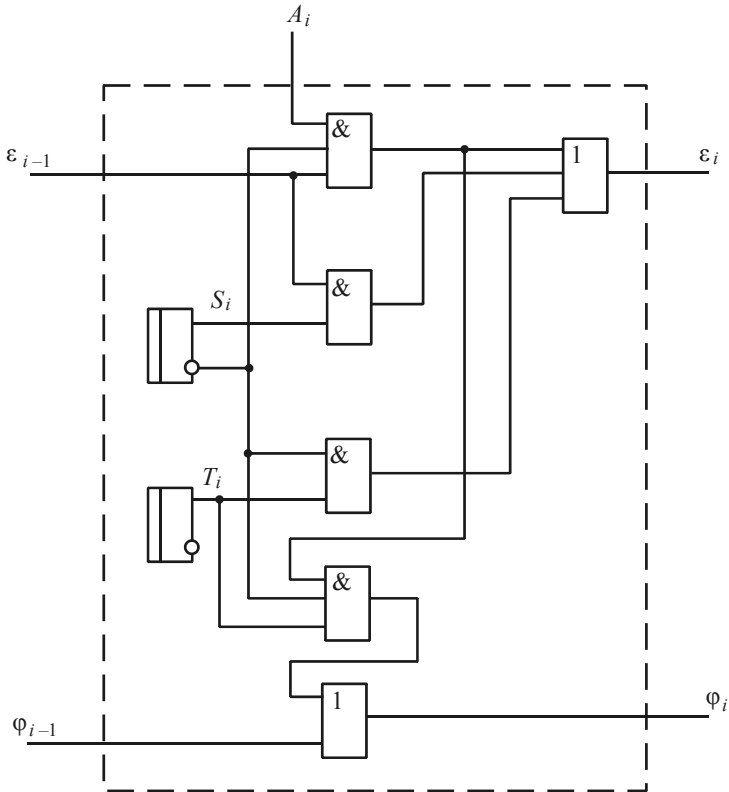


Рис. 2.7. МЛМ  $S$ -структуры

Если формула неповторна, но аргументы в ее записи не упорядочены, то вычислить ее значение с помощью  $S$ -структуры не всегда воз-

можно. Например, формулу  $f = A_5 A_4 \vee A_2 A_1 \vee A_6$  можно вычислить, если записать ее в виде  $f = A_1 A_2 \vee A_4 A_5 \vee A_6$ , соответствующие  $\alpha$ - и  $S$ -коды имеют вид  $\alpha = 010011$ ,  $S = 001000$ . Формулу  $f = A_1 A_3 \vee A_2$  вычислить невозможно [14].

Сформулируем *правило*, в соответствии с которым можно было бы определить вычислимость формулы.

Пусть формула представлена неповторной ДНФ, аргументы которой не упорядочены по вхождению. Находим аргумент с наименьшим индексом и записываем всю конъюнкцию, куда входит этот аргумент, предварительно расставив аргументы в конъюнкции в порядке возрастания: слева направо. Последний аргумент будет иметь индекс  $t$ . Если в оставшейся части формулы найдется хотя бы один аргумент с индексом, меньшим  $t$ , то формула не вычислима  $S$ -структурой. Если же таких нет, то в оставшейся части формулы находим аргумент с наименьшим индексом и выписываем вторую конъюнкцию так же, как и в первом случае.

Таким образом, с помощью  $S$ -структуры осуществляется вычисление неповторных упорядоченных булевых формул, представленных в ДНФ или КНФ как с пропусками аргументов, так и без них.

### 2.5.2.3. О неоднозначности записи настроечных кодов в $S$ -структуре

Рассмотрим формулу

$$f = A_1 A_2 \vee A_4 A_6, \quad (2.9)$$

ее настроечные коды имеют вид

$$\alpha_1 = 010001,$$

$$S = 001010.$$

Поскольку аргументы  $A_3$  и  $A_5$  удалены, то формально  $\alpha$ -код можно записать в виде двух других вариантов:

$$\alpha_2 = 001001,$$

$$\alpha_3 = 011001.$$

Эта неоднозначность записи  $\alpha$ -кодов обусловлена тем, что формула (2.9) может быть получена путем вычеркивания аргументов  $A_3$  и  $A_5$  из следующих трех различных формул:

$$f_1 = A_1 A_2 \vee A_3 A_4 A_5 A_6,$$

$$f_2 = A_1 A_2 A_3 \vee A_4 A_5 A_6,$$

$$f_3 = A_1 A_2 \vee A_3 \vee A_4 A_5 A_6.$$

Однако применительно к  $S$ -средам это справедливо лишь для формул  $f_1$  и  $f_3$ . Если аргументы  $A_3$  и  $A_5$  вычеркнуть из формулы  $f_2$ , то формально получается формула (2.9), а при вводе в  $S$ -структуру настроечных кодов

$$\alpha_2 = 001001,$$

$$S = 001010$$

будет реализовываться

$$f = A_1 A_2 A_3 A_6,$$

не совпадающая с выражением (2.9).

Пусть  $\Psi$  – формула, которая получилась на основе некоторой формулы  $f$  путем вычеркивания из нее определенных аргументов;  $\Psi_2$  – формула, реализуемая  $S$ -структурой, настроенной  $\alpha$ - и  $S$ -кодами, где  $\alpha$ -код записан на основе формулы  $f$ , а  $S$ -код отражает отсутствующие аргументы. Выясним, в каких случаях

$$\Psi_1 \neq \Psi_2.$$

Ответ на этот вопрос дают следующие теоремы.

**Т е о р е м а 2.2.** Если в записи некоторой неповторной упорядоченной булевой формулы  $f$ , представленной в виде дизъюнкции  $k$  конъюнкций, пронумерованных в последовательности  $1, 2, \dots, k$ :

$$f = \tau_1 \vee \tau_2 \vee \dots \vee \tau_k,$$

удалить последний аргумент в  $i$ -й конъюнкции, где  $i = 1, 2, \dots, k - 1$ , и записать  $S$ -код, то при вводе в  $S$ -структуру этого  $S$ -кода и  $\alpha$ -кода, записанного на основе формулы  $f$ ,  $S$ -структура настроится на формулу  $\Psi$ , отличающуюся от формулы  $f$  тем, что в ней вместо дизъюнкции  $\tau_{j-1} \vee \tau_j$  ( $j = 1, 2, \dots, k$ ) будет записана конъюнкция  $t_{j-1} \tau_j$ , где  $t_{j-1}$  – конъюнкция, полученная на основе конъюнкции  $\tau_{j-1}$  путем удаления последнего аргумента.

**Д о к а з а т е л ь с т в о.** Пусть задана неповторная упорядоченная булева формула  $n$  аргументов

$$f = \tau_1 \vee \tau_2 \vee \dots \vee \tau_k,$$



где

$$\begin{aligned} \tau_1 &= A_1 A_2 \dots A_{S_1-1} A_{S_1}, \\ \tau_2 &= A_{S_1+1} A_{S_1+2} \dots A_{S_2-1} A_{S_2}, \\ \tau_3 &= A_{S_2+1} A_{S_2+2} \dots A_{S_3-1} A_{S_3}, \\ &\dots \\ \tau_k &= A_{S_{k-1}+1} A_{S_{k-1}+2} \dots A_{S_k-1} A_{S_k}. \end{aligned}$$

Найдем для нее  $\alpha$ -код. Затем удалим последний аргумент в  $i$ -й конъюнкции и определим соответствующий  $S$ -код. Оба найденные кода введем в  $S$ -структуру. Благодаря  $S$ -коду ячейка  $A_{S_i}$  будет исключена.

Выходы ячейки с номером  $S_{i-1}$  соединятся с соответствующими входами ячейки с номером  $S_{i+1}$ . В разрядах  $\alpha$ -кода с указанными номерами содержатся нули, обозначающие операцию конъюнкции соответствующих аргументов. Следовательно, вместо дизъюнкции вида  $\tau_{j-1} \vee \tau_j$  в формуле  $f$  окажется конъюнкция  $t_{j-1} \tau_j$ , где

$$t_{j-1} = A_{S_{j-1}+1} A_{S_{j-1}+2} \dots A_{S_{j-1}},$$

что и требовалось доказать.

**Т е о р е м а 2.3.** Если формула представлена единственной конъюнкцией, то она всегда вычислима и имеет  $2^r$  вариантов  $\alpha$ -кода, где  $r$  – число единиц в  $S$ -коде, оканчивающемся нулем.

**Д о к а з а т е л ь с т в о.** Пусть задана формула

$$f = A_1 A_2 \dots A_{k-1} A_{k+1} A_{k+2} \dots A_n,$$

где  $n > k$ . Если аргументы формулы не упорядочены, то, расположив их в порядке возрастания индексов, получим вычислимую запись. Ее  $S$ -код имеет вид

$$\underbrace{000\dots 01}_{k-1 \text{ нулей}} \underbrace{00\dots 0}_{n-k \text{ нулей}},$$

$\alpha$ -код формулы представится в виде

$$\alpha = \underbrace{00000\dots 001}_{n-1 \text{ нулей}}.$$

Согласно теореме 2.1  $\alpha$ -код можно записать и в виде

$$\alpha = \underbrace{000\dots 01}_{k-1 \text{ нулей}} \underbrace{1000\dots 01}_{n-k \text{ нулей}},$$

что соответствует выражению

$$f^n = A_1 A_2 \dots A_k \vee A_{k+1} \dots A_n,$$

где аргумент  $A_k$ , являющийся последним в первой конъюнкции, удален.

Все сказанное справедливо для любой ячейки, следовательно, если в  $S$ -коде содержится  $k$  единиц, причем  $S$ -код оканчивается нулем, то  $r$  триггеров, не влияющих на значение формулы, могут находиться в одном из  $2^r$  состояний, что и требовалось доказать.

**Т е о р е м а 2.4.** Если  $S$ -код совпадает с  $\alpha$ -кодом, то ячеистая структура реализует формулу  $f \equiv 0$ .

**Д о к а з а т е л ь с т в о.** Если длина ячеистой структуры равна единице, то формула, описывающая ее выход, имеет вид

$$\varphi_1 = A_1 T_1 \bar{S}.$$

Это выражение можно получить из (2.8), если учесть, что  $\varepsilon_0 = 1$  и  $\varphi_0 = 1$ . Очевидно, что при  $S_1 = T_1$  имеем  $\varphi_1 \equiv 0$ .

Пусть длина структуры равна двум ячейкам. Выход ее представится в виде

$$\varphi_2 = A_2 \varepsilon_1 T_2 \bar{S}_2 \vee \varphi_1.$$

При  $T_2 = S_2$  первая конъюнкция равна нулю, а при  $T_1 = S_1$ , как было показано выше,  $\varphi_1 \equiv 0$ , следовательно,  $\varphi_2 \equiv 0$ .

Рассуждая аналогично, замечаем, что в формуле

$$\varphi_i = A_i \varepsilon_{i-1} T_i \bar{S}_i \vee \varphi_{i-1},$$

где  $i = 1, 2, \dots, n$ ,  $n$  – число ячеек, первая конъюнкция равна нулю, так как  $T_i = S_i$ , а вторая равна нулю по причине  $T_j = S_j$  ( $j = 1, 2, \dots, i-1$ ). Следовательно, если  $\alpha$ -код и  $S$ -код равны, то структура реализует формулу, тождественно равную нулю, что и требовалось доказать.

**С л е д с т в и е и з т е о р е м ы 2.4.** Если  $n$  – длина ячеистой структуры, то формула  $f \equiv 0$  может быть представлена  $2^n$  различными  $\alpha$ -кодами и  $2^n$   $S$ -кодами при равенстве  $S$ - и  $\alpha$ -кодов. Справедливость этого утверждения следует из того, что всего существует  $2^n$   $n$ -разрядных двоичных чисел, каждое из которых может быть  $\alpha$ - и  $S$ -кодами.

Формулу  $f \equiv 0$  в общем случае можно задать многими способами. Кроме уже указанного случая, когда  $\alpha = S$ , существуют и другие варианты. Например, если  $\alpha = 0$ , т.е.  $\alpha$ -код не содержит единиц, то структура реализует формулу  $f \equiv 0$  независимо от  $S$ -кода. Это значит, что при

$\alpha = 0$  существует  $2^n$  вариантов настройки  $S$ -структуры на формулу  $f \equiv 0$ , образуемых путем изменения  $S$ -кода. Если  $S$ -код не содержит нулей, то  $S$ -структура также реализует формулу  $f \equiv 0$   $2^n$  способами, каждому из которых соответствует определенный  $\alpha$ -код. Справедливость этого утверждения следует из того, что при  $S$ -коде, не содержащем нулей, выход структуры соединен с входом первой ячейки, на который постоянно подан нулевой уровень напряжения. Общий случай распознавания формул  $f \equiv 0$  по виду настроечного кода освещен в следующем разделе.

#### 2.5.2.4. Распознавание формулы константа нуль по виду настроечных кодов

В общем случае варианты настройки  $S$ -структуры на формулу  $f \equiv 0$  определяются нижеприведенной теоремой. Но сначала введем некоторые понятия. Формула, описывающая  $S$ -структуру, зависит от служебных аргументов  $T_i$  и  $S_i$ , каждому из которых соответствует отдельный триггер. Будем рассматривать эти триггеры как двухразрядный двоичный регистр, где  $T_i$  соответствует старшему разряду, а  $S_i$  – младшему. В каждом регистре может быть записано одно из четырех чисел 00, 01, 10, 11, которые рассмотрим как цифры четверичной системы счисления 0, 1, 2, 3. Тогда каждому  $n$ -разрядному четверичному числу будет соответствовать некоторая булева формула, на которую настроится  $S$ -структура, и всякий настроечный код можно представить в четверичной системе.

**Т е о р е м а 2.5.** Если в настроечном коде отсутствует цифра 2, то структура реализует формулу  $f \equiv 0$ .

**Д о к а з а т е л ь с т в о.** Пусть дана структура, настроечный код которой содержит цифры 0, 1, 3 и не содержит цифру 2. Ячейки, в которые введены цифры 1 и 3, можно удалить, поскольку их входы соединены с соответствующими выходами. Тогда останется структура, настроечный код которой состоит из одних нулей, т.е. коды  $S$  и  $\alpha$  будут равными. Согласно теореме 2.4 такая структура реализует формулу  $f \equiv 0$ , что и требовалось доказать.

**Т е о р е м а 2.6.** Если в настроечный код входит цифра 2, то структура реализует формулу  $f \neq 0$ .

**Д о к а з а т е л ь с т в о.** Пусть дана структура, настроечный код которой содержит все четверичные цифры. Удалим ячейки, содержащие цифры 1 и 3. Тогда останется структура, содержащая настроечные коды

0 и 2. Такая структура совпадает с  $F$ -структурой. Согласно правилам нахождения  $\alpha$ -кодов единица в коде обозначает конец конъюнкции (ее последнюю букву). Следовательно, присутствие единицы в  $\alpha$ -коде говорит о наличии соответствующей конъюнкции в заданной формуле, тождественно не равной нулю.

Таким образом, если в настроечном коде имеется цифра 2, то  $S$ -структура реализует формулу  $f \neq 0$ , что и требовалось доказать.

**С л е д с т в и е 1** из теорем 2.5 и 2.6.  $S$ -структура, состоящая из  $n$  ячеек, реализует  $k$  тождественно равных нулю булевых формул, где  $k = 3^n$ , и  $N$  тождественно не равных нулю булевых формул, где

$$N = 4^n - 3^n. \quad (2.10)$$

Справедливость этих утверждений следует из того, что настроечные коды, приводящие к формуле  $f \equiv 0$ , могут состоять из последовательности длины  $n$ , образованной тремя цифрами 0, 1, 3. Поставим их в соответствие цифрам троичной системы счисления. Тогда каждое  $n$ -разрядное троичное число будет представлять некоторый настроечный код для  $f \equiv 0$ . Количество всех троичных  $n$ -разрядных чисел равно  $3^n$ , следовательно, и  $k = 3^n$ . Всего существует  $4^n$  настроечных кодов, следовательно, число настроечных кодов для  $f \neq 0$  равно  $4^n - 3^n$ , что и требовалось доказать.

**С л е д с т в и е 2.** Если настроечный код состоит только из двоек, то  $S$ -структура реализует дизъюнкцию  $n$ -аргументов.

**С л е д с т в и е 3.** Если настроечный код состоит из  $n - 1$  нулей и оканчивается двойкой, то  $S$ -структура реализует конъюнкцию  $n$ -аргументов.

**С л е д с т в и е 4.** Если в  $i$ -м разряде настроечного кода находится двойка, а в разрядах с номерами  $i + 1, i + 2, \dots, n$  двоек нет, то  $S$ -структура реализует формулу, зависящую не более чем от  $i$  аргументов.

#### 2.5.2.5. Об избыточности в кодировании булевых формул

В данном разделе покажем, что количество неповторных упорядоченных булевых формул, реализуемых  $S$ -структурой, вследствие неоднозначности представления их настроечными кодами значительно меньше числа  $N$ , приведенного в выражении (2.10). Например, при  $n = 2$  имеем

$$N = 4^2 - 3^2 = 7.$$

В приложении 1 (П.1.1) приведены все варианты реализации формул двух аргументов, из которой видно, что среди семи тождественно не равных нулю формул лишь четыре являются различными:

$$\begin{aligned} f_1 &= A_1 A_2; \quad f_3 = A_1; \\ f_2 &= A_2; \quad f_4 = A_1 \vee A_2. \end{aligned}$$

При  $n = 3$  имеем

$$N = 4^3 - 3^3 = 37.$$

В приложении 1 (П.1.2) приведены все 64 настроечных кода. Находим, что существует 13 различных, тождественно не равных нулю формул трех аргументов, которые могут быть реализованы  $S$ -структурой при  $n = 3$ :

$$f_1 = A_1 (200, 201, 203, 210, 230, 211, 213, 231, 233);$$

$$f_2 = A_2 (120, 320, 121, 123, 321, 323);$$

$$f_3 = A_3 (112, 132, 312, 332);$$

$$f_4 = A_1 A_2 (020, 021, 023);$$

$$f_5 = A_1 A_3 (012);$$

$$f_6 = A_2 A_3 (102, 302);$$

$$f_7 = A_1 A_2 A_3 (002);$$

$$f_8 = A_1 \vee A_2 (220, 221, 223);$$

$$f_9 = A_1 \vee A_3 (032, 212, 232);$$

$$f_{10} = A_2 \vee A_3 (122, 322);$$

$$f_{11} = A_1 \vee A_2 \vee A_3 (222);$$

$$f_{12} = A_1 \vee A_2 A_3 (202);$$

$$f_{13} = A_1 A_2 \vee A_3 (022).$$

В скобках перечислены настроечные коды для каждой из 13 формул. Коды представлены в четверичной системе счисления.

Выясним, сколько существует различных формул, которые может реализовать  $S$ -структура, состоящая из  $n$  ячеек, т.е. выведем формулу, связывающую величины  $N$  и  $n$ .

Пусть даны  $n$  аргументов  $A_1, A_2, \dots, A_n$ . Выделим из них группу, содержащую  $i$  аргументов,  $i = 1, 2, \dots, n$ . Количество таких групп равно  $C_n^i$  [2]. Упорядочим аргументы в группе. Чтобы образовать из них формулу, между соседними аргументами необходимо поставить знак дизъюнкции либо конъюнкции.

В последовательности, содержащей  $i$  аргументов, существует  $i - 1$  промежутков, куда можно записывать знаки конъюнкции или дизъюнкции. Следовательно,  $i$  аргументов дадут  $2^{i-1}$  различных формул для одной группы аргументов. А так как число групп  $C_n^i$ , то при  $i$  аргументах получим  $2^{i-1} C_n^i$  соответствующих формул. Поскольку  $i = 1, 2, \dots, n$ , то общее число  $N$  различных функций равно:

$$N = 2^0 C_n^1 + 2^1 C_n^2 + 2^2 C_n^3 + \dots + 2^{n-1} C_n^n = \sum_{i=1}^n 2^{i-1} C_n^i.$$

Полученная формула не только дает количество различных бесповторных булевых формул с упорядоченными аргументами, но и предоставляет возможность записать все такие формулы. Для примера запишем все формулы при  $n = 4$ :

$$N = C_4^1 + 2C_4^2 + 4C_4^3 + 8C_4^4 = 40.$$

Первое слагаемое говорит о том, что соответствующие ему формулы содержат по одному аргументу:

$$f_1 = A_1; f_3 = A_3;$$

$$f_2 = A_2; f_4 = A_4.$$

Второму слагаемому выражения соответствуют формулы, зависящие от двух аргументов. Число выборов двух аргументов из четырех равно  $C_4^2 = 6$ . Два аргумента дают только один промежуток для записи

знаков дизъюнкции и конъюнкции, следовательно, второе слагаемое дает 12 функций:

$$f_5 = A_1 A_2; f_{11} = A_1 \vee A_2;$$

$$f_6 = A_1 A_3; f_{12} = A_1 \vee A_3;$$

$$f_7 = A_1 A_4; f_{13} = A_1 \vee A_4;$$

$$f_8 = A_2 A_3; f_{14} = A_2 \vee A_3;$$

$$f_9 = A_2 A_4; f_{15} = A_2 \vee A_4;$$

$$f_{10} = A_3 A_4; f_{16} = A_3 \vee A_4.$$

Третьему слагаемому соответствуют формулы, зависящие от трех аргументов. Три аргумента можно выбрать  $C_4^3 = 4$  способами. Между тремя аргументами имеется два промежутка, куда можно записывать знаки дизъюнкции и конъюнкции. Следовательно, имеем 16 формул:

$$f_{17} = A_1 A_2 A_3; f_{25} = A_1 \vee A_2 A_3;$$

$$f_{18} = A_1 A_2 A_4; f_{26} = A_1 \vee A_2 A_4;$$

$$f_{19} = A_1 A_3 A_4; f_{27} = A_1 \vee A_3 A_4;$$

$$f_{20} = A_2 A_3 A_4; f_{28} = A_2 \vee A_3 A_4;$$

$$f_{21} = A_1 A_2 \vee A_3; f_{29} = A_1 \vee A_2 \vee A_3;$$

$$f_{22} = A_1 A_2 \vee A_4; f_{30} = A_1 \vee A_2 \vee A_4;$$

$$f_{23} = A_1 A_3 \vee A_4; f_{31} = A_1 \vee A_3 \vee A_4;$$

$$f_{24} = A_2 A_3 \vee A_4; f_{32} = A_2 \vee A_3 \vee A_4.$$

Четвертому слагаемому соответствуют формулы, в которые входят все четыре аргумента. Для четырех аргументов существует 8 способов соединения их знаками дизъюнкции и конъюнкции:

$$f_{33} = A_1 A_2 A_3 A_4; f_{37} = A_1 \vee A_2 A_3 A_4;$$

$$f_{34} = A_1 A_2 A_3 \vee A_4; f_{38} = A_1 \vee A_2 A_3 \vee A_4;$$

$$f_{35} = A_1 A_2 \vee A_3 A_4; f_{39} = A_1 \vee A_2 \vee A_3 A_4;$$

$$f_{36} = A_1 A_2 \vee A_3 \vee A_4; f_{40} = A_1 \vee A_2 \vee A_3 \vee A_4.$$

Других формул, зависящих от четырех аргументов, являющихся неповторными и записанных в ДНФ, с упорядоченными аргументами не существует.

Вычислим величину  $N$  и найдем число настроечных кодов и количество не равных нулю формул для значений  $n = 1, 2, 3, 4, 5, 6$  (табл. 2.1); рассмотренное кодирование булевых формул обладает значительной избыточностью.

Таблица 2.1

$n$	Число настроечных кодов $4^n$	Число тождественно не равных нулю формул $4^n - 3^n$	Число различных формул $N = \sum_{i=1}^n 2^{i-1} C_n^i$
1	4	1	1
2	16	7	4
3	64	37	13
4	256	175	40
5	1024	791	121
6	4096	3367	364

Например, при  $n = 6$  из всех возможных 4096 настроечных кодов 729 являются неиспользуемыми, поскольку они настраивают  $S$ -структуру на вычисление одной и той же формулы  $f \equiv 0$ . Все оставшиеся 3367 настроечных кодов делятся на 364 подмножества, где каждое подмножество соответствует определенной формуле, на которую может быть настроена  $S$ -структура. Все настроечные коды, принадлежащие одному и тому же множеству, настраивают  $S$ -структуру на одну и ту же формулу.

#### 2.5.2.6. Нахождение булевой формулы по настроечному коду

Если в классе ДНФ задана неповторная упорядоченная булева формула  $f$  без пропусков аргументов и к ней применена  $S$ -операция (вычеркивание аргументов на основе  $S$ -кода), в результате которой получена формула  $\varphi$ , то согласно теореме 2.2 по  $\alpha$ -коду формулы  $f$  и заданному  $S$ -коду  $S$ -структурой будет реализована формула  $z$ , не всегда совпа-



дающая с формулой  $\varphi$ . Сформулируем правило, позволяющее находить формулу  $z$ . Чтобы выяснить, какую формулу будет реализовывать  $S$ -структура, необходимо записать  $\alpha$ - и  $S$ -коды один под другим так, чтобы соответствующие разряды находились в одних и тех же колонках. После этого вычеркиваем разряды в  $\alpha$ -коде, соответствующие единицам  $S$ -кода. Оставшиеся разряды образуют новый код, который условимся называть  $\alpha_0$ -кодом; формула, записанная на основе  $\alpha_0$ -кода, и будет являться искомой формулой  $z$ .

**Пр и м е р 1.** Пусть даны  $\alpha$ - и  $S$ -коды:

$$\alpha = 01011011,$$

$$S = 01001010.$$

Удаляем в  $\alpha$ -коде разряды с номерами 2, 5, 7 (нумерация разрядов слева направо), тогда получим:

$$\alpha_0 = 00101.$$

Чтобы записать алгебраическое выражение формулы  $z$ , каждому разряду  $\alpha_0$ -кода необходимо поставить в соответствие аргументы согласно  $S$ -коду:

$$\alpha_0 = \begin{matrix} 0 & 0 & 1 & 0 & 1 \\ A_1 & A_3 & A_4 & A_6 & A_8. \end{matrix}$$

Искомая формула  $z$  имеет вид:

$$z = A_1 A_3 A_4 \vee A_6 A_8.$$

Если настроечный код задан в четверичной системе счисления, то для того чтобы восстановить по нему формулу, на которую настроится  $S$ -структура, необходимо вычеркнуть из настроечного кода цифры 1 и 3, которым соответствуют единицы в  $S$ -коде. Тогда в настроечном коде останутся цифры 0 и 2. Группа нулей с примыкающей к ним справа цифрой 2 обозначает конъюнкцию соответствующих аргументов. Если таких групп в настроечном коде несколько, то соответствующие конъюнкции соединяются знаками дизъюнкции.

**Пр и м е р 2.** Какую формулу реализует  $S$ -структура при настроечном коде 0121221010?

Запишем настроечный код и под его цифрами укажем соответствующие им аргументы:

$$\begin{array}{cccccccccc} 0 & 1 & 2 & 1 & 2 & 2 & 1 & 0 & 1 & 0 \\ A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10}. \end{array}$$

В коде нет троек, следовательно, вычеркиваем только единицы. Искомая формула имеет вид

$$z = A_1 A_3 \vee A_5 \vee A_6.$$

Заметим, что в найденной формуле  $z$  нет аргументов  $A_8$  и  $A_{10}$ , хотя разряды 8 и 10 в  $\alpha$ -коде не вычеркнуты. Чтобы убедиться в справедливости формулы  $z$ , достаточно отметить, что в заданном настроечном коде после шестого разряда нет цифр 2, ограничивающих длину конъюнкции в записи формулы.

**Пример 3.** Какую формулу реализует настроечный код 130013313?

Вычеркивая единицы и тройки, получим код, состоящий из двух нулей, что соответствует формуле  $z \equiv 0$ . К этому же результату придем, если будем учитывать, что согласно теореме 2.5 код, не содержащий цифры 2, настраивает  $S$ -структуру на формулу  $z \equiv 0$ .

В заключение отметим, что нахождение формулы по настроечному коду осуществляется однозначно.

### 2.5.3. Вычисление неповторных упорядоченных булевых формул выше второго порядка

Изложим кратко результаты анализа МЛМ (рис. 2.8), позволяющего строить изотропные среды для вычисления булевых формул высоких порядков.

Структура рассматриваемого МЛМ описывается следующей системой булевых формул [17, 22]:

$$\begin{cases} f_1 = y_1 \bar{z}_1 z_2 \vee (y_1 \vee z_1 \vee z_2) x \bar{z}_3, \\ f_2 = y_1 z_1 z_2 \vee (y_1 \vee \bar{z}_1 \vee z_2) y_2, \end{cases}$$

где  $z_1, z_2, z_3$  – настроечные входы МЛМ;  $x, y_1, y_2$  – информационные входы;  $f_1, f_2$  – выходы МЛМ.

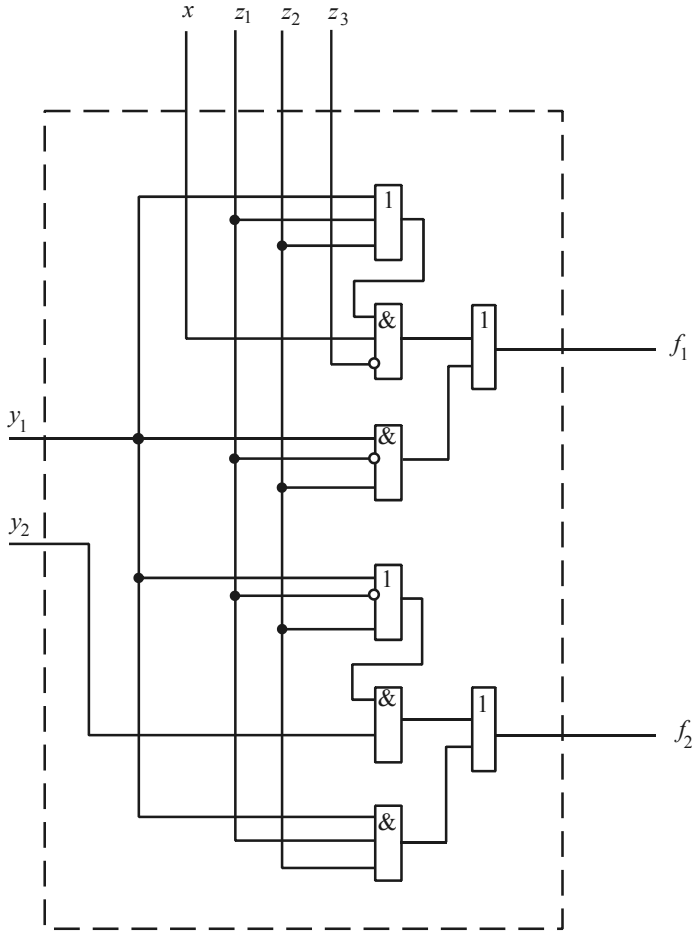


Рис. 2.8. Ячейка T-структуры

Многофункциональный логический модуль, который в дальнейшем будем называть T-ячейкой, реализует следующие системы формул:

- 1) при  $z_1 = 0, z_2 = 0, z_3 = 0$       2) при  $z_1 = 0, z_2 = 0, z_3 = 1$   
 (см. рис. 2.9, а)                              (см. рис. 2.9, б)

$$\begin{cases} f_1 = y_1x, \\ f_2 = y_2; \end{cases} \qquad \begin{cases} f_1 = 0, \\ f_2 = y_2; \end{cases}$$

3) при  $z_1 = 0, z_2 = 1, z_3 = 0$   
(рис. 2.9, в)

$$\begin{cases} f_1 = y_1 + x, \\ f_2 = y_2; \end{cases}$$

4) при  $z_1 = 0, z_2 = 1, z_3 = 1$   
(рис. 2.9, г)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2; \end{cases}$$

5) при  $z_1 = 1, z_2 = 0, z_3 = 0$   
(рис. 2.9, д)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 y_2; \end{cases}$$

6) при  $z_1 = 1, z_2 = 0, z_3 = 1$   
(рис. 2.9, е)

$$\begin{cases} f_1 = 0, \\ f_2 = y_1 y_2; \end{cases}$$

7) при  $z_1 = 1, z_2 = 1, z_3 = 0$   
(рис. 2.9, ж)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 + y_2; \end{cases}$$

8) при  $z_1 = 1, z_2 = 1, z_3 = 1$   
(рис. 2.9, з)

$$\begin{cases} f_1 = 0, \\ f_2 = y_1 + y_2. \end{cases}$$

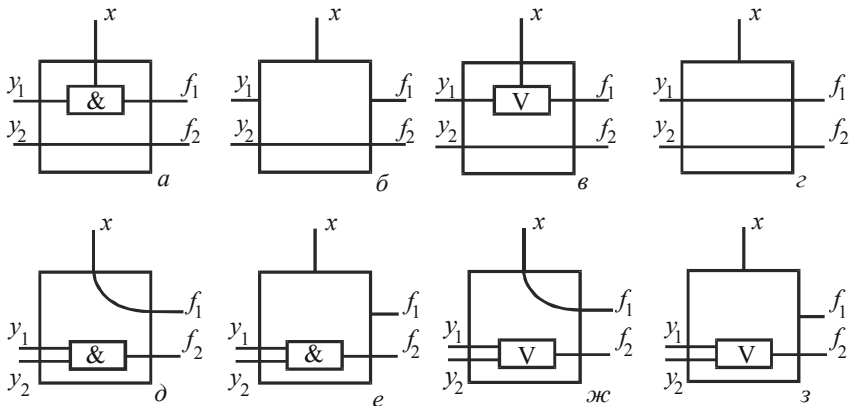


Рис. 2.9. Структурные схемы, полученные путем настройки

Таким образом, с помощью изотропной структуры ( $T$ -структуры), построенной из рассматриваемых  $T$ -ячеек, можно реализовать бесповторные упорядоченные ДНФ или КНФ, скобочные формы, а также все формулы с пропусками аргументов.

#### 2.5.4. Вычисление неупорядоченных булевых формул

Приведенные в разделах 2.5.1 – 2.5.3  $F$ -,  $S$ - и  $T$ -структуры обеспечивают вычисление только упорядоченных булевых формул. Выясним,

как можно вычислять неупорядоченные формулы, представленные в ДНФ.

Пусть задана произвольная формула  $n$  аргументов в ДНФ и пусть она имеет  $k$  конъюнкций. Очевидно, что длина конъюнкций не может превышать  $n$  букв, поскольку  $n$  – это длина минтерма (члена СДНФ) [7]. В разделе 2.5.2 было отмечено, что аргументы каждой из конъюнкций, входящих в указанную ДНФ формулы, можно упорядочить. Следовательно, заданную формулу  $n$  аргументов можно представить дизъюнкцией неповторных упорядоченных формул с пропусками аргументов:

$$f = \lambda_1 \vee \lambda_2 \vee \dots \vee \lambda_k = \bigvee_{i=1}^k \lambda_i, \quad (2.11)$$

где  $\lambda_i$  – конъюнкции заданной формулы  $f$ .

Исследуем три варианта вычисления формулы (2.11):

1) комбинационной схемой, представленной изотропной либо квазиизотропной матрицей (т.е. составленной из  $S$ - и  $T$ -ячеек);

2) многотактным автоматом путем последовательного вычисления за  $k$  тактов, когда в течение каждого такта вычисляется значение только одной конъюнкции формулы (2.11);

3) линейной изотропной структурой, построенной из МЛМ, ориентированных на вычисление как упорядоченных, так и неупорядоченных булевых формул.

**Первый вариант.** Построим изотропную матрицу на основе  $S(T)$ -структур, где символ  $S(T)$  обозначает, что структура состоит из  $S$ - либо  $T$ -ячеек (рис. 2.10).

Вместо матрицы можно рассматривать линейную  $S(T)$ -структуру, состоящую из  $n \times k$  ячеек, где входы  $A_1, A_2, \dots, A_{n+1}, A_{2n+1}, \dots, A_{(k-1)n+1}$  соединяются между собой и подключаются к устройству, моделирующему вход  $A_1$ . Входы  $A_2, A_{n+2}, A_{2n+2}, \dots, A_{(k-1)n+2}$  также соединяются между собой и объявляются входом аргумента  $A_2$  и т.д. до  $A_n$ , которому соответствует соединение  $A_n, A_{2n}, \dots, A_{kn}$ . Такую структуру целесообразно применять, если основным требованием, предъявляемым к устройству, является быстрое действие.

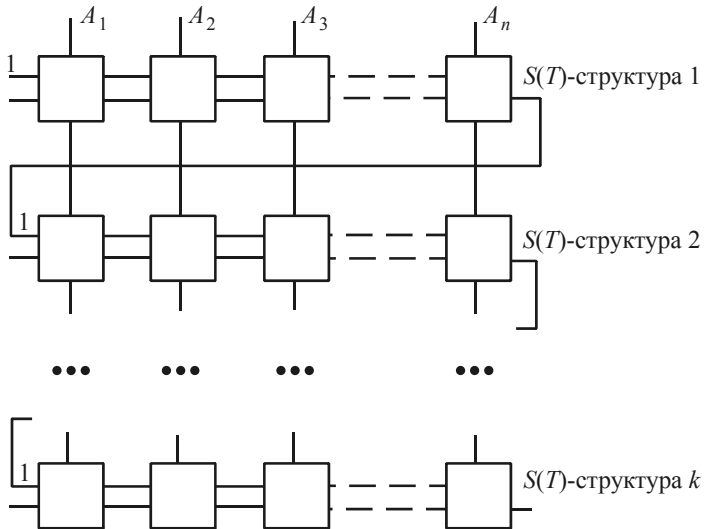


Рис. 2.10. Изотропная матрица

**Второй (многоактный) вариант.** Все настроечные коды для каждой конъюнкции  $\lambda_i$  записываются в запоминающее устройство (ЗУ), откуда последовательно поступают на настроечные входы  $S(T)$ -структуры (рис. 2.11).

С каждым тактом смены настроечных кодов на синхровход триггера  $E$  поступает импульс. Если на каком-либо настроечном коде выходной сигнал триггера  $E$  принимает единичное значение, то триггер  $E$  переходит в единичное состояние, что сигнализирует о равенстве единице формулы  $f$ . Если же после полного просмотра ЗУ триггер  $E$  остался в нулевом состоянии, то формула на данном наборе значений аргументов равна нулю. ЗУ можно организовать различными способами. Условимся считать, что каждому настроечному коду соответствует определенный адрес ЗУ. Тогда можно пронумеровать конъюнкции  $\lambda_i$  и считать, что номер этой конъюнкции совпадает с адресом, по которому в ЗУ находится ее настроечный код. Если не учитывать код  $\beta$ , которым задается распределение инверсий над аргументами, то длина выходного кода, хранящегося по заданному адресу ЗУ, равна  $2n$ . Рассмотрим пример для  $n = 7$ .

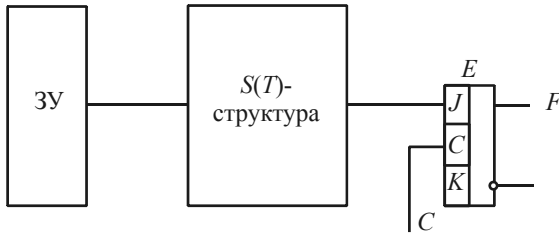


Рис. 2.11. Многотактный автомат

Пусть дана формула

$$f = A_1A_2 \vee A_1A_5A_6 \vee A_1A_3A_7 \vee A_1A_6A_7 \vee A_4A_5A_7. \quad (2.12)$$

Эта формула не является бесповторной, но каждая ее конъюнкция есть бесповторная формула с пропусками:

$$\lambda_1 = A_1A_2; \lambda_2 = A_1A_5A_6; \lambda_3 = A_1A_3A_7; \lambda_4 = A_1A_6A_7; \lambda_5 = A_4A_5A_7.$$

В ЗУ будут записаны следующие коды:

	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$	
$S =$	0	0	1	1	1	1	1,	
$\alpha =$	0	1	0	0	0	0	0	по адресу 1,
$S =$	0	1	1	1	0	0	1,	
$\alpha =$	0	0	0	0	0	1	0	по адресу 2,
$S =$	0	1	0	1	1	1	0,	
$\alpha =$	0	0	0	0	0	0	1	по адресу 3,
$S =$	0	1	1	1	1	0	0,	
$\alpha =$	0	0	0	0	0	0	1	по адресу 4,
$S =$	1	1	1	0	0	1	0,	
$\alpha =$	0	0	0	0	0	0	1	по адресу 5.

Таким образом, значение этой формулы будет определено не более чем за пять тактов. Формулу (2.12) можно вычислить и за четыре такта, если учесть, что первая и последняя конъюнкции образуют бесповторную упорядоченную формулу с пропусками аргументов  $A_3, A_6$ :

$$\lambda_1 = A_1 A_2 \vee A_4 A_5 A_7,$$

$S$ - и  $\alpha$ -коды (см. разд. 2.5.2) имеют вид

	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$
$S =$	0	0	1	1	1	1	1,
$\alpha =$	0	1	0	0	0	0	0.

Возникает вопрос о наименьшем количестве тактов, за которые можно вычислить значение формулы, ответ на который дан в следующем разделе.

### 2.5.5. Метод декомпозиции для вычисления произвольных булевых формул

Пусть задана произвольная формула  $f$ , представленная в ДНФ и зависящая от  $n$  аргументов. Подвергнем ее операции декомпозиции следующим образом:

$$f = \varphi_1 \vee \varphi_2 \vee \dots \vee \varphi_n, \quad (2.13)$$

где  $\varphi_i$  ( $i = 1, 2, \dots, n$ ) – неповторные упорядоченные формулы с пропусками или без пропусков аргументов.

Чтобы формулу  $f$  вычислить за наименьшее число тактов и с наименьшими затратами памяти, необходимо найти минимальный список формул  $\varphi_i$ , дизъюнкция которых дает формулу  $f$ . В процессе решения данной задачи разработан метод нахождения минимального списка формул  $\varphi_i$ , которые в дальнейшем будем называть  $\varphi$ -импликантами.

Суть метода состоит в следующем. Находим конъюнкцию, содержащую наименьший номер аргумента. Если их несколько, то выбираем из них первый произвольным образом. Если выбранная конъюнкция состоит более чем из одного аргумента, то она оканчивается аргументом с номером  $\delta_1$ . Затем находим конъюнкцию, содержащую первым аргумент, индекс которого является ближайшим к  $\delta_1$ . Это будет вторая конъюнкция, которая войдет в  $\varphi$ -импликанту. Вторая конъюнкция оканчивается аргументом с индексом  $\delta_2$ . Аналогично находим третью, четвертую и т.д. конъюнкции, пока не получим конъюнкцию, оканчивающуюся аргументом с номером  $\delta_i$ , таким, что других конъюнкций, первый аргумент которых превышает  $\delta_i$ , в заданной формуле  $f$  нет.



Закодируем двоичными числами все найденные  $\varphi$ -импликанты. Для этого запишем в ряд конъюнкции, образующие формулу (2.13). Каждой из них поставим в соответствие двоичный разряд и единицами будем отмечать конъюнкции, входящие в ту или иную  $\varphi$ -импликанту. Перечислив все  $\varphi$ -импликанты, получим таблицу двоичных чисел, где каждому числу будет соответствовать определенная  $\varphi$ -импликанта. Обозначим  $\varphi$ -импликанты символами  $\varphi_1, \varphi_2, \dots, \varphi_n$ . Затем составим булево уравнение, представляющее собой конъюнкцию скобочных выражений, где в каждой скобке записана дизъюнкция  $\varphi$ -импликант. Число скобочных выражений в уравнении равно числу конъюнкций заданной формулы, т.е. каждой конъюнкции заданной формулы соответствует скобочное выражение в уравнении. Чтобы найти  $\varphi_1, \varphi_2, \dots, \varphi_n$ , входящие в скобочные выражения, достаточно обратить внимание на единицы в соответствующей колонке полученной таблицы.

Представим булево уравнение в ДНФ. После минимизации получим дизъюнкцию конъюнкции символов  $\varphi_1, \varphi_2, \dots, \varphi_n$ . Каждая конъюнкция дает одно решение задачи декомпозиции.

Для иллюстрации метода рассмотрим пример. Дана формула

$$f = A_1A_2A_3 \vee A_1A_4A_6 \vee A_4A_7A_8A_9 \vee A_5A_7A_8 \vee \\ \vee A_9A_{10}A_{12}A_{14} \vee A_9A_{13} \vee A_{10}A_{11}A_{12} \vee A_{13}A_{14},$$

которая не является неповторной, следовательно, ее необходимо подвергнуть операции декомпозиции. Наиболее простой вариант декомпозиции состоит в том, что вычисление ведется отдельно по каждой конъюнкции. При этом потребуется 8 тактов. Выясним, сколько тактов потребуется при минимальном варианте. Составляем таблицу (прил. 1, табл. П.1.3), где полный список  $\varphi$ -импликант, выраженных через исходные аргументы заданной формулы, имеет вид

$$\varphi_1 = A_1A_2A_3 \vee A_4A_7A_8A_9 \vee A_{10}A_{11}A_{12} \vee A_{13}A_{14};$$

$$\varphi_2 = A_1A_2A_3 \vee A_5A_7A_8 \vee A_9A_{10}A_{12}A_{14};$$

$$\varphi_3 = A_1A_2A_3 \vee A_5A_7A_8 \vee A_9A_{13};$$

$$\varphi_4 = A_1A_2A_3 \vee A_5A_7A_8 \vee A_{10}A_{11}A_{12} \vee A_{13}A_{14};$$

$$\varphi_5 = A_1A_4A_6 \vee A_9A_{10}A_{12}A_{14};$$

$$\varphi_6 = A_1 A_4 A_6 \vee A_9 A_{13};$$

$$\varphi_7 = A_1 A_4 A_6 \vee A_{10} A_{11} A_{12} \vee A_{13} A_{14}.$$

Составляем булево уравнение. Прежде всего рассмотрим колонку таблицы, где записана конъюнкция  $A_4 A_7 A_8 A_9$ , так как в ней только одна единица. Это значит, что конъюнкция  $A_4 A_7 A_8 A_9$  войдет в какую-либо формулу искомого списка только в единственном случае: когда в список будет включена  $\varphi$ -импликанта  $\varphi_1$ . Очевидно, что эта  $\varphi$ -импликанта войдет во все варианты декомпозиции. Далее, поскольку импликанта  $\varphi_1$  введена в список, в него будут введены и конъюнкции  $A_1 A_2 A_3$ ,  $A_{10} A_{11} A_{12}$ ,  $A_{13} A_{14}$ . Вычеркиваем из таблицы верхнюю строку и колонки, обозначенные единицами в этой строке. Построим новую таблицу (см. прил. 1, табл. П.1.4), а по ней – булево уравнение в виде конъюнкции дизъюнкций:

$$(\varphi_5 \vee \varphi_6 \vee \varphi_7) (\varphi_2 \vee \varphi_3 \vee \varphi_4) (\varphi_2 \vee \varphi_5) (\varphi_3 \vee \varphi_6) = 1.$$

Раскроем скобки:

$$\begin{aligned} (\varphi_5 \vee \varphi_2 \varphi_6 \vee \varphi_2 \varphi_7) (\varphi_3 \vee \varphi_2 \varphi_6 \vee \varphi_4 \varphi_6) = & \varphi_3 \varphi_5 \vee \varphi_2 \varphi_3 \varphi_6 \vee \varphi_2 \varphi_3 \varphi_7 \vee \\ & \vee \varphi_2 \varphi_5 \varphi_6 \vee \varphi_2 \varphi_6 \vee \varphi_2 \varphi_6 \varphi_7 \vee \varphi_4 \varphi_5 \varphi_6 \vee \varphi_2 \varphi_4 \varphi_6 \vee \varphi_2 \varphi_4 \varphi_7. \end{aligned}$$

Выполним все операции поглощения, тогда получим минимальную ДНФ:

$$\varphi_3 \varphi_5 \vee \varphi_2 \varphi_6 \vee \varphi_2 \varphi_3 \varphi_7 \vee \varphi_4 \varphi_5 \varphi_6 = 1.$$

Отсюда следует, что существует четыре варианта искомого списка формул, из которых два варианта соответствуют случаю, когда список содержит три формулы. Эти два минимальных варианта имеют вид

$$\begin{cases} \varphi_1 = A_1 A_2 A_3 \vee A_4 A_7 A_8 A_9 \vee A_{10} A_{11} A_{12} \vee A_{13} A_{14}, \\ \varphi_3 = A_1 A_2 A_3 \vee A_5 A_7 A_8 \vee A_9 A_{13}, \\ \varphi_5 = A_1 A_4 A_6 \vee A_9 A_{10} A_{12} A_{14}; \end{cases} \quad (2.14)$$

$$\begin{cases} \varphi_1 = A_1 A_2 A_3 \vee A_4 A_7 A_8 A_9 \vee A_{10} A_{11} A_{12} \vee A_{13} A_{14}, \\ \varphi_2 = A_1 A_2 A_3 \vee A_5 A_7 A_8 \vee A_9 A_{10} A_{12} A_{14}, \\ \varphi_6 = A_1 A_4 A_6 \vee A_9 A_{13}. \end{cases} \quad (2.15)$$

В списке (2.14) конъюнкция  $A_1A_2A_3$  встречается дважды. Одну из них можно удалить, например, из формулы  $\varphi_3$ . Аналогично удаляем эту конъюнкцию и из формулы  $\varphi_2$  списка (2.15).

В связи с тем, что в формуле  $\varphi_3$  системы (2.14) конъюнкцию  $A_1A_2A_3$  можно было и не вычеркивать, делаем вывод, что в общем случае представление минимального списка  $\varphi$ -импликант  $\alpha$ - и  $S$ -кодами неоднозначно.

### 2.5.6. Однотактное вычисление неупорядоченных булевых формул

Как отмечалось в разделе 2.5.4, существует третий путь вычисления неупорядоченных булевых формул с помощью линейных изотропных сред на многофункциональных логических модулях, обеспечивающих вычисление как упорядоченных, так и неупорядоченных булевых формул. Применение МЛМ подобного типа сопряжено с трудностями их разработки. Проиллюстрируем МЛМ [8], ориентированный на вычисление вышеназванных булевых формул, представленных классами 1–6 и подклассом  $J$ , определяемым следующим образом.

Пусть дана формула, явно или неявно зависящая от аргументов  $x_1, x_2, \dots, x_n$ . Запишем эти аргументы в порядке возрастания их индексов слева направо. Аргумент, имеющий наименьший индекс, будем называть минимальным, а наибольший – максимальным. Диапазоном формулы назовем замкнутый интервал, границы которого образуют индексы минимального и максимального аргументов. Условимся, что интервалы двух различных формул пересекаются, если минимальный аргумент одной из формул входит в интервал другой.

Если формула  $f$  представлена в виде [23]:

$$f = l * \omega * q,$$

то она входит в подкласс  $J$ , где  $l$  и  $\omega$  – упорядоченные формулы с пересекающимися диапазонами;  $q$  – упорядоченная формула, минимальный аргумент которой не входит в диапазон функций  $l$  и  $\omega$ .

В общем случае формула  $q$  может быть тождественно равна нулю; \* – знак конъюнкции или дизъюнкции. Формулы  $l$  и  $\omega$  могут быть любого порядка.

Логическая схема МЛМ (рис. 2.12) описывается следующей системой булевых формул [21]:

$$\begin{cases} f_1 = y_1(\bar{z}_1z_2 \vee \bar{z}_1z_3\bar{z}_4 \vee x\bar{z}_3), \\ f_2 = (\bar{z}_1x \vee z_3x \vee z_2 \vee \bar{z}_2y_1 \vee \bar{z}_1\bar{z}_3 \vee \bar{z}_1z_4)y_2 \vee \\ \vee xz_4 \vee y_1z_1z_2, \end{cases} \quad (2.16)$$

где  $x, y_1, y_2$  – информационные входы;  $z_1, z_2, z_3, z_4$  – настроечные входы;  $f_1, f_2$  – выходы МЛМ.

Многофункциональный логический модуль, который в дальнейшем будем называть  $H$ -ячейкой, реализует следующие системы булевых формул:

1) при  $z_4 = 0, z_3 = 0, z_2 = 0, z_1 = 0$   
(рис. 2.13, а)

$$\begin{cases} f_1 = y_1x, \\ f_2 = y_2; \end{cases}$$

2) при  $z_4 = 0, z_3 = 0, z_2 = 0, z_1 = 1$   
(рис. 2.13, б)

$$\begin{cases} f_1 = x, \\ f_2 = y_1y_2; \end{cases}$$

3) при  $z_4 = 0, z_3 = 0, z_2 = 1, z_1 = 0$   
(рис. 2.13, в)

$$\begin{cases} f_1 = y_1 \vee x, \\ f_2 = y_2; \end{cases}$$

4) при  $z_4 = 0, z_3 = 0, z_2 = 1, z_1 = 1$   
(рис. 2.13, г)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 \vee y_2; \end{cases}$$

5) при  $z_4 = 0, z_3 = 1, z_2 = 0, z_1 = 0$   
(рис. 2.13, д)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2x; \end{cases}$$

6) при  $z_4 = 0, z_3 = 1, z_2 = 0, z_1 = 1$   
(рис. 2.13, е)

$$\begin{cases} f_1 = 0, \\ f_2 = xy_2; \end{cases}$$

7) при  $z_4 = 0, z_3 = 1, z_2 = 1, z_1 = 0$   
(рис. 2.13, ж)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2; \end{cases}$$

8) при  $z_4 = 0, z_3 = 1, z_2 = 1, z_1 = 1$   
(рис. 2.13, з)

$$\begin{cases} f_1 = 0, \\ f_2 = y_1 \vee y_2; \end{cases}$$

9) при  $z_4 = 1, z_3 = 0, z_2 = 0, z_1 = 0$   
(рис. 2.13, и)

$$\begin{cases} f_1 = y_1x, \\ f_2 = y_2 \vee x; \end{cases}$$

10) при  $z_4 = 1, z_3 = 0, z_2 = 0, z_1 = 1$  (рис. 2.13, к)

$$\begin{cases} f_1 = x, \\ f_2 = y_1y_2 \vee x; \end{cases}$$

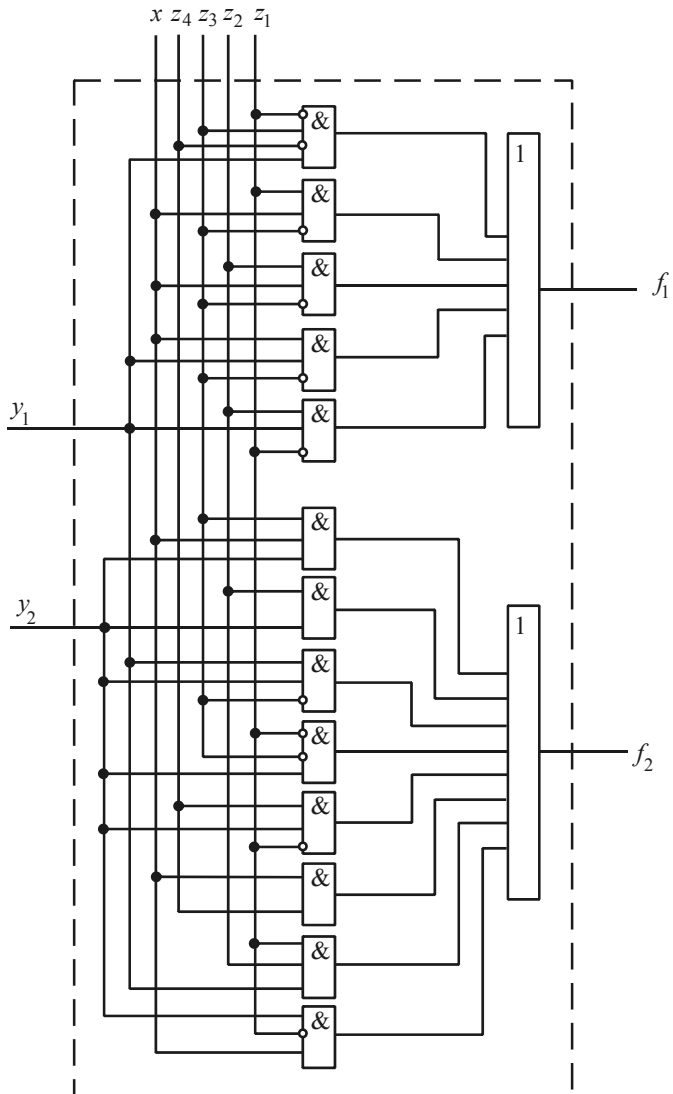


Рис. 2.12. Ячейка H-структуры

11) при  $z_4 = 1, z_3 = 0, z_2 = 1, z_1 = 0$  (рис. 2.13, л)

$$\begin{cases} f_1 = y_1 \vee x, \\ f_2 = y_2 \vee x; \end{cases}$$

12) при  $z_4 = 1, z_3 = 0, z_2 = 1, z_1 = 1$  (рис. 2.13, м)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 \vee y_2 \vee x; \end{cases}$$

13) при  $z_4 = 1, z_3 = 1, z_2 = 0, z_1 = 0$  (рис. 2.13, н)

$$\begin{cases} f_1 = 0, \\ f_2 = y_2 \vee x; \end{cases}$$

14) при  $z_4 = 1, z_3 = 1, z_2 = 0, z_1 = 1$  (рис. 2.13, п)

$$\begin{cases} f_1 = 0, \\ f_2 = x; \end{cases}$$

15) при  $z_4 = 1, z_3 = 1, z_2 = 1, z_1 = 0$  (рис. 2.13, р)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2 \vee x; \end{cases}$$

16) при  $z_4 = 1, z_3 = 1, z_2 = 1, z_1 = 1$  (рис. 2.13, с)

$$\begin{cases} f_1 = 0, \\ f_2 = y_1 \vee y_2 \vee x. \end{cases}$$

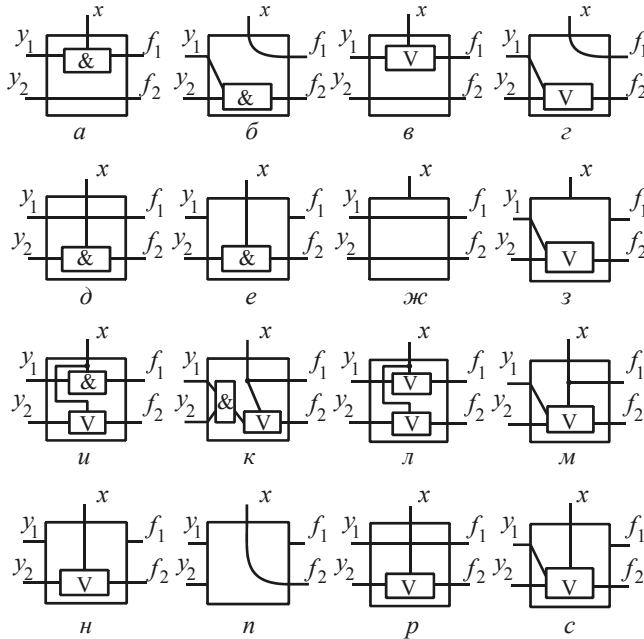


Рис. 2.13. Структурные схемы, полученные путем настройки

Таким образом, изотропные среды, построенные на  $H$ -ячейках, обеспечивают реализацию как класса неповторных упорядоченных произвольных нормальных формул из  $h$  букв (в том числе любых скобочных), так и формул из класса неупорядоченных и повторных формул, определяемых подклассом  $J$ .

### 2.5.7. Вычисление систем булевых формул из классов неповторных упорядоченных и неупорядоченных формул

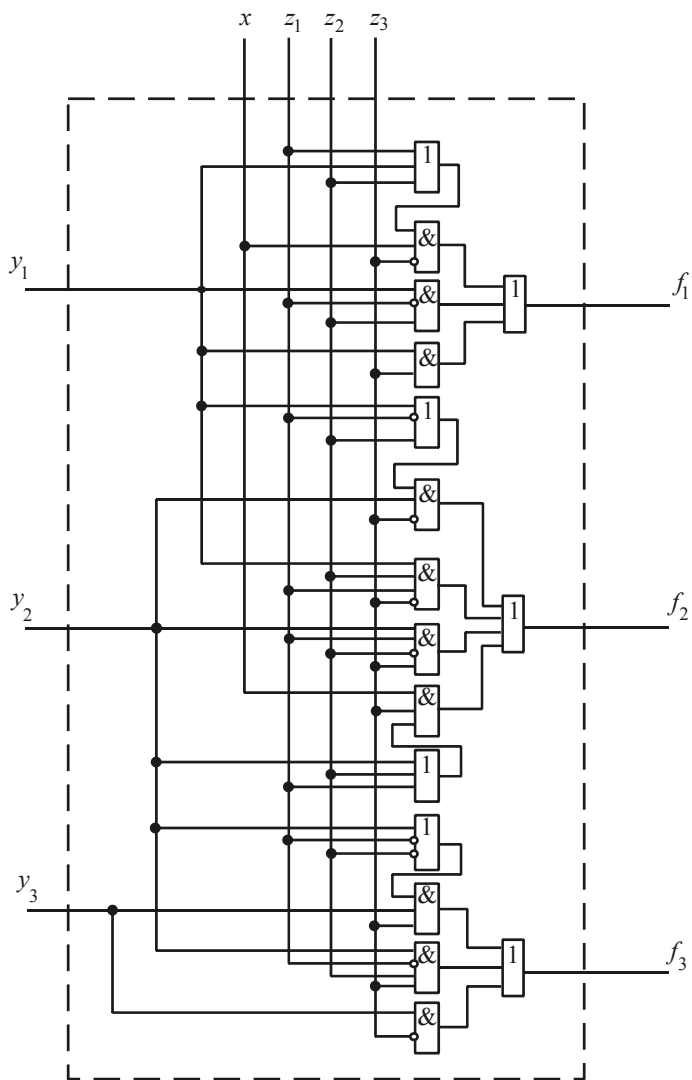
В разделе 2.5.5 показано, что существующие МЛМ охватывают классы 1–6 и подкласс  $J$  классификации (см. рис. 2.2). В данном и следующем разделах покажем реализацию разработанных МЛМ, ориентированных на вычисления систем булевых формул из классов неповторных упорядоченных и неупорядоченных формул, а также МЛМ, реализующих класс повторных упорядоченных произвольных нормальных булевых формул из  $h$  букв и систем булевых формул как с пропусками аргументов, так и без них.

Ячейка (рис. 2.14) реализует системы булевых формул из классов неповторных упорядоченных и неупорядоченных формул.

Напомним, что формулу будем считать неповторной, если каждый аргумент входит в нее не более одного раза. Неповторной будем считать формулу и в том случае, если существуют тождественные преобразования, в результате которых формула, содержащая повторные аргументы, приводится к виду, не содержащему повторных аргументов. Во всех остальных случаях формула является повторной.

Структура предлагаемой ячейки описывается следующей системой формул [12]:

$$\begin{cases} f_1 = y_1(\bar{Z}_1 Z_2 \vee Z_3) \vee (y_1 \vee Z_1 \vee Z_2) x \bar{Z}_3, \\ f_2 = y_1 Z_1 Z_2 \bar{Z}_3 \vee y_2 Z_1 \bar{Z}_2 Z_3 \vee (y_1 \vee \bar{Z}_1 \vee Z_2) y_2 \bar{Z}_3 \vee \\ \vee (y_2 \vee Z_1 \vee Z_2) x Z_3, \\ f_3 = y_3 \bar{Z}_3 \vee (y_2 \vee \bar{Z}_1 \vee \bar{Z}_2) y_3 Z_3 \vee y_2 \bar{Z}_1 Z_2 Z_3. \end{cases} \quad (2.17)$$

Рис. 2.14. Ячейка  $L$ -структуры



Ячейка путем настройки реализует следующие системы формул:

1) при  $Z_1 = 0, Z_2 = 0, Z_3 = 0$

(рис. 2.15, а)

$$\begin{cases} f_1 = xy_1, \\ f_2 = y_2, \\ f_3 = y_3; \end{cases}$$

5) при  $Z_1 = 1, Z_2 = 0, Z_3 = 0$

(рис. 2.15, б)

$$\begin{cases} f_1 = x, \\ f_2 = y_1y_2, \\ f_3 = y_3; \end{cases}$$

2) при  $Z_1 = 0, Z_2 = 0, Z_3 = 1$

(рис. 2.15, в)

$$\begin{cases} f_1 = y_1, \\ f_2 = xy_2, \\ f_3 = y_3; \end{cases}$$

6) при  $Z_1 = 1, Z_2 = 0, Z_3 = 1$

(рис. 2.15, г)

$$\begin{cases} f_1 = y_1, \\ f_2 = x \vee y_2, \\ f_3 = y_3; \end{cases}$$

3) при  $Z_1 = 0, Z_2 = 1, Z_3 = 0$

(рис. 2.15, д)

$$\begin{cases} f_1 = x \vee y_1, \\ f_2 = y_2, \\ f_3 = y_3; \end{cases}$$

7) при  $Z_1 = 1, Z_2 = 1, Z_3 = 0$

(рис. 2.15, е)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 \vee y_2, \\ f_3 = y_3; \end{cases}$$

4) при  $Z_1 = 0, Z_2 = 1, Z_3 = 1$

(рис. 2.15, з)

$$\begin{cases} f_1 = y_1, \\ f_2 = x, \\ f_3 = y_2 \vee y_3; \end{cases}$$

8) при  $Z_1 = 1, Z_2 = 1, Z_3 = 1$

(рис. 2.15, ж)

$$\begin{cases} f_1 = y_1, \\ f_2 = x, \\ f_3 = y_2y_3. \end{cases}$$

Проиллюстрируем работу однородных сред, построенных из предлагаемых ячеек на следующих примерах.

**Пример 1.** Для реализации системы булевых формул из классов бесповторных упорядоченных и неупорядоченных формул вида

$$\begin{cases} f_1 = x_1x_2x_3 \vee x_4x_5 \vee x_6, \\ f_2 = (x_7 \vee x_9)(x_8 \vee x_{10}) \end{cases}$$

строится однородная среда с настроечными кодами (рис. 2.16).

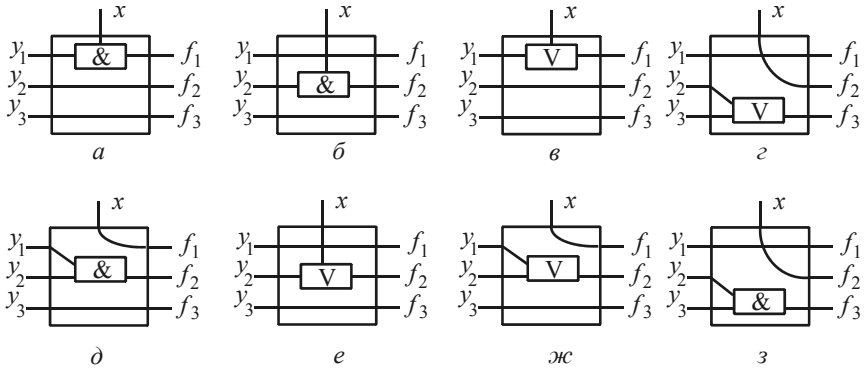


Рис. 2.15. Структурные схемы, полученные путем настройки

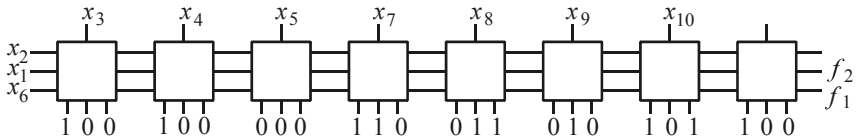


Рис. 2.16. Иллюстрация к примеру 1

Пример 2. Аналогично можно найти настроечные коды для системы формул вида

$$\begin{cases} f_1 = (x_1 x_3 \vee x_4) x_2, \\ f_2 = x_5 \vee x_6 x_7. \end{cases}$$

Ее однородная среда и настроечные коды представлены на рис. 2.17.

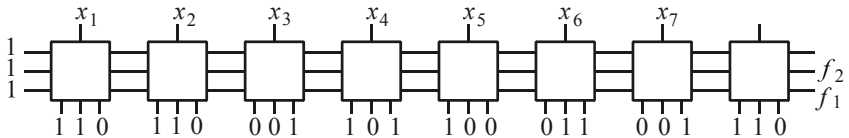


Рис. 2.17. Иллюстрация к примеру 2

Пример 3. На рис. 2.18 показаны настроечные коды каждой ячейки однородной среды, реализующей систему формул вида

$$\begin{cases} f_1 = (x_1 x_2 \vee x_3 x_4) x_5, \\ f_2 = (x_6 \vee x_7)(x_8 \vee x_9). \end{cases}$$

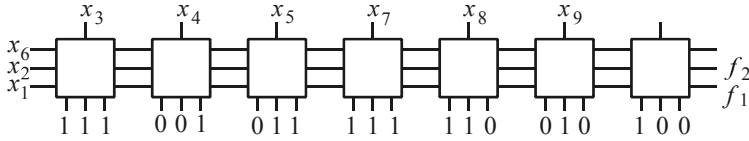


Рис. 2.18. Иллюстрация к примеру 3

Таким образом, изотропная среда, построенная на  $L$ -ячейках, обеспечивает реализацию систем булевых формул из классов неповторных упорядоченных и неупорядоченных булевых формул.

**2.5.8. Вычисление повторных упорядоченных произвольных нормальных булевых формул из  $h$  букв и систем булевых формул как с пропусками аргументов, так и без них**

Ячейка (рис. 2.19) реализует класс повторных упорядоченных произвольных нормальных булевых формул из  $h$  букв, а также системы булевых формул как с пропусками аргументов, так и без них.

Структура предлагаемой ячейки описывается следующей системой формул [13]:

$$\begin{cases} f_1 = \bar{Z}_3\bar{Z}_4[x(Z_2 \vee y_1 \vee Z_1) \vee \bar{Z}_1Z_2y_1] \vee Z_3\bar{Z}_4y_1 \vee \bar{Z}_3Z_4x \vee \\ \vee Z_3Z_4[x(Z_1 \vee \bar{Z}_2 \vee y_1) \vee \bar{Z}_1Z_2y_1], \\ f_2 = \bar{Z}_3\bar{Z}_4[y_2(Z_2 \vee \bar{Z}_1 \vee y_1) \vee Z_1Z_2y_1] \vee Z_3\bar{Z}_4y_2(x \vee Z_2) \vee \bar{Z}_3Z_4[y_2(Z_1 \vee \\ \vee \bar{Z}_2y_1 \vee y_1x) \vee Z_1(y_1 \vee Z_2x)] \vee Z_3Z_4[\bar{Z}_1Z_2(y_2 \vee x) \vee y_1(\bar{Z}_2 \vee x)], \\ f_3 = \bar{Z}_3\bar{Z}_4[y_3(Z_1 \vee \bar{Z}_2 \vee x) \vee \bar{Z}_1\bar{Z}_2x] \vee Z_3\bar{Z}_4[y_3(Z_2 \vee \bar{Z}_1 \vee x) \vee Z_1Z_2x] \vee \\ \vee \bar{Z}_3Z_4[y_3(\bar{Z}_1 \vee Z_2 \vee x) \vee \bar{Z}_1\bar{Z}_2x] \vee Z_3Z_4[y_3(Z_1 \vee Z_2 \vee y_2) \vee Z_1y_2]. \end{cases}$$

Ячейка путем настройки реализует следующие системы формул:

1) при  $Z_4 = 0, Z_3 = 0, Z_2 = 0,$   
 $Z_1 = 1$  (рис. 2.20, а)

$$\begin{cases} f_1 = x, \\ f_2 = y_1y_2, \\ f_3 = y_3; \end{cases}$$

2) при  $Z_4 = 0, Z_3 = 0, Z_2 = 1,$   
 $Z_1 = 0$  (рис. 2.20, б)

$$\begin{cases} f_1 = x \vee y_1, \\ f_2 = y_2, \\ f_3 = xy_3; \end{cases}$$

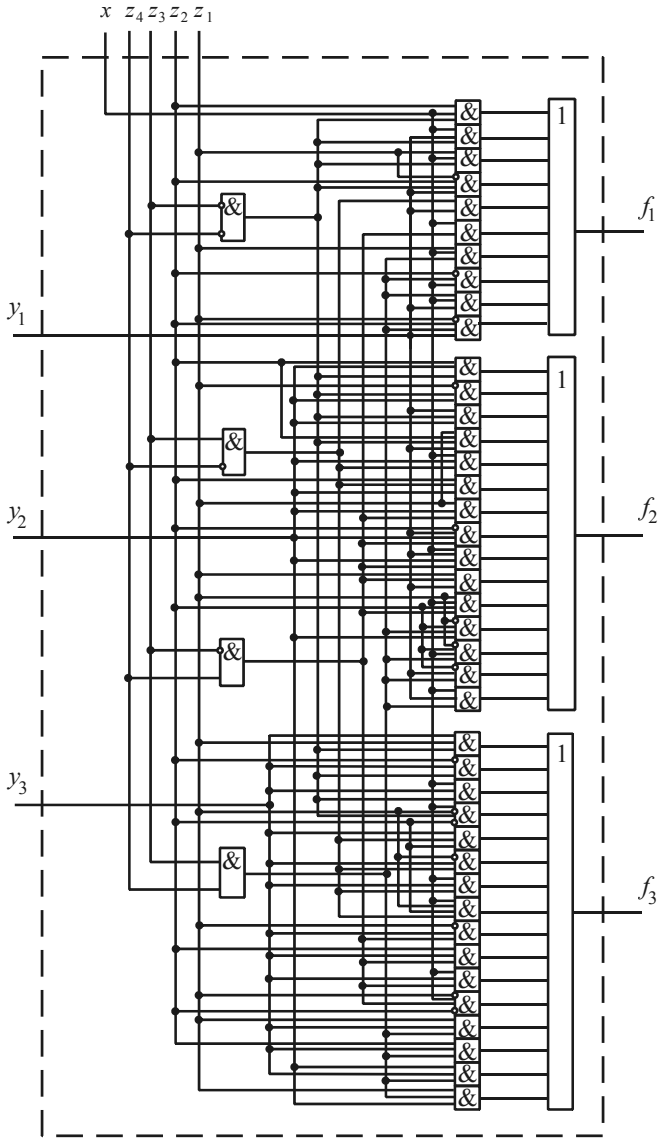


Рис. 2.19. Ячейка V-структуры

3) при  $Z_4 = 0, Z_3 = 0, Z_2 = 1, Z_1 = 1$  (рис. 2.20, в)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 \vee y_2, \\ f_3 = y_3; \end{cases}$$

4) при  $Z_4 = 0, Z_3 = 1, Z_2 = 0, Z_1 = 0$  (рис. 2.20, г)

$$\begin{cases} f_1 = y_1, \\ f_2 = xy_2, \\ f_3 = y_3; \end{cases}$$

5) при  $Z_4 = 0, Z_3 = 1, Z_2 = 0, Z_1 = 1$  (рис. 2.20, д)

$$\begin{cases} f_1 = y_1, \\ f_2 = xy_2, \\ f_3 = xy_3; \end{cases}$$

6) при  $Z_4 = 0, Z_3 = 1, Z_2 = 1, Z_1 = 0$  (рис. 2.20, е)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2, \\ f_3 = y_3; \end{cases}$$

7) при  $Z_4 = 0, Z_3 = 1, Z_2 = 1, Z_1 = 1$  (рис. 2.20, ж)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2, \\ f_3 = x \vee y_3; \end{cases}$$

8) при  $Z_4 = 1, Z_3 = 0, Z_2 = 0, Z_1 = 0$  (рис. 2.20, з)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 y_2, \\ f_3 = x \vee y_3; \end{cases}$$

9) при  $Z_4 = 1, Z_3 = 0, Z_2 = 0, Z_1 = 1$  (рис. 2.20, и)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 \vee y_2, \\ f_3 = xy_3; \end{cases}$$

10) при  $Z_4 = 1, Z_3 = 0, Z_2 = 1, Z_1 = 0$  (рис. 2.20, к)

$$\begin{cases} f_1 = x, \\ f_2 = xy_1 y_2, \\ f_3 = y_3; \end{cases}$$

11) при  $Z_4 = 1, Z_3 = 0, Z_2 = 1, Z_1 = 1$  (рис. 2.20, л)

$$\begin{cases} f_1 = x, \\ f_2 = x \vee y_1 \vee y_2, \\ f_3 = y_3; \end{cases}$$

12) при  $Z_4 = 1, Z_3 = 1, Z_2 = 0, Z_1 = 0$  (рис. 2.20, м)

$$\begin{cases} f_1 = x, \\ f_2 = y_1, \\ f_3 = y_2 y_3; \end{cases}$$

13) при  $Z_4 = 1, Z_3 = 1, Z_2 = 0, Z_1 = 1$  (рис. 2.20, н)

$$\begin{cases} f_1 = x, \\ f_2 = y_1, \\ f_3 = y_2 \vee y_3; \end{cases}$$

14) при  $Z_4 = 1, Z_3 = 1, Z_2 = 1, Z_1 = 0$  (рис. 2.20, о)

$$\begin{cases} f_1 = y_1, \\ f_2 = x \vee y_2, \\ f_3 = y_3; \end{cases}$$

15) при  $Z_4 = 1, Z_3 = 1, Z_2 = 1,$   
 $Z_1 = 1$  (рис. 2.20, *p*)

$$\begin{cases} f_1 = x, \\ f_2 = xy_1, \\ f_3 = y_2 \vee y_3; \end{cases}$$

16) при  $Z_4 = 0, Z_3 = 0, Z_2 = 0,$   
 $Z_1 = 0$  (рис. 2.20, *c*)

$$\begin{cases} f_1 = xy_1, \\ f_2 = y_2, \\ f_3 = x \vee y_3. \end{cases}$$

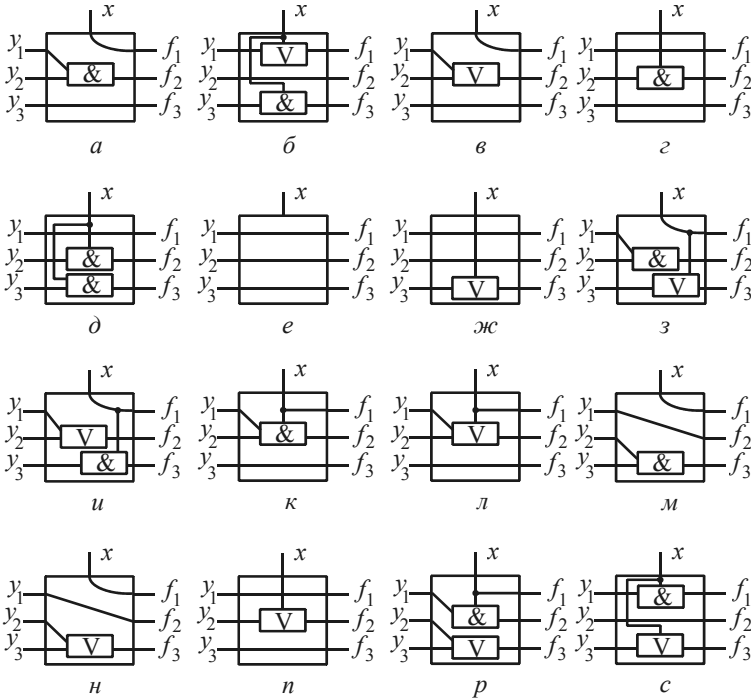


Рис. 2.20. Структурные схемы, полученные путем настройки

Работа однородных сред, построенных из предлагаемых ячеек, показана на следующих примерах.

Пример 1. Реализация повторной упорядоченной формулы без пропусков аргументов, представленной в КНФ,

$$f_1 = (x_1 \vee x_2)(x_2 \vee x_3)(x_3 \vee x_4)$$

приведена на рис. 2.21.

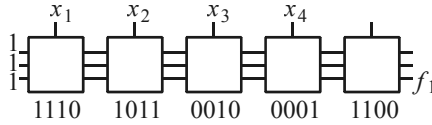


Рис. 2.21. Иллюстрация к примеру 1

Пример 2. Повторная упорядоченная формула без пропусков аргументов, представленная в ДНФ,

$$f_2 = x_1x_2 \vee x_2x_3 \vee x_3x_4$$

реализуется линейной однородной структурой (рис. 2.22).

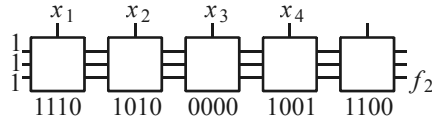


Рис. 2.22. Иллюстрация к примеру 2

Пример 3. На рис. 2.23 показаны настроечные коды каждой ячейки однородной структуры, реализующей повторную упорядоченную формулу без пропусков аргументов, представленную в форме высших (4-го) порядков:

$$f_3 = (x_1 \vee x_2)x_3 \vee x_3(x_4 \vee x_5 \vee x_6) .$$

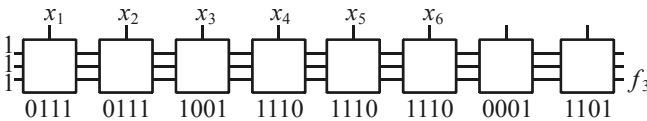


Рис. 2.23. Иллюстрация к примеру 3

Пример 4. Реализация повторной упорядоченной формулы с пропусками аргументов  $x_2$  и  $x_6$ , представленной в ДНФ,

$$f_4 = x_1x_3 \vee x_1x_4 \vee x_3x_4 \vee x_5 \vee x_7$$

показана на рис. 2.24.

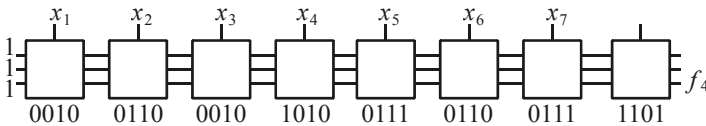


Рис. 2.24. Иллюстрация к примеру 4

Пример 5. Неупорядоченная повторная формула, представленная в ДНФ,

$$f_5 = x_1x_3x_6 \vee x_2x_4x_6 \vee x_5x_7$$

реализуется структурой, приведенной на рис. 2.25.

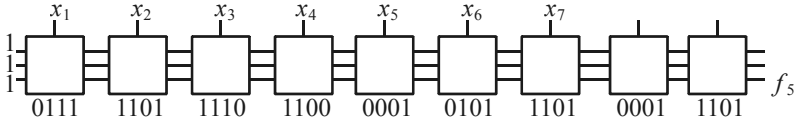


Рис. 2.25. Иллюстрация к примеру 5

Пример 6. На рис. 2.26 представлена реализация системы формул высокого порядка с пропусками аргументов вида

$$\begin{cases} f_6 = \{[(x_1x_6 \vee x_8)x_{10} \vee x_{11}]x_{12} \vee x_{13}\} \{[(x_2x_3x_4 \vee x_5)x_7x_9 \vee x_{14}]\} \vee \\ \vee [(x_{15}x_{16} \vee x_{17})x_{18} \vee x_{20}]x_{21}, \\ f_7 = \{[(x_1x_2 \vee x_3 \vee x_4)x_5x_6 \vee x_7 \vee x_9]x_{14} \vee x_{16}\} x_{17} \vee x_{18} \} x_{20} \vee x_{21}. \end{cases}$$

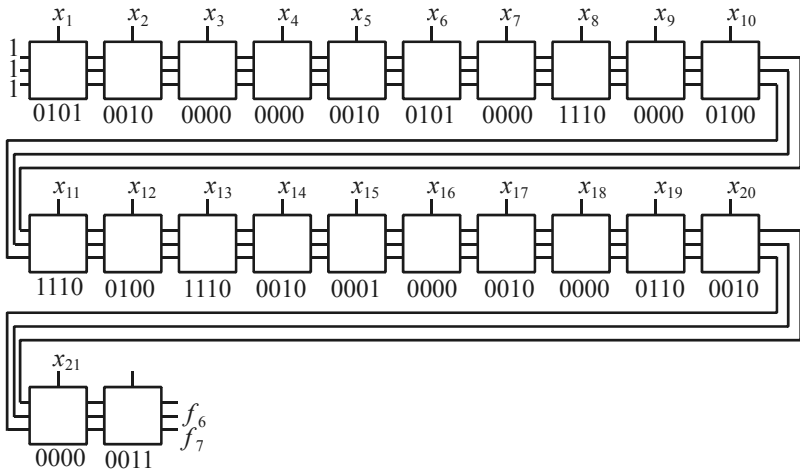


Рис. 2.26. Иллюстрация к примеру 6



Пример 7. Для реализации неповторной упорядоченной формулы вида

$$f_8 = [(x_1 \vee x_2)(x_3 \vee x_4) \vee (x_5 \vee x_6)(x_7 \vee x_8)][(x_9 \vee x_{10})(x_{11} \vee x_{12}) \vee (x_{13} \vee x_{14})(x_{15} \vee x_{16})]$$

можно построить древовидную структуру с выделенными каскадами (рис. 2.27). Схема каскада и настроечные коды для его ячеек показаны на рис. 2.28.

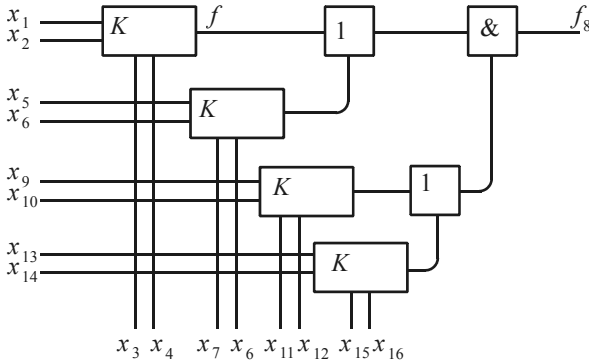


Рис. 2.27. Иллюстрация к примеру 7

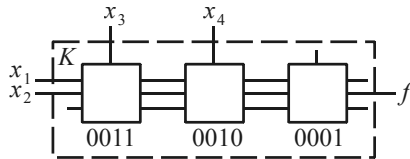


Рис. 2.28. Схема каскада из примера 7

Пример 8. Повторная неупорядоченная формула без пропусков аргументов, представленная в ДНФ,

$$f_9 = x_1x_3 \vee x_2x_3 \vee x_2x_4$$

реализуется однородной структурой (рис. 2.29).

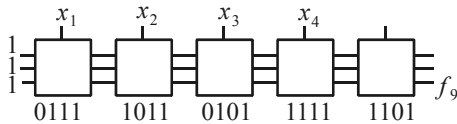


Рис. 2.29. Иллюстрация к примеру 8

Таким образом, изотропная среда, построенная на  $V$ -ячейках, реализует класс повторных упорядоченных произвольных нормальных булевых формул из  $h$  букв, а также системы булевых формул как с пропусками аргументов, так и без них.

## 2.6. ВЫВОДЫ

Основной результат исследований, изложенный в данной главе, заключается в решении проблемы вычисления систем булевых формул из классов неповторных упорядоченных и неупорядоченных, а также класса повторных упорядоченных произвольных нормальных булевых формул из  $h$  букв и систем булевых формул как с пропусками аргументов, так и без них.

1. Предложена классификация булевых формул. Установлено, что для каждого класса достаточно реализовать лишь одну схему, структура которой описывается формулой представителя класса.

2. Предложена булева модель логики перестраиваемых структур, используя которую можно создавать практические логические управляющие устройства с перестраиваемой структурой, реализующие алгоритм управления из заранее заданного класса.

3. Проведен анализ существующих многофункциональных логических модулей и на их основе синтезированы новые структуры.

3. Предложены многоканальные линейные однородные структуры из  $h-1$  ячеек для реализации булевых формул в рассматриваемом базисе из  $h$  букв.

4. Предложены плоскостные однородные структуры для реализации указанных классов булевых формул.

5. Установлено, что для настройки структуры логического управляющего устройства на конкретную формулу достаточно знать только код этой формулы.

## ЛИТЕРАТУРА

1. *Артюхов В.Л., Копейкин Г.А., Шальто А.А.* Настраиваемые модули для управляющих логических устройств. – Л.: Энергоатомиздат, 1981. – 168 с.
2. *Гиндикин С.А.* Алгебра логики в задачах. – М.: Наука, 1972. – 288 с.
3. *Колдуэлл С.* Логический синтез релейных устройств / Пер. с англ. Г.К. Москатова, А.Д. Таланцева. – М.: ИЛ, 1962. – 740 с.

4. Лазарев В.Г., Пийль Е.И., Турута Е.Н. Построение программируемых управляющих устройств. – М.: Энергоатомиздат, 1984. – 192 с.

5. Миллер Р. Теория переключательных схем. – М.: Наука, 1970. Т. 1. – 416 с.

6. Шальто А.А. Логическое управление. Методы аппаратной и программной реализации алгоритмов. – СПб.: Наука, 2000. – 780 с.

7. Шевелев Ю.П. Дискретная математика. – Томск: ТУСУР, 1998, 1999. – Ч. 1, 2. – 114, 120 с.

8. Шевелев Ю.П., Шидловский В.С. А.с. 1476456 (СССР). Ячейка однородной среды // Б.И. – 1989. – № 16.

9. Шидловский С.В. Изотропная среда в системе автоматизированного управления и контроля // Труды VI Междунар. науч.-практич. конф. студентов, аспирантов и молодых ученых «Современные техника и технологии». – Томск: Изд-во ТПУ, 2000. – С. 161–162.

10. Шидловский С.В. Многофункциональный логический модуль для реализации операций удаления аргументов из булевых функций // Радиотехнические устройства, информационные технологии и системы управления: Тез. докл. региональной науч.-техн. конф. студентов и молодых специалистов. – Томск, 2001. – С.15–16.

11. Шидловский С.В. Перестраиваемые структуры на многофункциональных логических модулях // Информационные системы: Труды постоянно действующей науч.-техн. школы-семинара студентов, аспирантов и молодых специалистов «Информационные системы мониторинга окружающей среды». – Томск: ТУСУР, 2003. – Вып. 2. – С.105–117.

12. Шидловский С.В. Ячейка однородной среды. Патент РФ на изобретение № 2251140 // БИ. – 2005. – № 12. – 9 с.

13. Шидловский С.В. Ячейка однородной среды. Патент РФ на изобретение № 2251141 // БИ. – 2005. – № 12. – 13 с.

14. Шидловский С.В., Светлаков А.А. Исследование функциональных возможностей многофункционального логического модуля, реализующего операции удаления аргументов из булевых функций // Вестник Сибирского отделения АН ВШ. – 2002. – №1(8). – С. 72–78.

15. Шидловский С.В., Светлаков А.А. Однородная среда в системе контроля знаний // Тез. докл. науч.-методич. конф. «Современное образование: Качество и новые технологии». – Томск: ТУСУР, 2000. – С. 69–70.

16. Шидловский С.В., Светлаков А.А., Шидловский В.С. Многофункциональный логический модуль для проблемно-ориентированных процессоров // Измерение, контроль, информатизация: Материалы Второй Междунар. науч.-техн. конф. – Барнаул, 2001. – С. 289–292.

17. Шидловский С.В., Светлаков А.А., Шидловский В.С. Преобразование структур данных в специализированных изотропных процессорах // Методы и

средства измерения в системах контроля и управления: Материалы Всерос. науч.-техн. конф. – Пенза, 2001. – С. 213–217.

18. *Шидловский С.В., Фоминых Ю.И.* Исследование функциональных возможностей однородной структуры, обеспечивающей вычисление неповторных ДНФ булевых функций // Вестник Сибирского отделения АН ВШ. – 2001. – № 1(7). – С. 50–59.

19. *Шидловский С.В., Шевелев Ю.П.* Автомат с перестраиваемой структурой в системе контроля знаний // VI Междунар. науч.-методич. конф. вузов и факультетов телекоммуникаций. – М., 2000. – С. 97–98.

20. *Якубайтис Э.А.* Логические автоматы и микромодули. – Рига: Зинатне, 1975. – 259 с.

21. *Shidlovskiy S.V.* Multifunctional Automaton for Computation of Disordered Boolean Functions // The VIII International Scientific and Practical Conference of Students, Post-graduates and Young Scientist «Modern Techniques and Technology» (МТТ'2002). – Tomsk: Tomsk Polytechnic University, 2002. – P. 40–41.

22. *Shidlovskiy S.V.* Computation of nonrecurrent ordered Boolean functions of higher than the second order // SIBEDEM 2002. Proceeding. – Tomsk: The Tomsk IEEE Chapter & Student Branch. Russia, 2002. – P. 45–48.

23. *Shidlovskiy S.V., Svetlakov A.A., Shidlovskiy V.S.* Computation of disordered Boolean functions // SIBEDEM 2002. Proceeding. – Tomsk: The Tomsk IEEE Chapter & Student Branch, 2002. – P. 41–44.

## **РЕАЛИЗАЦИЯ БУЛЕВОЙ МОДЕЛИ ЛОГИКИ ПЕРЕСТРАИВАЕМЫХ СТРУКТУР С ПРИМЕНЕНИЕМ ИЗОТРОПНЫХ СРЕД**

---

---

### **3.1. ВВОДНЫЕ ПОНЯТИЯ**

Создание эффективно и надежно функционирующих систем логического управления для автоматизации различного рода производственных объектов является весьма важной и актуальной задачей.

При разработке современных систем логического управления особое внимание должно уделяться структурному и функционально-логическому уровням проектирования. Значение этих уровней обусловлено формированием на них важных решений, отвечающих за качество и надежность системы, а также за содержание и направленность последующих работ на остальных уровнях проектирования [11].

В настоящее время существует множество методов проектирования систем логического управления и их классификаций по ряду признаков, основными из которых являются автоматизированные и неавтоматизированные. Последний метод весьма трудоемкий и не отвечает современным информационным технологиям.

При реализации первого метода в качестве главных рабочих принципов выходят модульность, многофункциональность и перестраиваемость будущих систем логического управления.

Использование модульности позволяет создать функционально-блочные системы, результатом которых является высокая работоспособность, сокращение времени наладки и восстановление, т.е. увеличение надежности системы.

Многофункциональность и перестраиваемость решают разнородные задачи управления. К примеру, сокращенный объем аппаратуры позволяет управлять ресурсами с различных уровней иерархии систем, что оказывает влияние на пропускную способность, живучесть и надежность, а в целом повышает тактико-технические характеристики системы.

В практике проектирования и моделирования используется множество моделей и языков, описывающих поведение дискретных систем. Применение модели конечного автомата, являющейся научной основой логического управления, ранее было затруднено при описании и оптимизации функционирования систем логического управления из-за огромных комбинационных вычислений, которые превышали возможности вычислительных средств. В настоящее время современные персональные компьютеры позволяют приблизиться к решению этой задачи.

### 3.2. ИМИТАЦИОННАЯ СИСТЕМА CELL SYSTEM

Для синтеза последовательных и комбинационных логических схем разработано большое количество методов минимизации функций алгебры логики [2]. Многие из этих методов принципиально предназначены для «ручной» минимизации, некоторые допускают формализацию процесса решения этой задачи и использование компьютерных технологий. При большом числе входных и выходных переменных синтезируемой схемы «ручная» минимизация описывающих ее формул становится неэффективной и во многих случаях просто невозможной, поэтому автоматизация этого процесса является актуальной задачей.

Среди признанных методов исследования сложных систем особое место занимает имитационное моделирование. Повышенное внимание к нему определяется не только возможностью анализа систем при условиях большой размерности и неполной информации о структуре системы, но и доступностью методологии для широкого круга специалистов.

В нашем случае мы рассматриваем некоторые аспекты имитационного моделирования однородных сред. Из существующих ОС выделим класс сред, которые настраиваются на реализацию того или иного дискретного автомата путем подачи на управляющие входы соответствующих настроечных кодов. Главное достоинство структур этого класса состоит в том, что синтез дискретного автомата сводится по существу лишь к нахождению настроечных кодов на основе системы булевых формул, описывающих работу автомата.

ОС могут быть реализованы аппаратными и программными способами. Относительно возможностей аппаратной реализации в [3–5] отмечено, что ОС представляют собой структуры, идеально приспособленные к особенностям техники массового производства. Аппаратная реализация необходима в тех случаях, когда требуется обеспечить мак-

симальное быстродействие автомата. Если же требования к быстродействию не доминируют, то ОС целесообразнее реализовывать многотактным способом, например с использованием компьютерных технологий.

Интерес к программному моделированию ОС неслучаен. Дело в том, что программная реализация булевых функций в «чистом» виде сопряжена со значительными трудностями даже в таких случаях, когда число аргументов не велико (10–15). Эти трудности вызваны многими причинами.

*Во-первых*, для хранения сложных булевых функций необходим значительный объем памяти компьютера. Например, если функцию представить в виде изображающего числа [9], то для ее машинного представления потребуется  $N$  бит памяти:

$$N = 2^s,$$

где  $s$  – число логических аргументов.

Такая реализация функций (которая является по сути СДНФ) ограничивается 10–15 аргументами, что с практической точки зрения во многих случаях совершенно недостаточно. Например, логика работы информационно-поисковых автоматов [7] даже в минимальном варианте их использования моделируется булевыми формулами 20 аргументов.

*Во-вторых*, время вычисления формулы быстро растет с увеличением числа аргументов.

Иное дело алгоритмы, моделирующие работу однородной среды. Если ОС состоит из  $s$  МЛМ, где каждый МЛМ имеет  $m$  входов, то для машинного представления любой неповторной булевой формулы достаточно  $s \cdot m$  бит памяти. Если программная реализация МЛМ ОС осуществляется в течение времени  $t$ , то вычисление всякой неповторной булевой формулы будет выполнено за время  $s \cdot t$  [15].

Отсюда следует, что имитационный метод моделирования ОС можно рассматривать как решение проблемы программного определения значений булевых формул многих аргументов.

При синтезе автомат представляется в виде системы булевых формул, каждая из которых записывается в СДНФ:

$$F_k = \bigvee_{i \in C_k} \left( \bigwedge_{j=1}^n Y_j^{a_{ij}} \right)_i \quad (k = \overline{1, p}), \quad (3.1)$$

где  $p$  – число выходов схемы;  $C_k$  – множество наборов переменных, на которых  $k$ -я формула определена и принимает истинное значение;  $C_k \subseteq G$  ( $G$  – множество наборов переменных, на которых хотя бы одна из формул  $F_k$  определена и принимает истинное значение);  $n$  – число переменных формулы  $F_k$ ;  $a_{ij}$  – показатель инверсирования;  $Y_j^{a_{ij}}$  –  $j$ -я переменная формулы  $F_k$ , определяемая по выражению

$$Y_j^{a_{ij}} = \begin{cases} Y_j & \text{при } a_{ij} = 1, \\ \bar{Y}_j & \text{при } a_{ij} = 0, \end{cases}$$

$$a_{ij}, Y_j^{a_{ij}}, F_k \in E, \quad E = \{0, 1\}.$$

В процессе минимизации осуществляется переход от СДНФ к сокращенной дизъюнктивной нормальной форме, а затем к их минимальной дизъюнктивно нормальной форме.

Поскольку существует некоторое множество  $L$ , образуемое набором переменных, на которых формула  $F_k$  не определена, причем эти наборы переменных не могут появиться на входах синтезируемой схемы, то они могут использоваться для образования элементарных конъюнкций минимального ранга, поглощающих максимальное количество исходных наборов переменных из множества  $G$ . Таким образом, в общем случае в выражении (3.1)  $i \in H_k = C_k \cup L$  [1].

На основе полученной минимизированной булевой формулы синтезируется комбинационная схема автомата. Данная концепция заложена в основу созданной автором логической системы имитационного моделирования Cell System, представляющей собой расширение динамической библиотеки Simulink интегрированной системы автоматизации математических и научно-технических расчетов Matlab (рис. 3.1).

Cell System включает в себя МЛИМ – ( $S$ -,  $T$ -,  $H$ -,  $L$ -,  $V$ -ячеек).

Каждый из включенных в библиотеку МЛИМ ориентирован на вычисление определенного класса булевых формул. Вышеупомянутые модули разрабатывались в перечисленной очередности, и для каждого модуля предыдущий является прототипом, для первой указанной ячейки ( $S$ -) прототипом является ячейка, описанная в работе [10]. Разработка каждого нового модуля связана с необходимостью вычисления последующих классов булевых формул, не поддающихся вычислению с по-



мощью уже разработанных. Причем каждая вновь разработанная ячейка наделяется свойством преемственности, т.е. последняя ячейка ( $V$ -) охватывает самую обширную часть классификации, приведенной в работе [14], и является наиболее универсальной по сравнению со всеми другими указанными модулями.

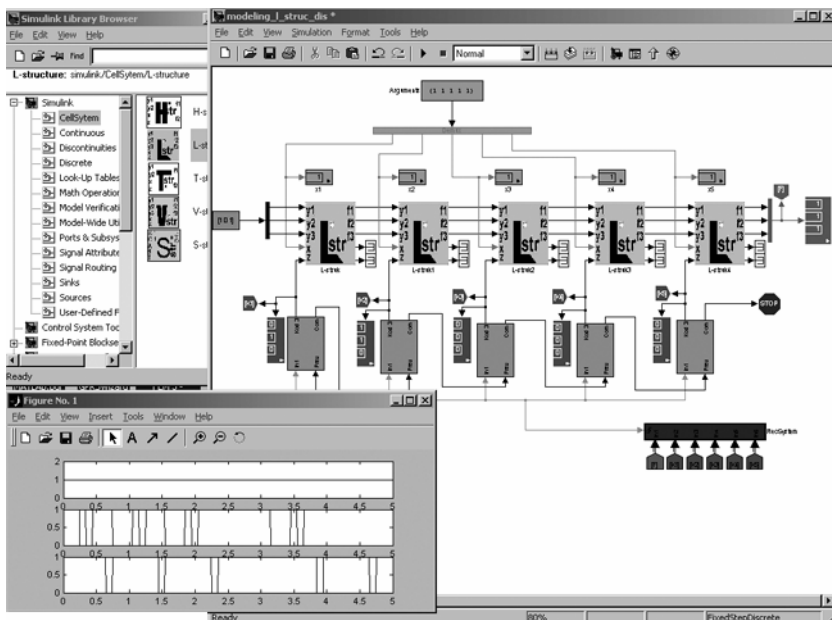


Рис. 3.1 . Система имитационного моделирования

Полученная динамическая библиотека МЛМ позволяет существенно облегчить работу при исследовании и построении на их основе одно-родных сред и самих МЛМ.

Представление моделей в такой форме очень хорошо согласуется с теоретическими выкладками и является отличной основой для более детальной проработки исследовательского материала. Однако когда возникает задача реализации того или иного устройства в промышленности с использованием микропроцессорных средств, появляются трудности с переводом модели, представленной в структурной форме. Для наиболее быстрого перевода модели в микропроцессорные средства

хотелось бы работать с некоей функцией, отражающей поведение устройства. Аргументы функции являются входом устройства, а принимаемые ею значения – его выходом [12].

Кроме структурной реализации, все рассмотренные выше МЛМ реализованы в среде MatLab в качестве функций. Для этого использовались системы булевых формул, описывающие каждую из МЛМ. Данный метод хорошо себя показал и при исследовании, требующем обрабатывать большое количество информации. Например, при определении функциональной полноты  $V$ -структуры необходимо было рассмотреть 256 различных комбинаций; с помощью Cell System на компьютере Pentium Celeron 700 MHz/RAM 256 MB это занимает 0,485 с, включая генерацию восьмиразрядного кода. В результате получается матрица данных размером  $256 \times 11$ , отражающая все возможные состояния МЛМ  $V$ -структуры. Для дальнейшей обработки этой информации создан блок анализа, выполняющий проверку МЛМ на полноту.

Одним из важных достоинств Cell System является ее интеграция со средой MatLab, в результате которой появляется возможность использовать в исследованиях наработки ученых всего мира. После исследования свойств вновь разработанного модуля и проверки его работоспособности следует рассмотреть вопрос о его практическом применении. Подобные модули могут широко использоваться в области скоростной обработки информации, а также в перспективном направлении – в системах автоматического регулирования. Построенные на основе МЛМ устройства управления наделяют всю систему автоматического регулирования новыми динамическими свойствами, такими как инвариантность к параметрическим возмущениям и повышение быстродействия, не присущие ей при использовании традиционных линейных законов регулирования. Исследование полученной системы аналитическими методами становится неэффективным, а иногда и невозможным. Так, единственным способом исследования на устойчивость систем автоматического регулирования более чем с одной нелинейностью (например, реальные физические системы), пожалуй, является имитационное моделирование.

### 3.3. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ИЗОТРОПНЫХ СРЕД

Рассмотрим однородную среду, представляющую собой линейную структуру, состоящую из пяти МЛМ (рис. 3.2). Каждая МЛМ описывается системой булевых формул вида (2.17).

Используя интерпретирующий метод [6, 8] для структуры, где каждая МЛМ имеет  $m = 4$  входа, можно подсчитать объем оперативной памяти, необходимой для запоминания любой бесповторной булевой формулы: он будет равен  $4s$  бит, где  $s$  – число аргументов. Это значит, что имеется реальная возможность реализации булевых формул, зависящих от сотен аргументов.

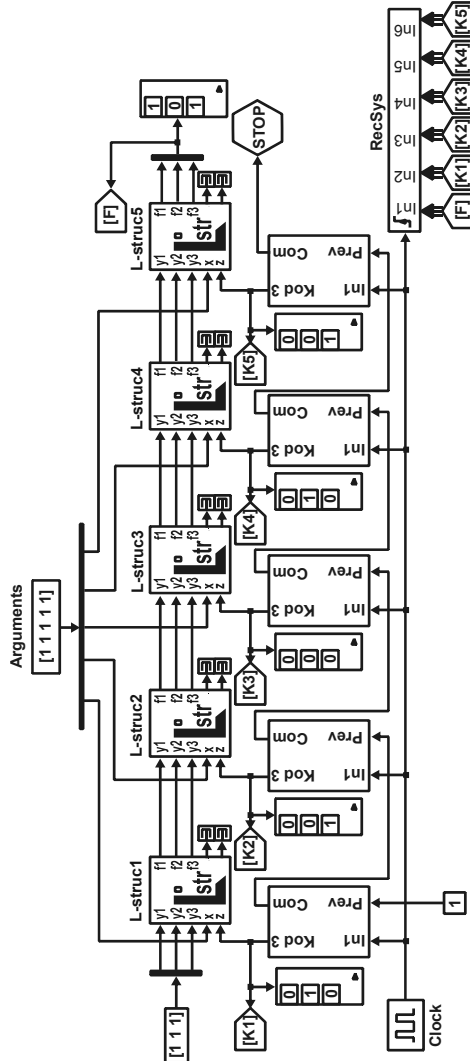


Рис. 3.2. Реализация однородной линейной среды из МЛМ L-структуры

Рассматриваемая ОС состоит из пяти МЛМ, описываемых системой (2.17), имеет три режимных входа ( $y_1$ ,  $y_2$  и  $y_3$ ). Следовательно, возможны восемь режимов его работы. Все они были экспериментально проверены в процессе имитационного моделирования. В левой части табл. 3.1 приведены все возможные

настроечные коды, правая часть разделена на восемь областей, каждой области соответствует определенный режим работы ОС. Режимы закодированы в тех же обозначениях, что и настроечные коды. Первый слева символ кода режима соответствует входу  $y_1$ , второй –  $y_2$  и третий –  $y_3$ .

Т а б л и ц а 3.1

**Таблица истинности функций  $f_1, f_2, f_3$  ОС  
при подаче на все МЛМ настроечного кода  $z_1, z_2, z_3$**

Настроеч- ные коды $z_1, z_2, z_3$	000	001	010	011	100	101	110	111
	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$
000	000	001	010	011	100	101	110	111
001	000	001	010	011	100	101	110	111
010	100	101	110	111	100	101	110	111
011	011	011	011	011	111	111	111	111
100	100	101	100	101	100	101	110	111
101	010	011	010	011	110	111	110	111
110	110	111	110	111	110	111	110	111
111	010	010	010	011	110	110	110	111

После проверки работоспособности имитационной модели МЛМ была исследована работа всей ОС на множестве контрольных примеров.

Дальнейшее более тщательное исследование работы имитационной модели ОС выявило некоторые особенности ее функционирования, которые на этапе логического анализа никак не проявлялись. Суть их в следующем.

**Первый случай.** Если на режимные входы первого МЛМ подать 0, т.е. принять  $y_1 = y_2 = y_3 = 0$ , и каждый из пяти МЛМ настроить на код 011, а на информационные входы подать единичные значения, то на выходе последнего МЛМ получим (см. табл. 3.1):

$$f_1 = 0, f_2 = f_3 = 1.$$

Хотя, как следует из формулы (2.17), надо было ожидать  $f_1 = f_3 = 0$ ;  $f_2 = 1$ , поскольку все МЛМ построены совершенно одинаково. Это объясняется тем, что режимный вход второго МЛМ при указанных условиях изменяется и устанавливается равным 010.

**Второй случай** обнаружен при этом же настроечном коде (011), поданном на все МЛМ, единичных значениях всех логических аргументов и режиме 100 первого МЛМ. Установлено, что  $f_1 = f_2 = f_3 = 1$ , хотя по формуле (2.17) следовало ожидать  $f_1 = f_2 = 1$ ;  $f_3 = 0$ , поскольку, как и в первом случае, все МЛМ настроены совершенно одинаково.

**Третий случай** обнаружен при настроечном коде 110, поданном на все МЛМ, единичных значениях всех логических аргументов и режиме 001 первого МЛМ. Было установлено, что  $f_1 = f_2 = f_3 = 1$ , хотя по формуле (2.17) следовало ожидать  $f_1 = f_3 = 1$ ;  $f_2 = 0$ .

На следующем этапе имитационного моделирования ОС различные ячейки настраивались на различные настроечные коды. Поскольку всего существует  $2^{15}$  различных вариантов настройки ОС, составленной из пяти МЛМ, то методом случайной выборки было определено несколько вариантов настройки всей ОС. Результаты вычисления десяти функций для единичных значений логических аргументов представлены в табл. 3.2.

Т а б л и ц а 3.2

Таблица истинности функций  $f_1, f_2, f_3$  при случайной выборке для каждой МЛМ линейной ОС

Настроечные коды					000	001	010	011	100	101	110	111
$z_1, z_2, z_3$					$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$	$f_1, f_2, f_3$
1	2	3	4	5								
000	111	001	010	110	110	110	110	111	110	110	110	111
000	000	110	110	001	110	111	110	111	110	111	110	111
101	110	001	010	001	110	111	110	111	110	111	110	111
000	000	001	100	011	110	111	110	111	110	111	111	111
011	101	101	001	000	010	011	011	011	110	111	111	111
001	001	110	110	000	110	111	110	111	110	111	110	111
111	011	111	111	111	011	011	011	011	111	111	111	111
001	010	110	110	111	110	111	110	111	110	111	110	111
101	110	000	000	101	110	111	110	111	110	111	110	111
011	100	011	101	110	110	111	111	111	111	111	111	111

Таким образом, если режимные и настроечные коды каждого МЛМ рассматривать как обобщенный шестизрядный код, то:

а) первый МЛМ может быть настроен на любой из 64 возможных обобщенных кодов;

б) для всех остальных МЛМ существуют обобщенные коды, которые не могут быть поданы на входы этих МЛМ.

Такой интересный результат оказалось возможным получить лишь в результате имитационного моделирования ОС [14].

**Т е о р е м а 3.1.** Если при настройке МЛМ ее выходы  $f_i$  тождественно равны информационным входам  $y$ , то построенная линейная ОС, состоящая из  $n$  таких модулей, соответствует одному при такой же настройке.

**Д о к а з а т е л ь с т в о.** Пусть линейная ОС состоит из  $n$  ячеек, настроенных таким образом, что выполняется равенство

$$f_{i,n} = y_{i,n}, \quad (3.2)$$

где  $i = 1, 2, 3$  – соответствующий номер входа и выхода ячейки.

Исходя из того, что структура является линейной, т.е. каждый вход последующей ячейки равен выходу предыдущей

$$y_{i,n} = f_{i,n-1},$$

учитывая (3.2), можно записать

$$f_{i,n} = y_{i,n} = f_{i,n-1}, \text{ или } f_{i,n} = f_{i,n-1} = \dots = f_{i,1} = y_{i,1}.$$

Что и требовалось доказать.

**Т е о р е м а 3.2.** Для каждого единичного множества  $A_{i+\bar{x}}$  ( $i = 1, 3, \dots, 2^n - 1$ ), состоящего из значения кода  $F$  на выходе МЛМ, при условии, что

$$F = Y, \quad (3.3)$$

где  $Y$  – код на входе;  $n$  – число выходов ячейки;  $x$  – значение аргумента и связанного с ним множества настроечных кодов  $B_{i+\bar{x}}$ , существует такое множество  $A_{i+1-\bar{x}}$ , при котором  $B_{i+\bar{x}} = B_{i+1-\bar{x}}$ , кроме случая, когда множеству  $A_{i+1-\bar{x}}$  принадлежит элемент, к старшему разряду  $\alpha$  которого два раза справа примыкает значение аргумента  $x$  ( $\alpha x x \in A_{i+1-\bar{x}}$ ), тогда  $B_{i+1-\bar{x}} = B_{i+\bar{x}} \cup \{\bar{x}11\}$ .

**Доказательство.** Обозначим каждое значение из 3-разрядного кода соответствующей буквой латинского алфавита в порядке возрастания, всего получится  $2^3$  букв.

По формуле, описывающей  $L$ -структуру (2.17), найдем коды настройки, при которых выполняется условие (3.3).

Пусть значение аргумента  $x = 1$ :

$$A_1 = \{a\}, B_1 = \{a, b\}; A_2 = \{b\}, B_2 = \{a, b\};$$

$$A_3 = \{c\}, B_3 = \{a, b, f, h\}; A_4 = \{d\}, B_4 = \{a, b, d, f, h\};$$

$$A_5 = \{e\}, B_5 = \{a, b, c, e\}; A_6 = \{f\}, B_6 = \{a, b, c, e\};$$

$$A_7 = \{g\}, B_7 = \{a, b, c, e, f, g, h\}; A_8 = \{h\}, B_8 = \{a, b, c, d, e, f, g, h\}.$$

Отметим, что  $B_1 = B_2$ ,  $B_5 = B_6$ , а симметрическая разность множеств

$$B_3 \oplus B_4 = B_3 \cap \bar{B}_4 \cup \bar{B}_3 \cap B_4 = \{d\} \text{ и } B_7 \oplus B_8 = \{d\}$$

дает нам один результат – элемент  $d = 011$  или  $d = \bar{x}11$ .

Аналогично при  $x = 0$ :

$$A_1 = \{a\}, B_1 = \{a, b, c, d, e, f, g, h\}; A_2 = \{b\}, B_2 = \{a, b, c, d, e, f, g\};$$

$$A_3 = \{c\}, B_3 = \{a, c, f, g\}; A_4 = \{d\}, B_4 = \{a, c, f, g\};$$

$$A_5 = \{e\}, B_5 = \{b, c, d, f, h\}; A_6 = \{f\}, B_6 = \{b, c, d, f\};$$

$$A_7 = \{g\}, B_7 = \{c, f\}; A_8 = \{h\}, B_8 = \{c, f\}.$$

Следовательно,  $B_3 = B_4$ ;  $B_7 = B_8$ .

Симметрическая разность даст нам  $B_1 \oplus B_2 = B_5 \oplus B_6 = \{h\}$ .

Таким образом, при  $x = 0$  существует равенство между множествами  $B_3$  и  $B_4$ , а также между  $B_7$  и  $B_8$ , множества же  $B_1$  и  $B_2$ , как, в свою очередь,  $B_5$  и  $B_6$ , отличаются друг от друга на элемент  $h = 111$  или  $h = \bar{x}11$  при принадлежности единичному множеству  $A_1$  элемента  $a = 000 = \alpha x x$ , или множеству  $A_5$  элемента  $e = 100 = \alpha x x$ , где  $\alpha$  – дво-

ичное число, соответствующее старшему разряду кода на выходе МЛМ. Что и требовалось доказать.

С л е д с т в и е из теорем 3.1 и 3.2.

Если в линейной ОС есть ячейки с настроечными кодами из набора  $B_i$  и кодом на входе  $A_i$ , то считается, что они удалены из структуры (т.е. осуществляется операция пропуска аргумента).

Таким образом, имитационное моделирование изотропных структур позволило провести проверку работоспособности этих структур, выявить ряд интересных результатов настройки структур обобщенными кодами, а также исключить все ошибки, которые могли возникнуть при объединении отдельных ячеек изотропной структуры.

При моделировании изотропных сред интерпретирующий метод можно использовать как для решения проблемы программного нахождения значений булевых формул практически неограниченного числа аргументов, так и для анализа этих сред.

### 3.4. СИНТЕЗ ЛИНЕЙНЫХ ИЗОТРОПНЫХ СРЕД

Рассмотрим четыре случая построения однородных сред.

1. Боковые входы ОС задействованы для подачи значений аргументов и в процессе функционирования могут принимать значения из множества  $\{0, 1\}$ .

2. Осуществляется поворот с одного из выходов ОС на один из ее входов, при этом на боковые входы также подаются значения аргументов.

3. Фиксированный код 111 подается на боковые входы ОС и в процессе функционирования не изменяется (вариант режимных входов).

4. Осуществляется поворот с одного из выходов ОС на один из ее входов, при этом на боковые входы подаются константы (режимные входы).

С л у ч а й 1 д л я  $L$ -с т р у к т у р ы. Зададимся системой булевых формул вида

$$\begin{cases} f_1 = x_6 \vee x_7, \\ f_2 = x_4 x_5, \\ f_3 = x_1 \vee x_2 x_3. \end{cases} \quad (3.4)$$



Соответствующая ей однородная среда, построенная на базе  $L$ -структуры с настройочными кодами, представлена на рис. 3.3.

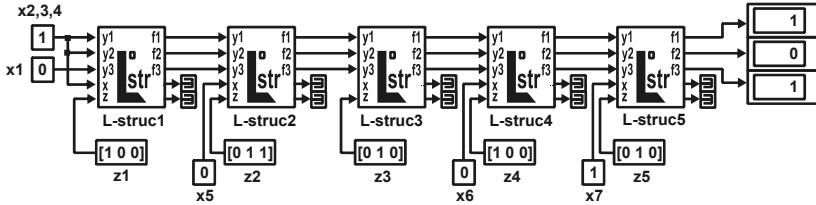


Рис. 3.3. Однородная среда на МЛМ  $L$ -структуры, реализующая систему (3.4)

В данном случае на функциональный вход  $x$  третьей ячейки подается нуль, чтобы в дальнейшем подавать в качестве аргумента выходные значения формул с пятой ячейки ОС, т.е. осуществлять поворот.

С л у ч а й 2 д л я  $L$ -с т р у к т у р ы. Будем поочередно производить поворот с каждого выхода ОС ( $f_1, f_2, f_3$ ) на вход третьей ячейки. На рис. 3.4 отражена структура ОС при повороте первого выхода  $f_1$ .

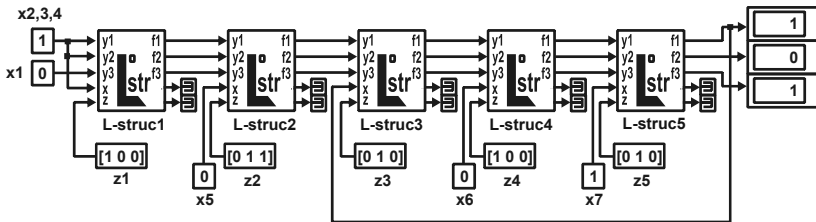


Рис. 3.4. Однородная среда в случае 2 для  $L$ -структуры

Система булевых формул в этом случае принимает вид

$$\begin{cases} f_1 = x_6 \vee x_7, \\ f_2 = (x_4 \vee x_6 \vee x_7)x_5, \\ f_3 = x_1 \vee x_2x_3. \end{cases}$$

В случае введения поворота со второго выхода пятой ячейки  $f_2$  ОС система булевых формул, описывающая поведение всей ОС, преобразуется к форме

$$\begin{cases} f_1 = x_6 \vee x_7, \\ f_2^{t+1} = [(x_4 \vee f_2)x_5]^t, \\ f_3 = x_1 \vee x_2x_3. \end{cases}$$

Здесь и в дальнейшем индекс  $t$ , расположенный над скобкой, указывает, что значения переменных, находящихся в скобках, относятся к моменту времени  $t$ .

И, наконец, в случае поворота третьего выхода  $f_3$  система булевых формул, описывающая ОС, примет вид

$$\begin{cases} f_1 = x_6 \vee x_7, \\ f_2 = (x_1 \vee x_2x_3 \vee x_4)x_5, \\ f_3 = x_1 \vee x_2x_3. \end{cases}$$

Реализация системы булевых формул (3.4) в однородной линейной среде из  $V$ -структур оказалась затруднительной, однако применение принципов квазиоднородных сред позволило решить эту задачу (рис. 3.5).

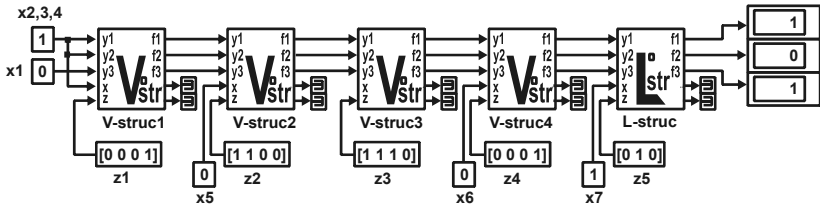


Рис. 3.5. Квазиоднородная среда, реализующая систему (3.4)

Случай 1 для  $V$ -структуры. Рассмотрим реализацию систем булевых формул вида

$$\begin{cases} f_1 = 0, \\ f_2 = (x_3 \vee x_4x_5)x_7, \\ f_3 = x_1x_2 \vee x_5 \vee x_6. \end{cases} \quad (3.5)$$

Соответствующая ей однородная среда, построенная на базе  $V$ -структуры с настроенными кодами, представлена на рис. 3.6.

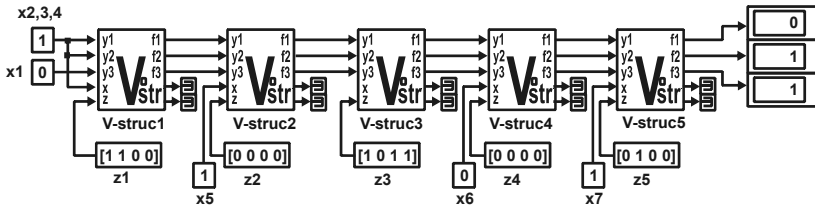


Рис. 3.6. Однородная среда на МЛМ  $V$ -структуры, реализующая систему (3.5)

Случай 2 для  $V$ -структуры. Аналогично рассмотренному выше примеру будем последовательно заводить выходы ОС ( $f_1$ ,  $f_2$ ,  $f_3$ ) на функциональный вход  $x$  третьей ячейки. Система формул при повороте с первого выхода  $f_1$  примет вид

$$\begin{cases} f_1^{t+1} = 0, \\ f_2 = [(x_3 \vee x_4 x_5 \vee f_1)x_7]^t, \\ f_3 = x_1 x_2 \vee x_5 \vee x_6. \end{cases}$$

В случае введения поворота со второго выхода  $f_2$  система формул преобразуется к виду

$$\begin{cases} f_1 = f_2 x_6, \\ f_2^{t+1} = [(x_3 \vee x_4 x_5 \vee f_2)x_7]^t, \\ f_3 = x_1 x_2 \vee x_5 \vee x_6. \end{cases}$$

И, наконец, в случае поворота третьего выхода  $f_3$  система булевых формул, описывающая ОС, примет вид

$$\begin{cases} f_1 = (x_1 x_2 \vee x_5 \vee x_6)x_6, \\ f_2 = (x_1 x_2 \vee x_3 \vee x_4 x_5 \vee x_5 \vee x_6)x_7, \\ f_3 = x_1 x_2 \vee x_5 \vee x_6. \end{cases}$$

Случай 3 для  $V$ -структуры. Рассмотрим реализацию булевой формулы вида

$$f = (x_1 \vee x_2)x_2 x_3 \vee x_4 \tag{3.6}$$

с помощью ОС, построенной на  $V$ -структурах (рис. 3.7). В данном случае осуществлять поворот будем только с первого выхода  $f_1$ , а реализацию формулы (3.6) будем производить на втором выходе  $f_2$ .

При настроечном коде 0110 на третьей ячейке происходит операция пропуска аргумента (ячейка принимает структуру, приведенную на рис. 2.20,  $e$ ), поэтому как такового поворота в ОС нет, т.е. если проводить аналогию с теорией автоматического управления, то цепь обратной связи в этом состоянии разомкнута.

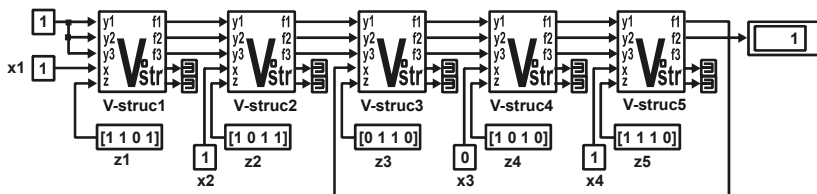


Рис. 3.7. Однородная среда к случаю 3 для  $V$ -структуры

С л у ч а й 4 для  $V$ -с т р у к т у р ы. Выясним, как влияет введение поворота на реализацию булевой формулы. Для этого изменим настроечный код третьей ячейки (рис. 3.7) на 1110. В итоге реализуемая средой булева формула примет вид

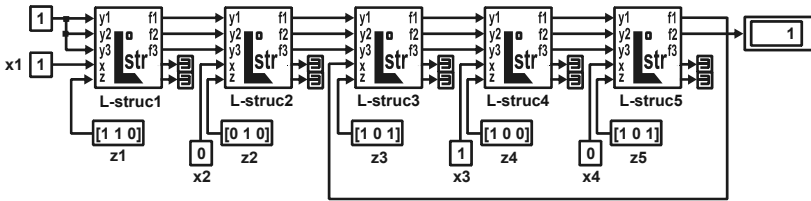
$$f = (x_1 \vee x_2 \vee x_3)x_2x_3 \vee x_4.$$

С л у ч а й 3 для  $L$ -с т р у к т у р ы. Рассмотрим аналогичную ситуацию для ОС, построенной на  $L$ -структурах. Реализуем формулу вида

$$f = x_1 \vee x_2 \vee x_4.$$

Соответствующая ей однородная среда с настроечными кодами представлена на рис. 3.8.

Аналогично вышерассмотренному примеру на третьей ячейке осуществляется операция пропуска аргумента; в этом случае потребовались боковые, режимные входы ОС. Данный аспект связан с тем, что  $L$ -ячейка не имеет явной структуры, выполняющей пропуск аргумента, поэтому при необходимости эту операцию нужно вводить искусственно. Таким образом, при настроечном коде 101 третья ячейка ОС не осуществляет поворот.

Рис. 3.8. Однородная среда к случаю 3 для  $L$ -структуры

С л у ч а й 4 д л я  $L$ -с т р у к т у р ы. При изменении настроечного кода третьей ячейки на 001, т.е. при осуществлении операции «разрешения» поворота, реализуемая формула примет вид

$$f = (x_1 \vee x_2)x_3 \vee x_4.$$

Таким образом, в ОС, реализуемых на  $L$ - или  $V$ -структурах, можно осуществлять операции «разрешение» или «запрет» поворота, не изменяя комбинационных связей между ячейками, а изменяя только настроечный код той, на которую заведен поворот. Следует также отметить, что операция «поворот» требует ввода дополнительной ячейки в структуру ОС. Введение же поворота в канал, с которого он осуществляется (замкнутый цикл), позволяет строить многотактные автоматы на основе МЛМ  $L$ - и  $V$ -структур.

### 3.5. ДЕКОМПОЗИЦИЯ ЛИНЕЙНЫХ ИЗОТРОПНЫХ СРЕД

Задачей декомпозиции линейных изотропных сред является исследование и изучение их реализаций при помощи МЛМ, соединенных между собой.

Пусть  $M_1, M_2, \dots, M_n$  – совокупность МЛМ, образуемых при декомпозиции ОС. Обозначим через  $F_i$ ,  $G_i$  соответственно множества выходных и входных переменных ячейки  $M_i$ . Для каждой ячейки  $M_i$  множество  $G_i$  определяется векторным произведением подмножеств  $Y_i, X_i$  и  $Z_i$ .

$Y_i$  обозначает множество входных переменных ячейки  $M_i$ , являющихся выходными переменными других ячеек,  $X_i$  – множество внешних входных аргументов, приложенных непосредственно к ячейке

$M_i$ , а  $Z_i$  – множество внешних входных переменных, отражающих настройку внутренней структуры ячейки  $M_i$ .

Приведем несколько определений.

**О п р е д е л е н и е 1.** Будем считать, что взаимное соединение ячеек  $M_1, M_2, \dots, M_n$  действует одновременно, если выходная переменная  $F_i$  каждой  $M_i$ -й ячейки ( $i=1, 2, \dots, n$ ) зависит от состояний входов  $Y_i$ , определяемых текущим структурным состоянием других ячеек, к которым подсоединена  $M_i$ , и от состояния внешних входов  $X_i$  и определяющих внутреннюю структуру входов  $Z_i$ .

Таким образом, упорядоченное множество из  $n$  элементов, образованных  $n$  состояниями ячеек, является состоянием всей ОС.

**О п р е д е л е н и е 2.** Подмножество взаимно соединенных МЛМ  $M_{i1}, M_{i2}, \dots, M_{ip}$  образуют петлю, если выход модуля  $M_{ik}$  является входом модуля  $M_{i(k+1)}$  ( $k=1, 2, \dots, p-1$ ) и если выход модуля  $M_{ip}$  является входом модуля  $M_{i1}$ .

**О п р е д е л е н и е 3.** Ячейка  $M_i$  называется предшествующей ячейке  $M_j$ , если одним из входов ячейки  $M_j$  является выход ячейки  $M_i$ . Подмножество ячеек  $M_{i1}, M_{i2}, \dots, M_{ip}$ , соединенных между собой, называют замкнутым, если все ячейки этого подмножества являются предшествующими. Для заданной ОС наименьшее замкнутое подмножество, содержащее  $M_i$ , будем обозначать  $C(M_i)$ .

**О п р е д е л е н и е 4.** Будем считать, что линейная ОС реализуется при помощи множества ячеек  $M_1, M_2, \dots, M_n$ , действующих одновременно и соединенных без обратных связей в том случае, если выполняются следующие условия:

- 1) ячейки  $M_1, M_2, \dots, M_n$  действуют одновременно и соединены без обратных связей (определения 1 и 2);
- 2) множество входов  $M$  является подмножеством множества возможных входов ОС, реализованного при помощи  $M_1, M_2, \dots, M_n$ ;

3) существует взаимно однозначное соответствие между множеством структурных состояний  $M$  и подмножеством декартова произведения множеств структурных состояний ячеек  $M_1, M_2, \dots, M_n$ .

Таким образом, если ОС и ячейки  $M_1, M_2, \dots, M_n$  находятся в одном из исходных структурных состояний, то после подачи некоторой входной последовательности  $X$  они окажутся в соответствующих конечных структурных состояниях.

**Основная теорема декомпозиции**

**Теорема 3.3.** Пусть  $M$  – линейная ОС, а  $M_1, M_2, \dots, M_n$  – множество ячеек, работающих одновременно, соединенных без обратных связей и выполняющих те же переходы между состояниями, что и  $M$ . При таких условиях каждому замкнутому подмножеству  $C(M_i)$  этих ячеек соответствует подстановочное разбиение  $\pi_i$ . Множество этих разбиений обладает свойством

$$\prod_{i=1}^n \pi_i = 0.$$

Кроме того, из условия  $C(M_i) \leq C(M_j)$  следует, что  $\pi_i \geq \pi_j$ , и наоборот, каждому множеству подстановочных разбиений  $T = (\pi_1, \pi_2, \dots, \pi_n)$  ОС  $M$  при  $\prod_{i=1}^n \pi_i = 0$  соответствует реализация этой ОС при помощи  $n$  соединенных между собой без обратных связей ячеек, действующих одновременно. При такой реализации выход ячейки  $M_i$  может быть входом ячейки  $M_j$  только в том случае, если  $\pi_i > \pi_j$ . В разбиении  $\pi_i$  каждый из блоков содержит все те состояния ОС  $M$ , которым соответствует один и тот же набор состояний ячеек, образующих подмножество  $C(M_i)$ .

**Доказательство:**

**Первая часть.** Пусть ОС  $M$  состоит из ячеек  $M_1, M_2, M_3$ , каждая из которых принимает по две структуры, определяемые настроечными кодами  $Z_1, Z_2$  и  $Z_3$  соответственно.

Следовательно, каждый настроечный код может принимать значения из двух наборов, например  $Z_i = \{z_{i1}^1 z_{i2}^1 z_{i3}^1 z_{i4}^1, z_{i1}^2 z_{i2}^2 z_{i3}^2 z_{i4}^2\}$ , тогда для

$i = 1, 2, 3$  существует множество  $S = \{s_1, s_2, \dots, s_8\}$ , состоящее из  $2^3 = 8$  элементов, отражающих состояние ОС  $M$ . Элементы множества  $S$  получают путем обобщения настроечных кодов каждой ячейки. Замкнутое множество  $C(M_2)$  образуется ячейками  $M_1$  и  $M_2$ . Ячейка  $M_1$  определяет переменную  $F_1$  и имеет структуры  $v_1$  и  $v_2$ ;  $v_1$  соответствует настроечным кодам  $s_1, s_2, s_7$  и  $s_8$  ОС  $M$ ,  $v_2$  – настроечным кодам  $s_3, s_4, s_5$  и  $s_6$ . Ячейка  $M_2$  определяет переменную  $F_2$ , обладает структурами  $v_3$  и  $v_4$  с настроечными кодами  $s_2, s_4, s_5, s_7$  и  $s_1, s_3, s_6, s_8$  соответственно.

Таким образом, в множестве  $C(M_2)$  состояния ОС  $M$  представляются следующим образом:

$$\begin{array}{llll} s_1 \rightarrow v_1, v_4 & s_3 \rightarrow v_2, v_4 & s_5 \rightarrow v_2, v_3 & s_7 \rightarrow v_1, v_3 \\ s_2 \rightarrow v_1, v_3 & s_4 \rightarrow v_2, v_3 & s_6 \rightarrow v_2, v_4 & s_8 \rightarrow v_1, v_4. \end{array}$$

Разбиение, соответствующее  $C(M_2)$  [ $\pi = s_1 s_8, s_2 s_7, s_3 s_6, s_4 s_5$ ] =  $\pi_1$ ], содержит в одном блоке структурные состояния  $M$ , которые имеют одно и то же представление в этом замкнутом множестве.

Покажем теперь, что в общем случае разбиение  $\pi_i$ , соответствующее  $C(M_i)$ , является подстановочным. Для этого заметим, что по определению замкнутого подмножества ячеек (определение 3) структурное состояние ячеек, входящих в  $C(M_i)$ , может быть определено по внешнему (функциональному) входному и настроечному входу ячеек из  $C(M_i)$ . Таким образом, если два настроечных кода из множества  $S$  ОС  $M$ , отражающих структуру  $v_1$  и  $v_2$ , содержатся в одном блоке разбиения  $\pi_i$ , то для каждой возможной входной переменной  $X_k$  информация о выходных переменных будет также содержаться в том же блоке  $\pi_i$ , т.е. этот блок определяется входной переменной  $X_k$  и настроечными кодами тех ячеек, которые содержатся в  $C(M_i)$ . Следовательно, разбиение  $\pi_i$  является подстановочным.



Кроме того, если  $C(M_j)$  содержит по меньшей мере все ячейки из  $C(M_i)$  [ $C(M_j) \geq C(M_i)$ ], то каждый блок из  $\pi_j$  будет содержаться в одном из блоков разбиения  $\pi_i$  и, следовательно,  $\pi_i \geq \pi_j$ . В рассматриваемом случае подмножеству  $C(M_1)$ , т.е. просто  $M_1$ , соответствует разбиение  $\pi_3 = (s_1s_2s_7s_8, s_3s_4s_5s_6)$ , а замкнутому подмножеству  $C(M_2)$ , содержащему  $M_1$  и  $M_2$ , соответствует разбиение  $\pi_1 = (s_1s_8, s_2s_7, s_3s_6, s_4s_5)$ . Таким образом,  $C(M_2) \geq C(M_1)$  при  $(s_1s_8, s_2s_7, s_3s_6, s_4s_5) \leq (s_1s_2s_7s_8, s_3s_4s_5s_6)$ . Что и требовалось доказать.

**В т о р а я ч а с т ь.** Пусть число разбиений  $n$  равно числу ячеек реализуемой декомпозиции ОС и ячейка  $M_j$  может получать информацию от  $M_i$ , т.е. быть соединенной с  $M_i$  только тогда, когда  $\pi_j \leq \pi_i$ .

Доказательство осуществляется в пять этапов.

1. Прежде рассмотрим все те разбиения множества  $T$ , которые не меньше любого другого разбиения этого множества. Пусть  $T_1$  – множество таких разбиений. Каждое разбиение из  $T_1$  определяет ячейку, и каждая из этих ячеек не будет получать информацию от других ячеек, т.е. будет иметь только внешние входы. Структура ячеек определяется блоками соответствующих разбиений. Следовательно, каждая ячейка будет иметь столько структур, сколько блоков содержится в соответствующем разбиении. К этому типу относятся ячейка  $M_1$  и разбиение  $\pi_3$  из доказательства первой части теоремы.

2. Если  $T_1 = T$ , то ОС  $M$  может быть реализована при помощи  $n$  ячеек, работающих параллельно. В этом случае число структур  $M_i$  равно числу блоков в разбиении  $\pi_i$ .

3. Если  $T_1 \neq T$ , рассмотрим разбиения множества  $T - T_1$ . Выберем в этом новом множестве те разбиения, которые не меньше любого другого разбиения из этого множества. Пусть такие разбиения образуют множество  $T_2$ .

4. Для каждого разбиения  $\pi_j$  из множества  $T_2$  рассмотрим все те разбиения  $\pi_i$  из множества  $T_1$ , которые больше  $\pi_j$ . Определим пересечение  $\pi_j^* = \prod \pi_i$  при  $\pi_i \in T_1 > \pi_j \in T_2$ .

Обозначим через  $l_j = [\pi_j / \pi_j^*]$  наибольшее число блоков  $\pi_j$ , содержащихся в одном блоке  $\pi_j^*$ . Для выбора МЛМ  $M_j$ , соответствующей разбиению  $\pi_j$ , будем учитывать, что существует разбиение  $\tau_j$  такое, что  $\tau_j \geq \pi_j$  ( $\tau_j$  содержит  $l_j$  блоков) и  $\tau_j \cdot \pi_j^* = \pi_j$ . Наконец, поскольку  $\pi_j$  является подстановочным разбиением и  $\tau_j \geq \pi_j$ ,  $(\pi_j, \tau_j)$  являются парой разбиений.

Таким образом, можно выбрать МЛМ  $M_j$ , имеющую  $l_j$  структур, соответствующих  $l_j$  блокам разбиения  $\tau_j$ . Эта ячейка получает информацию только от ячеек, соответствующих тем разбиениям из множества  $T_1$ , которые больше  $\pi_j$ .

5. Если  $T \neq T_1 + T_2$ , то способом, аналогичным предыдущему, строят следующий каскад декомпозиции до тех пор, пока не будут рассмотрены все разбиения  $T$ . Так, будут рассмотрены разбиения из множества  $T - (T_1 + T_2)$ , которые не меньше любого другого разбиения из этого нового множества. Они образуют множество  $T_3$ , исходя из которого можно определить соответствующие ячейки, получающие информацию от ячеек множества  $T_2$  и т.д. Что и требовалось доказать.

Таким образом, предложено решение задачи декомпозиции линейных изотропных сред.

### 3.6. ВЫВОДЫ

Отметим основные результаты, полученные в данной главе:

1. Впервые создана логическая система имитационного моделирования Cell System для рассматриваемого класса многофункциональных логических модулей. Таким образом, у разработчика появляется мощ-

ный инструментарий, позволяющий синтезировать и исследовать МЛИМ, построенные на их основе изотропные и квазиизотропные среды и их взаимодействие с системами, имеющими практическую направленность в той или иной области науки и техники.

2. Установлено, что при моделировании изотропных сред интерпретирующий метод можно использовать как для решения проблемы программного нахождения значений булевых функций практически неограниченного числа аргументов, так и для анализа этих сред.

3. Впервые решены задачи реализации операций «разрешение» и «запрет» поворота без изменения комбинационной связи между ячейками.

4. Предложено решение задачи декомпозиции линейных изотропных сред.

## ЛИТЕРАТУРА

1. Баранов С.И., Скляр В.А. Цифровые устройства на программируемых БИС с матричной структурой. – М.: Радио и связь, 1986. – 272 с.
2. Гаврилов М.А., Девятков В.В., Пупырев Е.И. Логическое проектирование дискретных автоматов. – М.: Наука, 1997. – 352 с.
3. Евреинов Э.В. Однородные вычислительные системы, структуры и среды. – М.: Радио и связь, 1981. – 208 с.
4. Каляев А.В. Многопроцессорные системы с программируемой архитектурой. – М.: Радио и связь, 1984. – 240 с.
5. Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов. – М.: Наука, 1985. – 320 с.
6. Лазарев В.Г., Пийль Е.И., Турута Е.Н. Построение программируемых управляющих устройств. – М.: Энергоатомиздат, 1984. – 192 с.
7. Однородные управляющие структуры адаптивных роботов / А.В. Каляев и др. – М.: Наука, 1990. – 152 с.
8. Пупырев Е.И. Перестраиваемые автоматы и микропроцессорные системы. – М.: Наука, 1984. – 192 с.
9. Фет Я.И. Параллельные процессоры для управляющих систем. – М.: Энергоатомиздат, 1981. – 160 с.
10. Шальто А.А. Многофункциональный логический модуль. А.с. № 798802 (СССР) // Б.И. – 1981. – № 3.
11. Шальто А.А. Логическое управление. Методы аппаратной и программной реализации алгоритмов. – СПб.: Наука, 2000. – 780 с.

12. *Шидловский С.В.* Автоматизация синтеза перестраиваемых структур // Известия Томского политехнического университета. – 2006. – № 1. – С. 190–192.

13. *Шидловский С.В.* Имитационное моделирование однородной среды // Докл. Томского государственного университета систем управления и радиоэлектроники. Т. 8: Автоматизированные системы обработки информации, управления и проектирования: Сб. научных трудов. – Томск: ТУСУР, 2003. – С. 99–103.

14. *Шидловский С.В.* Ячейка однородной среды. Патент РФ на изобретение № 2251141 // Б.И. – 2005. – № 12. – 13 с.

15. *Шидловский С.В., Светлаков А.А.* Применение информационных технологий при структурном моделировании элементов и устройств систем автоматического управления // Материалы региональной науч.-методич. конф. «Современное образование: Системы и практика обеспечения качества». – Томск: ТУСУР, 2002. – С. 79–80.

## ДИНАМИКА СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

---

---

### 4.1. ВВОДНЫЕ ПОНЯТИЯ

Автоматизация производства – этап машинного производства, характеризуемый освобождением человека от непосредственного выполнения функций управления производственными процессами и передачей этих функций автоматическим устройствам. При автоматизации производства процессы получения, преобразования, передачи и использования энергии, материалов и информации выполняются автоматически [18].

Полная автоматизация является составной частью технического процесса и обеспечивает наивысшую технико-экономическую эффективность, поэтому одним из важнейших критериев целесообразности проводимой автоматизации должны быть результаты предшествующих экономических исследований.

Отличительной особенностью любого машинного производства является частичная автоматизация. В рамках рассмотрения конкретного технологического процесса управление сводится к реализации воздействию средств труда на предмет труда в заданной последовательности. Определение этой последовательности и есть определение закона управления данным процессом. Для простых технологических процессов, происходящих всегда при одних и тех же условиях, закон управления может быть представлен в виде совершенно определенной и неизменной программы движений орудий труда относительно обрабатываемого материала – предмета труда.

Комплексная автоматизация – системный охват автоматизации производственных и экономико-административных процессов в рамках агрегата, отдельного технологического процесса, цеха, предприятия и более высоких производственных и хозяйственных формаций. Комплексная автоматизация базируется на достигнутом уровне развития

кибернетики и, в частности, ее разделов – технической кибернетики и экономической кибернетики [18].

Автоматизация играет узловую роль в нашем обществе. Разнообразные автоматические устройства позволяют увеличить выпуск продукции в расчете на одного работающего, чтобы сбалансировать инфляционные издержки и рост заработной платы. Потому в промышленности используется термин «производительность» (в первую очередь – производительность труда) – отношение выпуска продукции к реальным затратам в расчете на один час рабочего времени. Кроме того, промышленные предприятия стремятся постоянно улучшать потребительские качества выпускаемой продукции, что без развитой автоматизации становится весьма затруднительной, а иногда просто неразрешимой задачей. Таким образом, автоматизация позволяет увеличить производительность и повысить качество выпускаемой продукции.

Технологический процесс, протекающий в регулируемом объекте, в общем случае характеризуется несколькими регулируемыми величинами. Соответствие между производительностью и нагрузкой агрегата (объекта) характеризует качественные показатели технологического процесса, и поэтому задача регулирования агрегата обычно заключается в поддержании заданных значений регулируемых величин. Поддержания заданных значений регулируемых величин сводится к своевременному воздействию на соответствующие регулирующие органы.

В большинстве случаев промышленные установки являются сложными регулируемыми объектами с несколькими регулируемыми величинами. Такие объекты можно разделить на ряд простых. В дальнейшем под термином «регулируемый объект» мы будем понимать элементарный объект, характеризуемый одной выходной регулируемой величиной и обладающий одним регулирующим органом.

Такое представление позволяет описать регулируемую величину как некоторую функцию, зависящую от соответствующих воздействий. Тогда сам объект регулирования представляет собой модель «вход – выход», где «выходом» является место контроля регулируемой величины, а «входом» – место приложения регулирующего воздействия. Кроме регулирующего воздействия, на регулируемую величину влияют возмущающие воздействия, вызванные особенностями конкретного технологического процесса. Поэтому в общем случае в объект регулирования могут быть введены дополнительные входы, позволяющие при-

близить описание регулируемой величины к реальному технологическому параметру с заданной степенью детализации.

Входы объекта считаются различными, если возмущающее воздействие одной и той же формы (например, скачкообразное), приложенное к этим входам, дает различные формы переходного процесса. Несколько выходов могут геометрически находиться в одной и той же точке (например, давление и температура рабочего тела на выходе линейного участка в трубопроводном транспорте).

Динамические свойства регулируемых объектов, как и других элементов систем регулирования, могут быть представлены в различных формах.

Строгого определения понятия объекта в общем случае не существует. Оно всегда противопоставляется другому, двойственному ему понятию «субъект». Субъект имеет способность воспринимать, преобразовывать и использовать информацию об объекте. Такая способность называется интеллектом. Объект – одно из самых общих понятий. Трудно привести пример того, что не является объектом. Все, что постигается, эксплуатируется или создается человеком, есть объекты. Объекты могут быть реальными и идеальными, в частности математические, в общем случае на их природу никаких требований не накладывается.

Среди всего разнообразия объектов выделяются так называемые системы, которые представляют собой совокупность элементов (объектов), находящихся в отношениях и связях друг с другом и образующих при этом определенную целостность и единство.

Любой объект обладает определенными свойствами, проявляющимися при взаимодействии с другими объектами. Если зафиксировать некоторую систему и в качестве одного из ее элементов помещать тот или иной объект, то система будет каким-то образом реагировать на эти объекты.

Объектом является все то, о чем возможна информация, т.е. объектом является все то, что является элементом некоторого множества [17].

Объект, представляющий собой совокупность взаимосвязанных элементов (объектов), называется системой. Всякая система обладает свойством структурности, определяемой только взаимосвязями между элементами системы и сохраняющейся, возможно, с некоторой ненуле-

вой вероятностью, при замене элементов системы на их заменители. Любой объект является элементом одновременно для многих систем.

## 4.2. ОСНОВНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ АДАПТИВНЫХ СИСТЕМ УПРАВЛЕНИЯ

Одной из центральных задач теории автоматического управления является задача стабилизации. Она существенно усложняется при неполной информации и неопределенности в поведении объекта управления. При классическом решении задачи стабилизации возникает необходимость в подстройке устройств управления в процессе функционирования, что неприемлемо для динамических процессов. Выходом из сложившейся ситуации является создание таких систем, которые бы автоматически приспосабливались к возникающим условиям при управлении тем или иным технологическим процессом. Подобные системы получили название «адаптивные».

Один из первых примеров адаптивного управления привели Дрейпер и Ли [2]. Они рассмотрели систему управления, способную оптимизировать работу двигателя внутреннего сгорания, несмотря на неопределенность вида его рабочих характеристик. Такой тип системы управления, который автоматически ищет оптимальную рабочую точку, называется *оптимизирующей системой управления*, или (иногда) *системой, поддерживающей максимум*.

Другой важный класс адаптивных систем управления был введен Уитекером и др. [1]. Уитекер и его коллеги использовали модель поведения идеальной системы для получения сигналов ошибки, обусловленных отличием действительного поведения системы от требуемого; эти сигналы затем служили для модификации параметров управляющего устройства таким образом, чтобы, насколько это возможно, поддерживать идеальное поведение, несмотря на точно неизвестную и изменяющуюся динамику системы. Системы управления такого типа называются *адаптивными системами управления с эталонной моделью*. Уже с 1959 г. эти системы применялись в авиационных системах управления полетами [4, 10].

В 1960 г. Ли и Вандер Велде описали еще один тип адаптивных систем управления, работающих по принципу автоматической компенсации изменчивости параметров с помощью автоколебаний в замкнутом контуре управления. Система такого типа называется *самоосциллирующей*.



щей адаптивной системой. Несколько схожий подход к построению адаптивных систем был предложен академиками Б.Н. Петровым и С.В. Емельяновым [13]. Метод основан на инвариантности траектории системы по отношению к значениям ее параметров в случае, когда управляющее воздействие определяется переключающей функцией и реле. Поведение системы при этом характеризуется движением по переключающей поверхности и называется *скользящим режимом*. Адаптивные системы такого типа называются *системами с переменной структурой*.

Методы построения адаптивных систем, рассмотренных выше, ограничены рамками детерминированного подхода. Иначе говоря, в этих методах не используются вероятностные величины. Неопределенность в этом классе задач описывается детерминированным образом, например вводится внутренняя неопределенность значений параметров.

Теория адаптивного управления базируется на различных эвристических или полуэвристических концепциях. Она открывает другой путь решения задачи управления в условиях неопределенности. Трудность заключается в том, что эти системы очень разнообразны, неоднотипны и порой необязательны.

Без преувеличения можно сказать, что сегодня процессы возникновения регуляторных механизмов исследованы недостаточно. Во всех случаях появление нового метода синтеза скорее обязано изобретению, чем теории [7]. Поэтому современная теория управления движется в области поиска общих принципов синтеза, позволяющих в конкретных обстоятельствах как бы автоматически получать требуемый закон управления.

Приближением к решению этой проблематики может стать теория систем с переменной структурой [13]. Дело в том, что при создании управляющего устройства, базирующегося на принципах перестраиваемости, существует возможность наделить его динамической библиотекой алгоритмов управления. Здесь можно провести аналогию с классическим регулятором (см. рис. 4.1, а), который имеет фиксированную структуру, закладываемую проектировщиком на соответствующей стадии создания системы автоматического управления исходя из характеристик объекта, полученных тем или иным образом.

В устройствах управления с перестраиваемой структурой (см. рис. 4.1, б) имеется несколько фиксированных алгоритмов  $A_i$

( $i = 1, 2, \dots, n$ ), выбор которых происходит по определенному набору правил о состоянии объекта, при непосредственном функционировании всей автоматической системы регулирования.

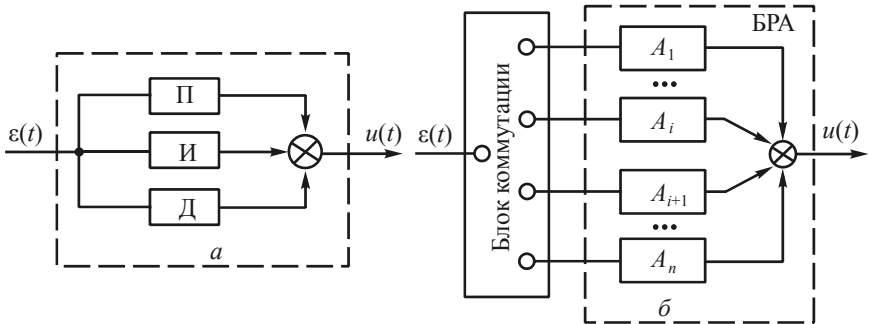


Рис. 4.1. Обобщенная внутренняя структура регулятора:  $a$  – ПИД;  $b$  – с перестраиваемой структурой

Как видно из рис. 4.1, управляющее устройство с перестраиваемой структурой может без труда перестроиться в классический регулятор при соответствующих наборах правил (например,  $A_1 \rightarrow k_n$ ,  $A_2 \rightarrow k_n / p$ ,  $A_3 \rightarrow k_d \cdot p$ , ошибка регулирования подается на первые три канала одновременно), но существует также и возможность получения новых свойств, не присущих ни одной из фиксированных структур.

Таким образом, возможность относительно простыми средствами улучшить динамику систем автоматического регулирования, используя принципы перестраиваемых структур, свидетельствует о растущем интересе к этому классу систем и является, по-видимому, очень перспективным направлением.

### 4.3. ТЕХНОЛОГИЧЕСКИЕ ПРОЦЕССЫ КАК ОБЪЕКТЫ УПРАВЛЕНИЯ

Технологические процессы представляют собой первичное звено создания материальных ценностей, они обеспечивают производство необходимой для существования общества продукции.

Понятие «технологический процесс» в дальнейшем будет использоваться нами в широком смысле и включать как непосредственные процессы производства определенного продукта, так и процессы обес-

печения нормального функционирования. Такое представление дает возможность с единой точки зрения подойти к решению задачи управления процессом производства определенного продукта в целом и задач управления отдельными технологическими агрегатами.

Это представление базируется на системном подходе, на комплексном решении задач обеспечения наилучшего в каком-то смысле их функционирования. Применение системного подхода необходимо, поскольку технологические процессы в настоящее время представляют собой сложные объекты управления с большим числом входных и выходных переменных. Сложные нелинейные взаимосвязи между переменными, распределенность в пространстве, их нестационарность, высокий уровень производственных шумов, недостаточная априорная информация о закономерностях хода процесса и другие причины создают значительные трудности в реализации адекватных моделей этих объектов; часто приходится непрерывно уточнять модель во время функционирования технологического объекта. Узкая приспособленность к выполнению определенных видов работ и их расчленение на ряд взаимосвязанных операций обычно обеспечивают высокую производительность технологических процессов [18].

Систему управления отдельными технологическими агрегатами обычно рассматривают как систему управления нижнего уровня в иерархической структуре управления технологическим процессом. Такая система должна обеспечивать оптимальное по заданному критерию функционирование технологического объекта. Критерий оптимальности для данного объекта задается системой управления технологического процесса в целом.

В результате осуществления технологического процесса над определенными свойствами входного сырья получают выходной продукт с другими свойствами. Для того чтобы можно было абстрагироваться от конкретных представлений входных и выходных свойств продукта, а также характеристик технологического процесса или параметров, характеризующих условия протекания технологического процесса, необходимо отвлечься от конкретной химической, физической, технологической природы этих характеристик. Это нужно сделать потому, что для построения общей теории управления технологическими процессами необходимо их представлять как объекты управления, и для этого должен быть использован язык теории управления [14].

Использование общих методов представления объектов в теории управления приводит в простейшем случае к следующему представлению технологического процесса в виде одномерного объекта (рис. 4.2), на входе которого действует переменная  $y(t)$ , характеризующая какое-либо свойство сырья, а на выходе имеем переменную  $x(t)$ , характеризующую какое-либо свойство готового продукта или полуфабриката.

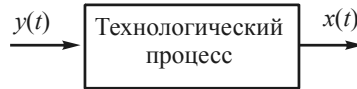


Рис. 4.2. Структурная схема одномерного технологического процесса

Современный технологический процесс представляется в виде многомерного объекта (рис. 4.3).



Рис. 4.3. Структурная схема многомерного технологического процесса с выделением параметров  $z$ , характеризующих ход процесса

На вход объекта действует векторная переменная  $Y(t)$  с составляющими  $y_1(t), \dots, y_n(t)$ . К таким переменным относятся все свойства (химический состав, размеры, механические свойства, скорость подачи, стоимость и т.д.) сырья или заготовок, используемых на данном технологическом объекте. Величины, характеризующие условия протекания технологического процесса, обозначены на рис. 4.3 через векторную функцию  $Z(t)$  с составляющими  $z_1(t), \dots, z_i(t)$ . К этим параметрам относятся характеристики хода технологического процесса: температура, давление, скорость, подача, число оборотов, производительность и т.д. Выходные величины описываются вектором

$$X(t) = \{x_1(t), \dots, x_m(t)\}.$$

Это выражение характеризует свойства полученного продукта или полуфабриката – химический состав, размеры, количество, стоимость и т.д.

Размерность векторов  $Y(t)$ ,  $X(t)$  и  $Z(t)$  очень велика, и учесть все их составляющие принципиально невозможно. Следовательно, при общем рассмотрении нет необходимости разделять переменные  $Y(t)$  и  $Z(t)$ , их обычно объединяют в группу входных переменных.

В итоге многомерный технологический процесс может быть представлен структурной схемой (рис. 4.4), каждая из независимых выходных переменных  $x_i(t)$ ,  $i = 1, \dots, m$ , определяется всеми входными переменными  $y_1(t), \dots, y_n(t)$ . Тогда для описания объекта необходимо иметь  $m$  операторов  $A_i$ :

$$x_i(t) = A_i\{y_1(t), \dots, y_n(t)\}.$$

Каждое выражение оператора зависит от выбранной модели, которая может быть представлена в виде системы обыкновенных дифференциальных уравнений, системы уравнений в частных производных, интегральных или интегро-дифференциальных уравнений и др. Выбор модели в основном определяется спецификой решаемой задачи.



Рис. 4.4. Структурная схема многомерного технологического процесса

Классификация объектов обычно производится в зависимости от способов представления оператора и входных и выходных переменных. Входные и выходные переменные могут быть как детерминированными, так и случайными.

Детерминированной моделью объекта называется модель, оператор которой детерминирован, т.е. устанавливает однозначное соответствие между входными и выходной переменными.

Стохастической моделью объекта называется модель, соответствующий оператор которой является случайным. Выходная величина стохастического объекта всегда случайна, даже при наличии детерминированных входных переменных.

Всякий процесс управления может включать реализацию следующих составляющих.

1. Получение информации о состоянии объекта путем контроля выходной переменной и определение заданных требований, предъявляемых к ее значениям.

2. Обработка информации о состоянии выходного параметра и сравнение полученного результата с заданием.

3. Принятие решений о модификации состояния объекта и выработка алгоритма управления УУ.

Под алгоритмом функционирования УУ понимается точное предписание порядка выполнения УУ операций по управлению объектом управления (ОУ).

Постоянно взаимодействуя с ОУ (рис. 4.5), УУ вырабатывает последовательность воздействий  $q_i \in \{q_1, \dots, q_k\}$  на ОУ в зависимости как от сигналов  $p_j \in \{p_1, \dots, p_m\}$ , поступающих в УУ от ОУ и характеризующих его состояние, так и от внешних сигналов  $r_\xi \in \{r_1, \dots, r_l\}$ , поступающих в УУ из других источников информации.

4. Обработка исполнительными органами воздействия на объект согласно выбранным законам управления.

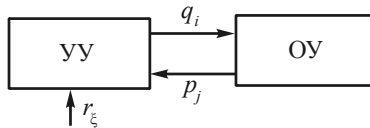


Рис. 4.5. Структурная схема взаимодействия ОУ и УУ

Реальные технологические процессы как объекты управления в большинстве нестационарны, нелинейны, многомерны, обладают многими внутренними обратными связями, у которых невелика или отсутствует вовсе априорная информация о форме и степени взаимосвязи между переменными в динамике в реальных условиях эксплуатации. Данный факт значительно усложняет получение адекватного (изоморфного) математического описания технологических процессов как объектов управления.

#### 4.4. МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ И ПОНЯТИЕ ФАЗОВОГО ПРОСТРАНСТВА

Совокупность физических устройств, характеризующихся тем, что протекающие в них процессы однозначно определяются приложенными

внешними воздействиями, взаимными связями между устройствами и начальными их сочетаниями (начальными условиями), называется динамической системой.

Это определение говорит о том, что мы имеем дело с объектами или физическими процессами, описываемыми системой дифференциальных уравнений следующего вида:

$$\begin{cases} \frac{dx_1}{dt} = X_1(t, x_1, x_2, \dots, x_n), \\ \dots \dots \dots \\ \frac{dx_n}{dt} = X_n(t, x_1, x_2, \dots, x_n), \end{cases} \quad (4.1)$$

где все функции  $X_i$  ( $i = 1, 2, \dots, n$ ) в общем случае нелинейные, непрерывные, ограничены и имеют ограниченные частные производные по  $x_i$  в рассматриваемой замкнутой области переменных  $t, x_1, \dots, x_n$  [11]. Здесь время  $t$  является независимой переменной, а  $x_1, x_2, \dots, x_n$  – выходными переменными системы, подлежащими определению.

В общем случае система уравнений является нелинейной с сосредоточенными или распределенными параметрами, изменяющимися в процессе случайным образом непрерывно или дискретно.

Среди всего многообразия динамических систем будем рассматривать только линейные системы с сосредоточенными параметрами. Такие системы описываются в общем виде линейными дифференциальными уравнениями с постоянными коэффициентами:

$$\begin{cases} \frac{dx_1}{dt} = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + y_1(t); \\ \frac{dx_2}{dt} = a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + y_2(t); \\ \dots \dots \dots \\ \frac{dx_n}{dt} = a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n + y_n(t). \end{cases} \quad (4.2)$$

Здесь функции  $y_1(t), \dots, y_n(t)$  описывают заданные возмущения (входные сигналы), приложенные к различным входам системы, а коэффициенты  $a_{ij}$  определяют физические параметры системы или их комбинации.

В случае, если система имеет один входной сигнал и один выходной, она приводится к одному уравнению  $n$ -го порядка:

$$\begin{aligned} a_n \frac{d^n x_{\text{ВЫХ}}}{dt^n} + a_{n-1} \frac{d^{n-1} x_{\text{ВЫХ}}}{dt^{n-1}} + \dots + a_1 \frac{dx_{\text{ВЫХ}}}{dt} + a_0 x_{\text{ВЫХ}} &= \\ &= b_m \frac{d^m x_{\text{ВХ}}}{dt^m} + b_{m-1} \frac{d^{m-1} x_{\text{ВХ}}}{dt^{m-1}} + \dots + b_1 \frac{dx_{\text{ВХ}}}{dt} + b_0 x_{\text{ВХ}}, \end{aligned} \quad (4.3)$$

или в операторной форме

$$W(p) = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0}, \quad m \leq n. \quad (4.4)$$

В работе будем рассматривать класс объектов, описываемых передаточными функциями вида

$$W(p) = \frac{k \exp(-\tau p)}{T_2^2 p^2 + T_1 p + 1}, \quad T_1 > 2T_2, \quad (4.5)$$

поскольку большое количество технологических объектов либо описывается передаточными функциями первого или второго порядка, с запаздыванием или без, либо их динамические характеристики могут быть аппроксимированы этими функциями [5]. Вытекает это из того, что функцию (4.5) можно разделить на две части:

$$W(p) = W_1(p)W_2(p) = W_1(p) \exp(-\tau p), \quad (4.6)$$

а передаточную функцию  $W_2(p) = \exp(-\tau p)$ , в свою очередь, можно разложить в ряд Паде четвертого порядка для представления в виде отношения полиномов:

$$W_2(p) = \frac{\tau^4 p^4 - 20\tau^3 p^3 + 180\tau^2 p^2 - 840\tau p + 1680}{\tau^4 p^4 + 20\tau^3 p^3 + 180\tau^2 p^2 + 840\tau p + 1680}. \quad (4.7)$$

Следовательно, при наличии звена запаздывания порядок передаточной функции возрастает на  $k$  раз, где  $k$  – порядок разложения экспоненциальной функции в ряд Паде. На рис. 4.6 приведены пример переходной характеристики объекта с запаздыванием и



аппроксимирующие кривые при замене передаточной функции транспортного запаздывания рядом Паде  $k$ -го порядка [15].

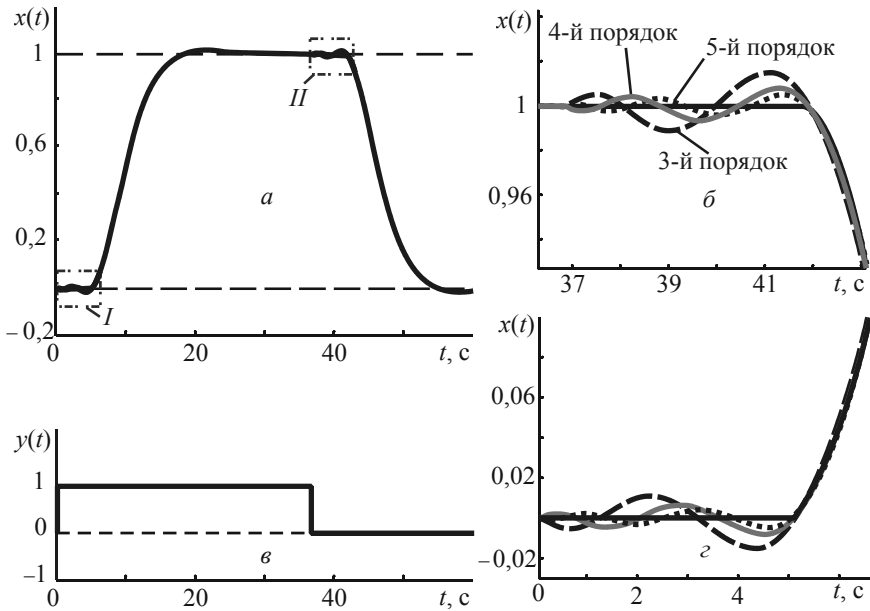


Рис. 4.6. Пример замены передаточной функции звена транспортного запаздывания рядом Паде:  $a$  – выходная величина объекта;  $б$  – участок  $\Pi$  из « $a$ » в увеличенном виде;  $в$  – входная величина объекта;  $г$  – участок  $I$  из « $a$ » в увеличенном виде

Как уже отмечалось, функцией (4.5) описывается большое количество всевозможных технологических объектов: технологические резервуары, подключенные через сопротивление к магистральному трубопроводу под давлением, различные технологические параметры для доменной печи. Примерами двуместных объектов ( $T_2 \neq 0$ ) могут служить две ступени пароперегревателей барабанного парового котла, а также многие другие объекты [8, 12].

Таким образом, *выбранная передаточная функция* описывает поведение достаточно большого класса технологических объектов управления.

Решение системы уравнений (4.1) при заданных начальных условиях дает возможность для каждой координаты состояния определить описывающую ее функцию времени  $x_i(t)$ , а процесс, наблюдаемый в системе, изобразить в виде линии в  $n$ -мерном пространстве координат состояния, для которой время  $t$  является параметром точки кривой, соответствующей уравнению  $x(t)$ , называемой *изображающей точкой*. Геометрическое место изображающих точек при изменении времени называется *траекторией процесса в пространстве состояний*.

При рассмотрении *автономных систем*, не подверженных воздействию от времени внешним воздействиям, пространство состояний обычно называют *фазовым пространством*, а траекторию изображающей точки – *фазовой траекторией*.

*Фазовым портретом* системы называется совокупность фазовых траекторий, полученных при различных начальных условиях.

Приведем иллюстрации фазовых портретов линейных систем, описываемых уравнениями второго порядка с постоянными коэффициентами. Предположим, что дифференциальное уравнение

$$\frac{d^2x(t)}{dt^2} - a_1 \frac{dx(t)}{dt} + a_0x(t) = b_0y(t)$$

описывает поведение некоторой динамической системы. Тогда характеристическое уравнение системы и его решение примут вид

$$p^2 - a_1p + a_0 = 0, \quad p_{1,2} = \frac{a_1 \pm \sqrt{a_1^2 - 4a_0}}{2}.$$

Устойчивость и качество системы в *малом* зависят от корней характеристического уравнения  $p_{1,2}$ , которые определяются его коэффициентами  $a_1$  и  $a_0$ .

Приведем координатную плоскость, выбрав  $a_1$  и  $a_0$  в качестве ее координат. Разобьем эту плоскость на области, соответствующие качественно различному поведению рассматриваемой системы (рис. 4.7):

– о б л а с т ь 1, в которой процессы устойчивы и апериодические; точка равновесия типа «устойчивый узел»;

– о б л а с т ь 2, в которой процессы устойчивы и колебательные; точка равновесия типа «устойчивый фокус»;

- о б л а с т ь 3, в которой процессы неустойчивые и колебательные; точка равновесия типа «неустойчивый фокус»;
- о б л а с т ь 4, в которой процессы неустойчивы и аperiodические; точка равновесия типа «неустойчивый узел»;
- о б л а с т ь 5, в которой процессы неустойчивы; точка равновесия типа «седло».

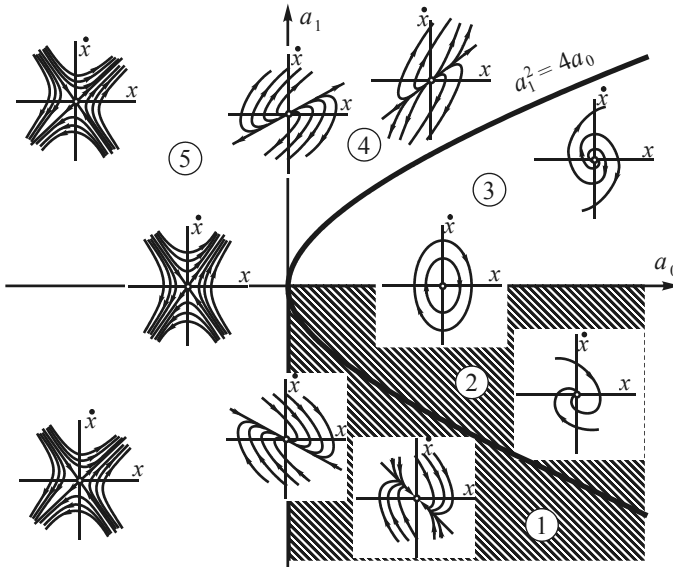


Рис. 4.7. Плоскость фазовых портретов

На границе областей 2 и 3 в системе возникают незатухающие колебания, амплитуда которых зависит от начальных условий; точка равновесия типа «центр».

Задача управления с точки зрения фазовых представлений движения сводится либо к перемещению изображающей точки из некоторой допустимой области начальных условий в заданную точку (или область) фазового пространства, либо к обеспечению движения изображающей точки по некоторой заданной траектории.

Вид траектории, по которой осуществляется переход системы из одного состояния в другое, позволяет судить о динамических свойствах системы (время переходного процесса, перерегулирования, колебатель-

ность, точность и т.п.). Синтез методом фазового пространства предполагает выбор такого управления, которое обеспечивает этот переход по желаемым траекториям.

#### **4.5. КЛАССИФИКАЦИЯ ПРОМЫШЛЕННЫХ ОБЪЕКТОВ УПРАВЛЕНИЯ**

По характеру происхождения технологических процессов объекты управления можно разделить на циклические, непрерывно-циклические и непрерывные.

В зависимости от установившегося значения выходной величины объекта при действии на его вход ступенчатого сигнала выделяют объекты с самовыравниванием и без самовыравнивания.

Объекты можно разделить в зависимости от количества входных и выходных величин и их взаимосвязи на одномерные (один вход и один выход) и многомерные. Последние могут быть многосвязными, когда наблюдается взаимное влияние каналов регулирования друг на друга, либо несвязными, когда взаимосвязь между их каналами мала.

Объекты управления могут делиться на линейные и нелинейные в зависимости от вида статической характеристики, определяющей связь между установившимися значениями входа и выхода объекта. В нелинейных статическая характеристика может быть гладкой, линеаризуемой в окрестности заданной точки либо носить существенно нелинейный характер. Большая часть систем автоматического регулирования относится к классу систем автоматической стабилизации режима работы объекта относительно его рабочей точки (относительно номинального режима работы). В этом случае в процессе работы отклонения переменных относительно рабочей точки будут малы, что позволяет использовать линейные модели объекта управления. Однако при смене рабочей точки происходит изменение коэффициента усиления объекта, что будет негативно влиять на динамику замкнутой системы.

Определение полной статической характеристики объекта для системы автоматической стабилизации не обязательно. Достаточно знать лишь динамический коэффициент усиления в окрестности рабочей точки. В то же время на некоторых объектах управления необходимо зна-

ние всей статической характеристики процесса. Если она носит нелинейный характер, то с целью стабилизации общего коэффициента усиления системы в замкнутый контур включают дополнительную нелинейность, обратную статической характеристике объекта. На практике такой подход реализуется путем использования регулирующих клапанов с различными видами расходной характеристики.

Реальные объекты управления занимают в пространстве какой-либо объем, поэтому регулируемая величина зависит не только от времени, но и от текущих координат точки измерения. Поэтому полное описание объекта управления будет состоять из системы дифференциальных уравнений с частными производными. При использовании точечного метода измерения одним датчиком система дифференциальных уравнений с частными производными переходит в систему уравнений с обычными производными. Это существенно упрощает построение математической модели объекта, позволяя определить его передаточную функцию. Однако при наличии множества датчиков, распределенных, например, по длине объекта, может возникнуть необходимость использования множества управляющих сигналов (распределенное управление).

Объекты могут быть как стационарными, так и нестационарными. В нестационарных объектах параметры изменяются с течением времени (дрейфуют). Примерами таких объектов могут быть химический реактор с катализатором, активность которого падает с течением времени, или аэрокосмический аппарат, масса которого по мере выгорания топлива уменьшается. Такие явления должны учитываться при проектировании соответствующих систем управления.

Ключевой целью построения математической модели объекта управления является определение структуры объекта, его статических и динамических характеристик. Особенно важно определение структуры для многомерных и многосвязных объектов управления. В то же время для локальных объектов управления определение структуры может быть сведено к определению порядка дифференциального уравнения описываемого объекта. Кроме того, оцениваются входные сигналы и возмущения, действующие на объект (их статистические характеристики, точки приложения, максимальные амплитуды). Значение этих ха-

рактических позволяет выбрать структуру регулятора и рассчитать параметры его настройки, ориентируясь также на критерий качества работы системы.

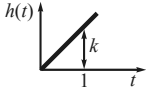

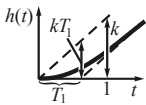

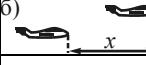
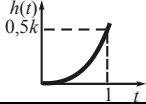

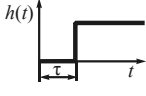
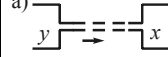
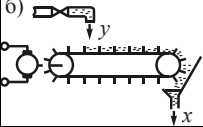
Передаточные функции типовых управляемых объектов различной физической природы приведены в табл. 4.1. Разумеется, эти передаточные функции соответствуют предельно упрощенному описанию управляемых объектов [16].

Т а б л и ц а 4.1

Передаточные функции типовых объектов управления

№ процесса	Передаточная функция $W(p)$	Переходная характеристика $h(t)$	Примеры	
1	2	3	4	
1	$k$		а)  б) 	а) Ток и напряжение в электрической цепи б) Давление и поток жидкости в трубопроводе
2	$\frac{k}{T_1 p + 1}$		а)  б)  в) 	а) Давление и поток в газопроводе б) Число оборотов электродвигателя в) Уровень жидкости в резервуаре
3	$\frac{k}{T_2 p^2 + T_1 p + 1}$		а)  б)  в) 	а) Температура печи б) Напряжение генератора в) Число оборотов электродвигателя
4			г)  д) 	г) Число оборотов крупного электродвигателя д) Курс корабля

Продолжение табл. 4.1

1	2	3	4
5	$\frac{k}{p}$		 Уровень жидкости в барабане котлоагрегата
6	$\frac{k}{p(T_1 p + 1)}$		а)  б)  а) Угол поворота электропривода б) Взаимное расположение самолетов
7	$\frac{k}{p^2}$		 Траектория ракеты, спутника
8	$\exp(-\tau p)$		а)  б)  а) Участок трубопровода, импульсная линия б) Ковшовый транспортер

#### 4.6. ИДЕНТИФИКАЦИЯ ОБЪЕКТА УПРАВЛЕНИЯ

При выборе типа регулятора и параметров его настроек необходимо знать динамические характеристики объектов регулирования.

Для уже действующих промышленных объектов наиболее достоверные данные могут быть получены экспериментальным путем. Часто промышленные объекты являются сложными физическими системами, описываемыми нелинейными дифференциальными уравнениями высокого порядка. Аналитическое определение их динамических свойств и линеаризация их характеристик по конструктивным и физическим данным являются чрезвычайно трудоемкой и сложной задачей. Такая оценка динамических свойств целесообразна лишь для проектируемых объектов управления.

Приведем методику экспериментального определения динамических характеристик действующих объектов регулирования. Она применима для объектов, представляющих собой устойчивые или нейтральные физические системы. Подобные системы в структурных схемах

можно представить в виде последовательно включенных типовых звеньев. Промышленные объекты, включающие в себя колебательные звенья, встречаются сравнительно редко. Как правило, при проектировании подбирают параметры объектов так, чтобы они были хорошо демпфированы. Благодаря этому объект рассматривают как последовательное соединение звеньев 1-го порядка (апериодических и интегрального) и звена с чистым запаздыванием [9].

Известно несколько методов определения динамических характеристик регулируемых объектов.

1. Методы, базирующиеся на искусственном воздействии на систему непериодическим сигналом, мощность которого велика по сравнению с уровнем помех в системе. В качестве воздействия выбирается обычно скачкообразное изменение регулирующего воздействия, и в результате определяются временные характеристики.

2. Методы, основанные на искусственном воздействии на систему периодическими сигналами разной частоты, амплитуда которых велика по сравнению с уровнем помех в системе. В результате определяются частотные характеристики.

3. Методы, базирующиеся на искусственном воздействии на систему синусоидальными сигналами, соизмеримыми с помехами в системе. В результате также определяются частотные характеристики.

4. Методы, не требующие искусственных воздействий, использующие возмущения, которые имеются в процессе нормальной эксплуатации.

Временной характеристикой объекта называют зависимость регулируемой величины от времени при скачкообразном изменении регулирующего воздействия. Применительно к объектам регулирования эту зависимость также называют кривой разгона.

На практике в большинстве случаев приходится сталкиваться с невозможностью построить кривую разгона полностью по причине технологических ограничений, налагаемых на максимальные отклонения регулируемой величины. Вследствие этого ограничиваются так называемой импульсной кривой разгона, которая представляет собой зависимость регулируемой величины от времени при скачкообразной подаче и последующем снятии регулирующего воздействия.

Из-за инерционности и ограниченности скорости перемещения регулирующих органов или других устройств, как правило строят,



«искаженную» кривую разгона, или «искаженную» импульсную кривую разгона.

**Схема опыта.** В многосвязном объекте выделяются регулируемые величины и регулирующие воздействия. Затем поочередно определяется связь каждого регулирующего воздействия с регулируемыми величинами. Причем другие регулирующие воздействия и регулируемые величины поддерживаются, где это возможно, с помощью регуляторов или вручную на заданном уровне, чтобы они не влияли на результаты эксперимента. Нагрузку при этом необходимо стабилизировать. Например, в системе, изображенной на рис. 4.8, регулируемые параметры  $y_1$  и  $y_4$  поддерживаются постоянными с помощью регуляторов  $Reg_1$  и  $Reg_4$ , регулирующее воздействие  $\mu_3$  – вручную для стабилизации режима во время снятия кривых разгона величин  $y_2$  и  $y_3$  при скачкообразном воздействии по  $\mu_2$  (объект во время снятия кривых разгона разомкнут по величинам  $y_2$  и  $y_3$ ). Этап такой работы должен проводиться совместно с технологами данного производства и на базе приобретенного опыта автоматизации аналогичных объектов.

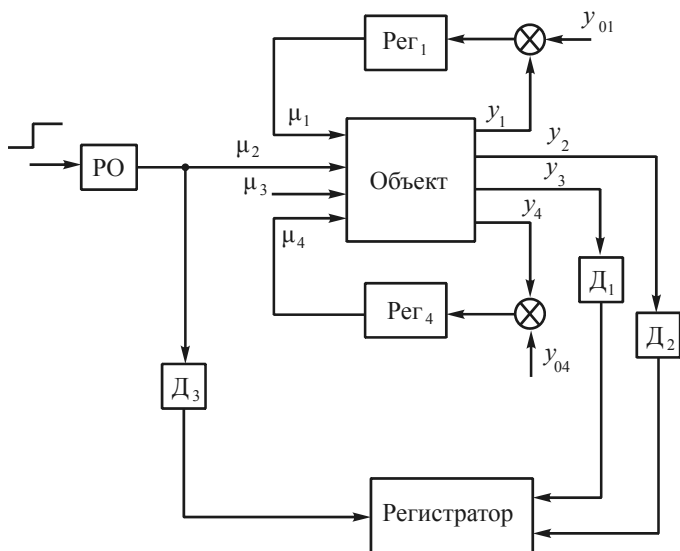


Рис. 4.8. Схема опыта снятия кривых разгона с объекта регулирования:

$Reg_1, Reg_4$  – регуляторы;  $D_1, D_2, D_3$  – датчики;

РО – регулирующий орган

При таких исследованиях очень важно правильно выбрать аппаратуру изменения регулируемых величин и регулирующих воздействий. Процесс измерения и регистрации указанных параметров должен производиться с помощью чувствительной и быстродействующей аппаратуры (постоянные времени аппаратуры должны быть малы по сравнению с постоянными времени объекта). Об этом можно судить либо по предварительным расчетам, либо по результатам пробных опытов, имея данные об аппаратуре.

В процессе подбора датчиков нужно отдавать предпочтение малоинерционным. Если это невозможно, то при определении характеристик следует использовать датчики и регистрирующие приборы, которые в дальнейшем войдут в качестве элементов в автоматическую систему регулирования (АСР). В этом случае целесообразно снимать совместные кривые разгона объекта, датчика и регистрирующего прибора.

**Выбор регулирующего воздействия.** В зависимости от конкретных ситуаций в качестве регулирующего воздействия может рассматриваться как непосредственно регулирующее воздействие (расход греющей и охлаждающей среды, количество топлива, жидкости и т.п.), так и положение регулирующего органа или сигнал исполнительному механизму. Последнее целесообразно лишь в случае, когда регулирующий орган или исполнительный механизм будут использованы в АСР данного объекта.

В случае наличия информации о том, что объект работает в широком диапазоне изменений регулирующих воздействий, необходимо снять характеристики при нескольких начальных значениях этих воздействий, так как характеристики могут оказаться нелинейными. Во многих случаях линеаризация системы во всем диапазоне изменений регулирующего воздействия невозможна. При изменении последнего может оказаться, что меняются не только параметры объекта, но и его структура.

Такое может происходить в АСР, программа которых (задаваемое значение регулируемой величины) и действующие на объект внешние возмущения меняются в широких пределах.

Если это возможно по технологическим требованиям и осуществимо аппаратурно, то снимается кривая разгона, т.е. регулирующее воздействие изменяется скачкообразно (рис. 4.9, а). Если же по технологическим требованиям регулируемая величина в процессе исследования

объекта может меняться только в узких пределах, то снимается импульсная кривая разгона. Форма изменения регулирующего воздействия для этого случая приведена на рис. 4.9, б, где  $t_{и}$  – длительность воздействия. В случаях, когда аппаратно невозможно осуществить практически мгновенное изменение регулирующего воздействия, снимаются «искаженные» кривые разгона. Формы регулирующих воздействий для этих случаев приведены на рис. 4.9, в и з (в – для случая, когда устройство, изменяющее регулирующее воздействие, является инерционным звеном первого порядка, з – для случая, когда ограничена скорость изменения регулирующего воздействия).

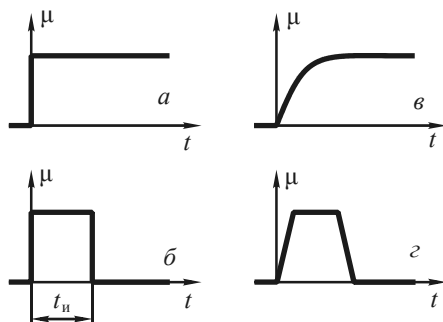


Рис. 4.9. Форма регулирующего воздействия

При проведении опыта должны учитываться максимальное значение возможных эксплуатационных возмущений, нелинейность характеристик и уровень помех. Обычно выбирают величину возмущения, равную 5–20% максимального значения (увеличение возмущения может привести к неточным результатам из-за нелинейности характеристик, уменьшение его при наличии помех затруднит обработку результатов).

**Порядок проведения опыта.** В начале необходимо достичь установившегося состояния объекта по изучаемым регулируемым величинам. В объекте значения регулируемых величин должны поддерживаться постоянными при выбранном начальном значении регулирующего воздействия. Установившееся состояние может быть получено стабилизацией внешних возмущений, а также других регулируемых величин и регулирующих воздействий, влияющих на исследуемые. Установившийся режим должен фиксироваться некоторое время в зависимости от

динамических свойств объектов. Чем больше время переходного процесса, которое обычно ориентировочно известно из пробных испытаний, тем больше время фиксирования.

Затем необходимо проверить стабильность нового установившегося состояния, т.е. состояния, которое будет устанавливаться после изменения регулирующего воздействия. Новый установившийся режим нужно стабилизировать аналогично ранее рассмотренному.

Далее можно перейти к снятию характеристик в объекте. При одном из выбранных начальных значений подается 2–3 раза выбранное воздействие и регистрируются регулируемые величины и регулирующее воздействие. Опыт повторяется для другого выбранного начального значения и т.д.

**Определение параметров переходного процесса.** Для процесса № 2 (см. табл. 4.1) постоянная времени определяется по формуле

$$T = \frac{t_{0,7}}{1,2},$$

где  $t_{0,7}$  – время, при котором ордината  $h(t)$  достигает значения  $0,7k$ .

Для процесса № 3 постоянные времени  $T_1$  и  $T_2$  можно определить с помощью диаграммы Ольденбурга и Сарториуса [3].

Порядок работы с диаграммой следующий.

На переходной характеристике провести касательную через точку перегиба (определение точки перегиба см. ниже) и определить значение времени  $T_C$  и  $T_A$ , как показано на рис. 4.10.

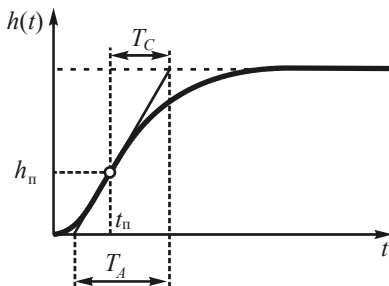


Рис. 4.10. Переходная характеристика системы

На диаграмме (рис. 4.11) отношение  $T_C/T_A$  ( $0,73 \leq T_C/T_A \leq 1$ ) используется как значение точки пересечения прямой с каждой из осей. Прямая линия, проведенная таким образом, пересекает кривую на диаграмме в двух точках, каждая из которых позволяет определить отношение  $T_1/T_A$  и  $T_2/T_A$ . По этим отношениям легко определить  $T_1$  и  $T_2$ .

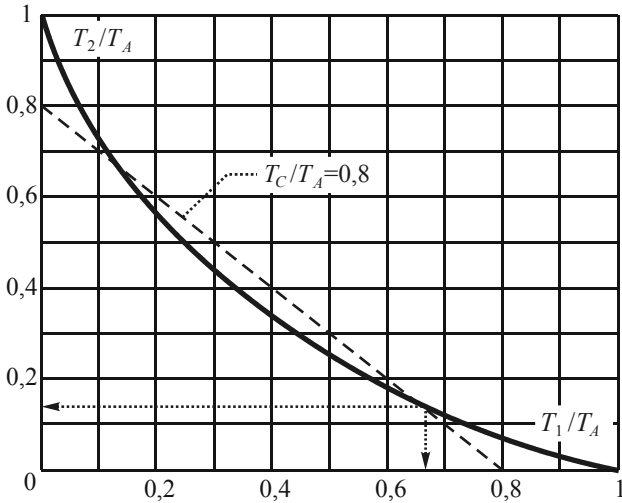


Рис. 4.11. Диаграмма Ольденбурга и Сарториуса

Под *точкой перегиба* понимается такая точка  $h(t)$ , в которой производная  $dh/dt$  имеет максимальное значение. Так как переходные функции многих промышленных объектов не имеют явно выраженной точки перегиба, то определение ее координат надо осуществлять следующим образом. В средней (наиболее быстро изменяющейся) части графика  $h(t)$  берется несколько координат  $h(t_i) = h_i$ ,  $i = 0, 1, 2, \dots, m$ ;  $m$  обычно не более 6–7;  $t_i - t_{i-1} = \Delta t = \text{const}$ , и вычисляются первые разности  $\Delta h_i = h_i - h_{i-1}$ ,  $i = 0, 1, 2, \dots, m-1$ . Далее находят максимальную величину  $\Delta h_i$ , соответствующее ей значение времени  $t_{\Pi} = t_i - 0,5\Delta t$  и, наконец, ордината  $h_{\Pi}$ .

Для процесса № 4 постоянные времени  $T_1$  и  $T_2$  определяются из формул

$$T_1 = \frac{2\alpha}{\alpha^2 + \beta^2}, \quad T_2 = \frac{1}{\alpha^2 + \beta^2},$$

где  $\alpha = -\frac{\ln(A_2/A_1)}{T_k}$ ;  $\beta = 2\pi/T_k$ ;  $A_1, A_2, T_k$  – характеристики переходного процесса, определяемые непосредственно из графика (рис. 4.12).

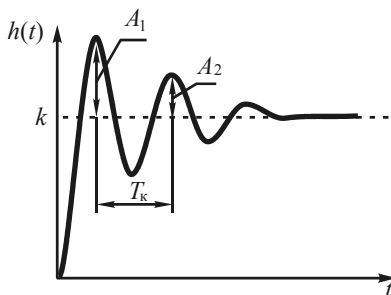


Рис. 4.12. Переходная характеристика системы

Для процесса № 5 коэффициент передачи объекта определяется из формулы

$$k = \operatorname{tg} \alpha,$$

где  $\alpha$  – угол наклона переходной характеристики от оси абсцисс.

#### 4.7. ПОНЯТИЕ СИСТЕМ С ПЕРЕМЕННОЙ СТРУКТУРОЙ

Значительные дополнительные возможности улучшения процессов регулирования дает нелинейное управление работой объекта путем изменения структуры управляющего устройства в зависимости от размеров и знаков входных величин, поступающих в управляющее устройство от измерительного устройства.

Могут использоваться комбинации линейных законов регулирования (их достоинства, обходя недостатки). Например, если известно, что при одном законе регулирования получается быстрое начальное изменение регулируемой величины, но с большим перерегулированием, а

при другом – медленное изменение, но плавный подход к новому установившемуся режиму, то можно, включив сначала первый закон, переключить затем систему на второй закон в некоторой точке  $A$ , когда отклонение  $y$  достигнет определенного значения  $y_A$ . В результате процесс регулирования приобретет форму кривой (рис. 4.13), объединяющей оба качества – быстроту и плавность процесса. Используя данный подход, мы получаем процесс регулирования без характерных для ПИ- и ПИД-регуляторов колебаний и перерегулирования при малом времени регулирования.

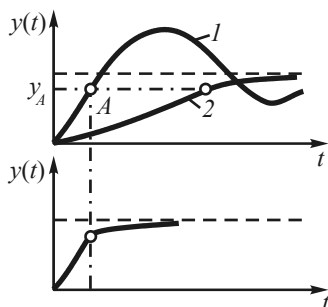


Рис. 4.13. Принцип работы системы с переменной структурой

Подобные системы относятся к классу нелинейных, даже если все звенья системы линейны. Происходит это за счет того, что в процессе работы происходят автоматические переключения между звеньями. Такую работу можно сравнить с тем, как получается нелинейная статическая характеристика из отрезков прямых линий. Но в данном случае имеет место нелинейная динамическая характеристика, составляемая из последовательности разных линейных дифференциальных уравнений, соответствующих первому и второму законам регулирования.

Синтез структуры системы осуществляется исходя из предположения, что совокупность функциональных элементов и характер связей между ними остаются раз и навсегда неизменными. Но такой взгляд на проектирование систем управления, вообще говоря, не является единственным. При построении управляющего устройства

можно заранее ввести в структуру системы такие функциональные элементы, которые во время протекания процесса управления скачкообразно меняют знак обратной связи, коэффициенты воздействия по различным координатам и т.д., т.е. изменяют структуру системы. Следовательно, в зависимости от выбранного алгоритма и имеющейся информации система будет обладать той или иной структурой. В такой системе удастся совмещать полезные свойства каждой из имеющейся совокупности структур, а быть может, и получить какие-либо новые свойства, не присущие любой из них. Такой подход позволяет существенно повысить качество управления.

В дальнейшем под системами с переменной структурой будем понимать системы, в которых связи между функциональными элементами меняются тем или иным образом в зависимости от состояния системы [6].

На рис. 4.14 представлена достаточно общая схема системы с переменной структурой.

В зависимости от того, какие координаты системы и внешние воздействия доступны для измерения, на вход УУ может поступать информация о величине ошибки  $\varepsilon$ , выходной координате исполнительного устройства  $\mu$ , задающем и возмущающем воздействиях  $s(t)$  и  $f(t)$ , о регулируемой величине  $y$ , о промежуточных координатах объекта  $y_1, \dots, y_i$  и исполнительного устройства  $\mu_1, \dots, \mu_k$ . Управляющее устройство содержит ряд функциональных звеньев, причем оператор каждого из них обозначен на рис. 4.14 аббревиатурой ФЗ с соответствующими индексами.

Любая из входных величин управляющего устройства может поступать на входы только «своих» звеньев. Управляющее воздействие  $u$  является суммой выходных величин всех звеньев. В УУ содержатся ключевые элементы (КЭ). Любой КЭ замыкает один из предусмотренных каналов передачи информации. Разнообразные сочетания положений контактов КЭ определяют совокупность имеющихся в распоряжении фиксированных структур. Блок управления структурой (БУС) на основе



анализа всей поступающей на УУ информации дает команду на изменение структуры системы. Вопрос синтеза такой системы сводится к выбору операторов всех звеньев и логических законов, в соответствии с которыми изменяется структура системы.

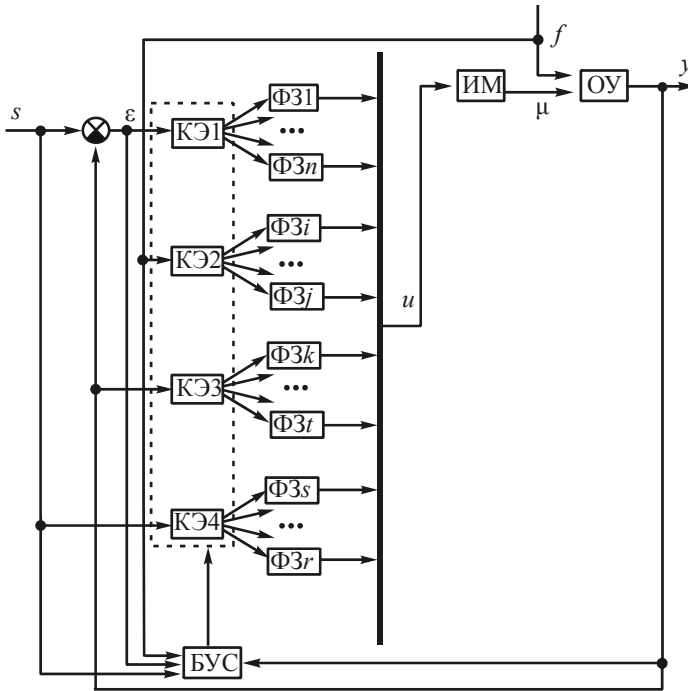


Рис. 4.14. Общая структурная схема системы с переменной структурой

На рис. 4.15 представлена обобщенная схема автоматического регулирования с переменной структурой. В нее входит блок измерения, или измерительное устройство (ИзУ), которое производит непрерывную оценку регулируемой величины и вычисляет величину сигнала рассогласования, который подается на блок формирователя функции переключения, осуществляющий анализ и переключение сигнала отклонения регулируемой величины  $\varepsilon(t)$  на необходимый закон регулирования (блок коммутируемых коэффициентов).

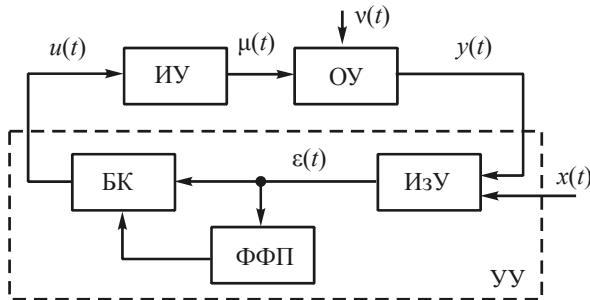


Рис. 4.15. Схема автоматического регулирования с переменной структурой: БК – блок коммутируемых коэффициентов; ИзУ – измерительное устройство; ФФП – блок формирования функции переключения; ИУ – исполнительное устройство

#### 4.8. МЕТОД ФАЗОВОГО ПРОСТРАНСТВА

Пусть линейная система описывается дифференциальным уравнением

$$\frac{d^n x}{dt^n} + a_{n-1} \frac{d^{n-1} x}{dt^{n-1}} + \dots + a_1 \frac{dx}{dt} + a_0 x = 0,$$

где  $a_0, a_1, \dots, a_{n-1}$  – постоянные величины.

Если  $\lambda_1, \lambda_2, \dots, \lambda_n$  – корни характеристического уравнения (предполагается, что кратные корни отсутствуют), то

$$x^{(j)} = \sum_{j=1}^n A_j \lambda_j^{i-1} \exp(\lambda_j t), \quad i = 0, \dots, n-1,$$

где  $A_j$  – постоянные интегрирования, зависящие от начальных условий;  $x^{(j)}$  – координаты фазового пространства  $X$ .

Очевидно, что линейная структура устойчива или любая траектория в фазовом пространстве  $X$  стягивается к началу координат, если  $\operatorname{Re} \lambda_j < 0$  ( $j = 1, \dots, n$ ). В рассмотренном случае это области 1 и 2 рис. 4.7.

Пусть имеются две, даже неустойчивые, линейные структуры, но в фазовом пространстве у одной из них существует гиперплоскость с ус-

тойчивым вырожденным движением. Тогда необходимо выбрать такую последовательность изменения этих структур, чтобы, во-первых, любая траектория в пространстве  $X$  пересекла эту гиперплоскость и, во-вторых, после момента попадания изображающей точки на нее в системе имела место структура с устойчивым вырожденным движением. Синтезированная таким образом система будет устойчивой для любых начальных условий.

Допустим, что первой структуре соответствует фазовый портрет на рис. 4.16, *а*, а второй – на рис. 4.16, *б*.

Появляется задача, сводящаяся к выбору такой последовательности изменения структур, чтобы любое движение системы было устойчиво. Разделим фазовую плоскость  $(x, x')$  на два листа, границами которых являются прямая  $S$  и ось  $x'$  (рис. 4.16, *в*). Если состояние системы таково, что изображающая точка находится на листе I, то ее движение должно происходить по раскручивающимся спиральям (система должна иметь первую структуру). На листе II изображающая точка должна двигаться по кривым гиперболического типа (система должна иметь вторую структуру). Из рис. 4.16, *в* видно, что изображающая точка всегда попадает на прямую  $S$ , которая является устойчивой траекторией для второй структуры. Подобный подход позволяет синтезировать устойчивую систему и отказаться от требования расположения корней в левой комплексной полуплоскости для каждой из имеющихся структур.

Иной способ, который может быть положен в основу построения системы с переменной структурой, целесообразно использовать в случае, если фазовое пространство для каждой из фиксированных неустойчивых структур не содержит гиперплоскостей с устойчивыми вырожденными движениями. В результате «сшивания» в определенной последовательности участков неустойчивых траекторий удастся получить устойчивое движение для любых начальных условий. Рассмотрим пример, когда в нашем распоряжении имеются две линейные структуры, находящиеся на границе устойчивости. Фазовые портреты каждой из структур представлены на рис. 4.17.

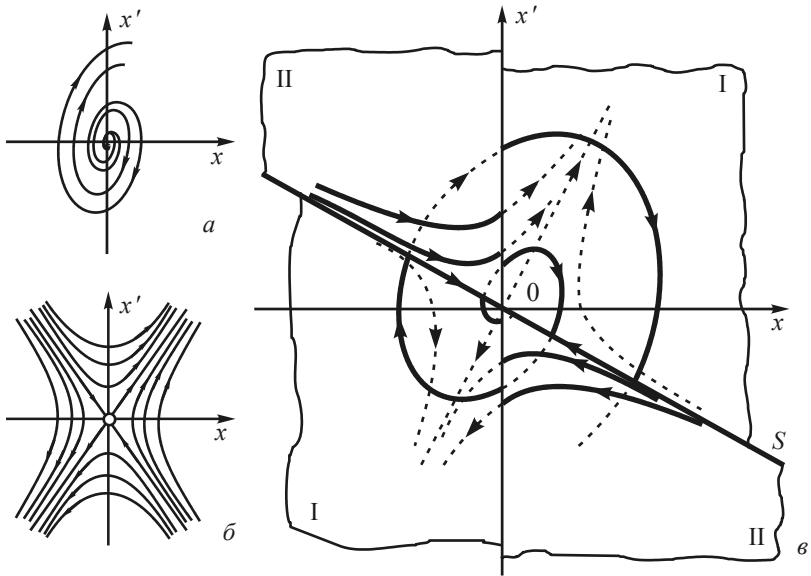


Рис. 4.16. Фазовые портреты

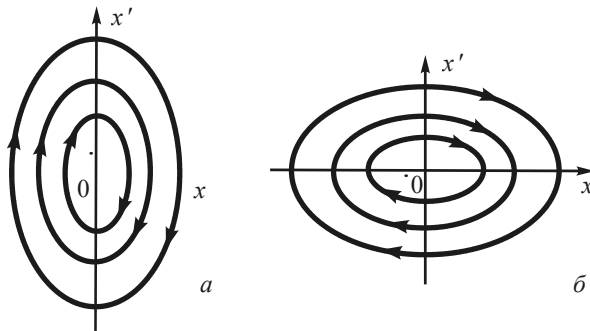


Рис. 4.17. Фазовые портреты в линейных структурах

Допустим, что в первом и третьем квадрантах плоскости  $(x, x')$  фазовыми траекториями изображающей точки являются траектории первой структуры (см. рис. 4.17, *a*), а во втором и четвертом квадрантах – фазовые траектории второй структуры (см. рис. 4.17, *б*). Несомненно,

если изменение структуры происходит на координатных осях и фазовая траектория системы имеет вид, представленный на рис. 4.18, то при любых начальных условиях движение в системе будет устойчивым.

Самой продуктивной идеей синтеза систем с переменной структурой оказалась идея создания искусственных вырожденных движений.

Суть этого подхода заключается в следующем: в фазовом пространстве  $X$  задается некоторая гиперплоскость  $S$ , движение в которой обладает желаемыми свойствами, причем траектории, лежащие в  $S$ , не принадлежат ни одной из имеющихся структур. Порядок изменения структур должен быть выбран таким, чтобы изображающая точка всегда попадала на эту гиперплоскость, а затем двигалась по ней.

В этом случае в момент попадания в систему будет существовать искусственное вырожденное движение, которое можно наделить рядом полезных свойств, не присущих каждой из фиксированных структур. Смысл создания искусственных вырожденных движений может быть проиллюстрирован на следующем примере. Допустим, в системе по-прежнему предусмотрена возможность использования двух фиксированных линейных структур, которым соответствуют фазовые портреты (рис. 4.19, а, б).

Пусть изменение структуры системы осуществляется на оси  $x'$  и на прямой  $S^*$ , которая не является фазовой траекторией ни для одной из имеющихся структур (см. рис. 4.19, в). Ось  $x'$  и прямая  $S^*$  разделяют фазовую плоскость на два листа, причем на листе I изображающая точка движется по эллиптическим кривым, а на листе II – по гиперболическим. Изображающая точка попадает на границу  $S^*$  из любого начального положения. Если угловой коэффициент прямой  $S^*$  больше углового коэффициента прямой  $S$ , то в окрестности  $S^*$  фазовые траектории будут направлены друг к другу (см. рис. 4.19, в). В результате, попав на  $S^*$ , изображающая точка в дальнейшем ее не покинет, т.е. пря-

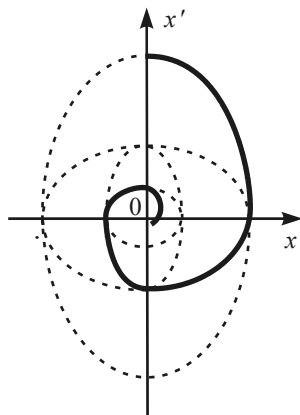


Рис. 4.18. Фазовая траектория

мая  $S^*$  является фазовой траекторией. Следовательно, удастся обеспечить существование вырожденных движений.

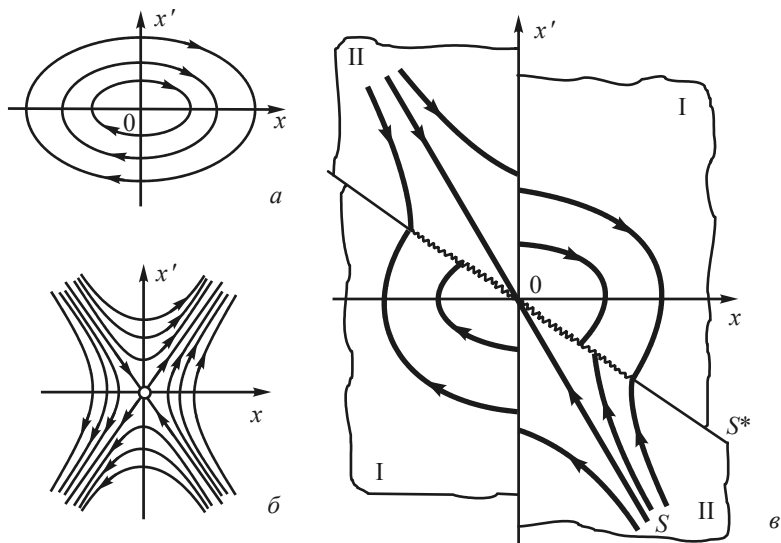


Рис. 4.19. Фазовые портреты

#### 4.9. ТИПЫ ДВИЖЕНИЯ В СИСТЕМАХ С ПЕРЕМЕННОЙ СТРУКТУРОЙ

При синтезе систем с ограниченной информацией о состоянии управляемого процесса, а также систем при наличии ограничений на уровни воздействий по различным координатам системы параметры управляющего устройства выбирались таким образом, чтобы, по крайней мере, одна из линейных структур имела устойчивое вырожденное движение. В момент попадания изображающей точки на фазовую траекторию, соответствующую устойчивому вырожденному движению, происходило переключение управления, т.е. замена одной линейной структуры на другую. В дальнейшем будем называть прямую, на которой осуществляется изменение структуры системы, прямой переключения.

В результате неточного знания параметров объекта, неточностей аппаратурной реализации, вызванных нестабильностью характеристик элементов управляющего устройства, и других технических причин не удастся обеспечить изменение структуры системы строго в требуемые моменты времени. Из-за этого в системах с переменной структурой могут возникать различные виды движения. Проиллюстрируем основные режимы движения, которые могут наблюдаться в этом классе систем.

Пусть существует система с переменной структурой второго порядка, описываемая следующей системой дифференциальных уравнений:

$$\begin{cases} \frac{dx_1}{dt} = x_2, \\ \frac{dx_2}{dt} = -a_2x_2 - a_1x_1 - bu, \end{cases} \quad (4.8)$$

где  $a_1, a_2, b - \text{const}, b > 0$ .

Специфика всех видов движения может быть целиком выявлена на примере системы, для которой управление имеет вид

$$u = \Psi x_1; \quad (4.9)$$

$$\Psi = \begin{cases} \alpha & \text{при } x_1 s > 0, \\ \beta & \text{при } x_1 s < 0; \end{cases} \quad (4.10)$$

$$s = x_2 + cx_1, \quad (4.11)$$

где  $\alpha, \beta, c - \text{постоянные коэффициенты}, c > 0, \alpha > \beta$ .

Согласно (4.10), (4.11), изменение структуры управляющего устройства осуществляется на прямых переключения  $x_1 = 0$  и  $S$ , заданной на плоскости  $(x_1, x_2)$  уравнением  $s = 0$ .

Пусть при  $\Psi = \alpha$  линейная структура (4.10) является колебательно неустойчивой, а при  $\Psi = \beta - \text{апериодически неустойчивой}$ . Для второй структуры имеется устойчивое вырожденное движение, определяемое отрицательным корнем  $\lambda$  характеристического уравнения.

Если коэффициент  $c$ , определяющий положение прямой переключения на фазовой плоскости  $(x_1, x_2)$ , равен величине  $-\lambda$ , в системе возникает такой режим работы, при котором изменение структуры системы происходит на фазовой траектории, соответствующей устойчивому вырожденному движению (рис. 4.20, а). Такой режим будем называть *режимом работы с движением по вырожденным траекториям*.

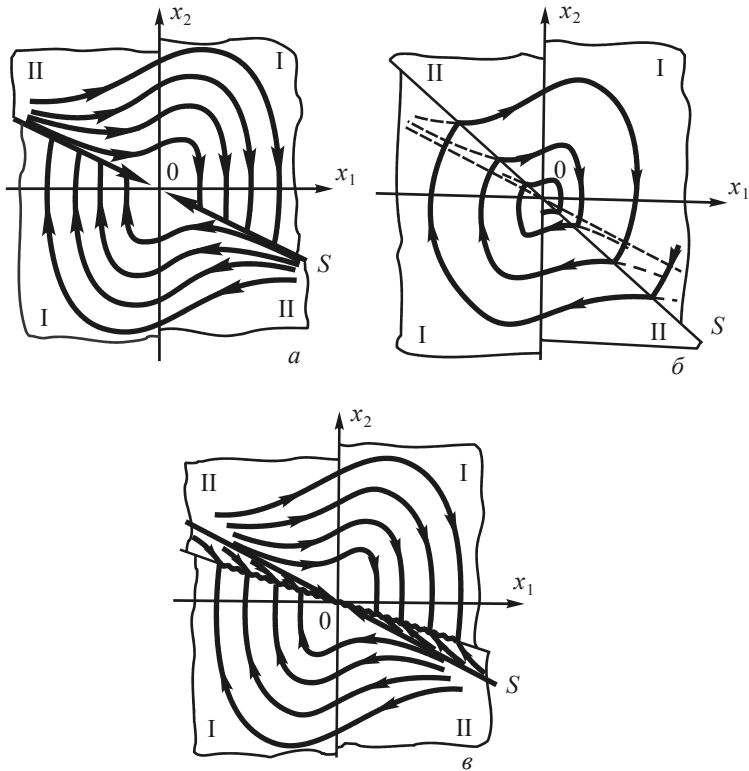


Рис. 4.20. Фазовые портреты

Пусть угловой коэффициент  $c$  прямой переключения больше  $-\lambda$ . Тогда устойчивая траектория линейной структуры, определяемая  $\Psi = \beta$ , будет целиком принадлежать листу I ( $x_1 > 0$ ), на котором движение изображающей точки происходит по раскручивающимся спиральям, соответствующим линейной структуре  $\Psi = \alpha$  (см. рис. 4.20, б).

Тогда после попадания на прямую переключения  $S$  изображающая точка будет уходить от нее, двигаясь по кривым гиперболического типа. В случае изменении знака  $x_1$  произойдет смена структуры системы. Далее изображающая точка, двигаясь по раскручивающимся спиральям, вновь попадет на прямую переключения  $S$  и т.д. В рассмотренной системе с некоторой конечной частотой будет изменяться структура управ-



ляющего устройства. Подобный режим работы системы будем называть *режимом переключений*. Как следует из фазового портрета, при выбранном значении  $s$  система устойчива, но положение равновесия достигается в колебательном режиме.

Пусть угловой коэффициент  $s$  прямой переключения меньше  $-\lambda$ . В таком соотношении параметров системы траектория, характеризующая устойчивое вырожденное движение, принадлежит листу  $\Pi$  ( $x_1 s < 0$ , см. рис. 4.20,  $\epsilon$ ) и фазовые траектории, а точнее говоря, векторы фазовых скоростей обеих структур, в окрестности прямой переключения  $S$  направлены к  $S$ . При попадании изображающей точки на прямую  $S$  в системе с бесконечно большой частотой возникают переключения структуры системы с одной на другую. Причем на фазовой плоскости изображающая точка будет двигаться вдоль прямой переключения. Подобный режим работы, при котором на прямой переключения изменение структуры происходит с бесконечно большой частотой, будем называть *скользящим режимом*. Следует отметить, что прямая переключения не является фазовой траекторией ни для одной из линейных структур. Таким образом, за счет скользящего режима удастся получить искусственное вырожденное движение. Вследствие этого остановимся подробнее на уравнениях движения системы в скользящем режиме и условиях его возникновения.

Так как изображающая точка, попав на прямую переключения, уже не может сойти с нее и продолжает свое движение по ней, следовательно, величина  $s$  в скользящем режиме равна нулю, т.е.

$$x_2 + cx_1 = 0. \quad (4.12)$$

Учитывая, что величина ошибки  $x$  равна  $x_1$ , а  $dx/dt = x_2$ , получаем из (4.12) дифференциальное уравнение движения системы в скользящем режиме относительно координаты ошибки:

$$dx/dt + cx = 0. \quad (4.13)$$

В соответствии с (4.13) движение рассматриваемой системы в скользящем режиме описывается уже уравнением первого порядка, и, что особенно важно, это уравнение не зависит от параметров неизменяемой части системы. Изменяя соответствующим образом коэффициент  $s$ , мы можем наделять движение системы в скользящем режиме желаемыми свойствами. Например, для устойчивости достаточно принять  $c > 0$ , а увеличивая  $c$ , можно повысить скорость затухания этого движения. При движении в скользящем режиме система приобретает свойство

инвариантности от характеристик управляемого объекта, а также получает возможность наделять ее желаемыми свойствами, что обуславливает широкое использование этого вида движения в рассматриваемых системах.

Заключительный этап процесса управления всегда будет протекать в скользящем режиме, если при выбранных линейных структурах управляющего устройства и при выбранной последовательности их изменения в любой точке прямой переключения  $S$  существует скользящий режим, а изображающая точка из любого начального положения попадает на  $S$ .

Условие существования скользящего режима на прямой  $S$  можно получить в соответствии с тем, что в окрестности  $S$  векторы фазовых скоростей обеих структур должны быть направлены навстречу друг другу.

Пусть изображающая точка отклонилась от прямой переключения на некоторое расстояние, и для определенности величина  $s$ , характеризующая это отклонение, положительна,  $s > 0$ . Для того чтобы изображающая точка вернулась на прямую  $S$ , в системе должна включиться такая линейная структура, при которой величина  $s$  будет убывать, т.е.  $ds/dt < 0$ . Аналогично можно убедиться в том, что для отклонений  $s < 0$  должна быть такая структура, при которой  $ds/dt > 0$ . Из этого вытекает условие, которое должно выполняться в окрестности  $S$  для того, чтобы на этой прямой существовал скользящий режим:

$$s \cdot ds/dt < 0. \quad (4.14)$$

При выполнении данного неравенства для любой точки прямой переключения  $S$ , т.е. на всей прямой  $S$ , существует скользящий режим. Условимся называть в этом случае прямую  $S$  *прямой скольжения* [13].

В результате динамическое изменение структуры управляющего устройства дает дополнительные возможности по улучшению процессов регулирования.

#### 4.10. ВЫВОДЫ

1. Использование принципов переменности структуры позволяет решить задачу высококачественного управления линейным объектом и при этом обеспечить минимальные перемещения регулирующего органа, не превосходящие их установившегося значения.

2. Существование неидеальностей в релейных элементах систем с переменной структурой (запаздывание при переключении, присутствие гистерезиса и т.п.) отрицательно сказывается на динамике всей системы, переходный процесс которой в этом случае сопровождается автоколебаниями.

3. Большое количество технологических процессов описывается передаточными функциями, принадлежащими к выделенному классу объектов.

## ЛИТЕРАТУРА

1. *Автоматизация* настройки систем управления / В.Я. Ротац, В.Ф. Кузищин, А.С. Клюев и др. – М.: Энергоатомиздат, 1984. – 272 с.

2. *Автоматизация* производства и промышленная электроника / Под ред. А.И. Берг, В.А. Трапезникова и др.: В 4-х т. – М.: Советская энциклопедия, 1962–1965.

3. *Андык В.С.* Теория автоматического управления: Учебное пособие к практическим занятиям. – Томск: Изд-во ТПУ, 2004. – 108 с.

4. *Боднер В.А.* Теория автоматического управления полетом. – М.: Наука, 1964. – 700 с.

5. *Дорф Р., Бишоп Р.* Современные системы управления / Пер. с англ. Б.И. Копылова. – М.: Лаборатория базовых знаний, 2002. – 832 с.

6. *Емельянов С.В.* Системы автоматического управления с переменной структурой. – М.: Наука, 1967. – 336 с.

7. *Емельянов С.В., Коровин С.К.* Новые типы обратной связи. – М.: Наука, 1997. – 352 с.

8. *Клюев А.С., Лебедев А.Т.* Наладка систем автоматического регулирования барабанных паровых котлов. – М.: Энергия, 1985. – 280 с.

9. *Круг Е.К., Мина О.М.* Электрические регуляторы промышленной автоматики. – М.: Госэнергоиздат, 1962. – 336 с.

10. *Системы с переменной структурой и их применение в задачах автоматизации полета:* Сб. ст. / Под ред. Б.Н. Петрова и С.В. Емельянова. – М.: Наука, 1968. – 324 с.

11. *Солодов А.В., Петров Ф.С.* Линейные автоматические системы с переменными параметрами. – М.: Наука, 1971. – 620 с.

12. *Стефани Е.П.* Основы расчета настройки регуляторов теплоэнергетических процессов. – М.: Энергия, 1972. – 376 с.

13. *Теория систем с переменной структурой* / Под ред. С.В. Емельянова. – М.: Наука, 1970. – 592 с.

14. *Фритч В.* Применение микропроцессоров в системах управления / Пер. с нем. – М.: Мир, 1984. – 464 с.

15. *Шидловский С.В.* Автоматизация технологических процессов и производств: Учебное пособие. – Томск: Изд-во НТЛ, 2005. – 100 с.
16. *Цыткин Я.З.* Основы теории автоматических систем. – М.: Наука, 1977. – 560 с.
17. *Чечкин А.В.* Математическая информатика. – М.: Наука, 1991. – 416 с.
18. *Энциклопедия кибернетики* / Под ред. В.М. Глушкова и др. – Киев, 1975. – Т. 1, 2.

## **СИНТЕЗ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ С ПЕРЕСТРАИВАЕМОЙ СТРУКТУРОЙ**

---

---

### **5.1. ПОСТАНОВКА ЗАДАЧИ УПРАВЛЕНИЯ**

Суть вопроса сводится к выбору такого управления  $u$ , при котором выходное значение  $y$  объекта управления совпадало бы с задающим значением  $s$  или их разница лежала бы в допустимых пределах при изменении внешнего возмущающего воздействия  $f$  и  $a$ .

Возмущение  $f$  называется координатным, а возмущение  $a$  – параметрическим. Под влиянием внешних возмущений, информации о которых часто недостаточно, взаимосвязь между входом и выходом объекта становится неоднозначной и неопределенной, что сильно затрудняет решение задачи.

Координатное возмущение есть неизвестная величина со стороны нагрузки на объект управления, которая проявляется в виде неконтролируемых произвольных изменений технологических параметров и по характеру изменения во времени может быть импульсной и медленно меняющейся. Параметрическое возмущение есть неизвестная величина из некоторого ограниченного множества, в результате действия которой происходит медленное изменение параметров объекта управления.

Следует отметить принципиальное различие между этими двумя типами возмущений. Рассмотрим случай, когда на объект управления  $W_{об}(p)$  со входным сигналом  $u$  и выходом  $y$  действуют оба типа возмущающих воздействий. Тогда выходная координата объекта примет вид

$$y = W_{об}(p, a)g = W_{об}(p, a)(f + u) = W_{об}(p, a)f + W_{об}(p, a)u .$$

Теперь наглядно видно качественное различие влияния возмущений  $f$  и  $a$  на выходе объекта. Координатное возмущение  $f$  вносит аддитивный и независимый от входа  $u$  вклад в реакцию объекта, равный

$W_{об}(p, a)f$ . Параметрическое же возмущение  $a$  изменяет только вид или параметры  $W_{об}(p, a)$  и не имеет независимого от  $u$  и  $f$  влияния на выход объекта.

Таким образом, возмущение  $f$  формирует «линейное» воздействие внешней среды на регулируемую координату, а возмущение  $a$  – «нелинейное» ее воздействие.

Рассмотрим, как классические законы регулирования влияют на качество переходного процесса системы автоматического регулирования. В операторной форме уравнение регулятора имеет вид

$$u = W_{пер}(p)\varepsilon, \quad (5.1)$$

где  $u$  – выходная величина регулятора (управляющее воздействие);  $\varepsilon$  – входная величина регулятора (ошибка регулирования). В скобках приведены названия величин для замкнутой одноконтурной системы автоматического регулирования, состоящей из последовательно соединенных регулятора и объекта управления, охваченных единичной отрицательной обратной связью.

Такой показатель качества, как статическая ошибка, является одним из важных и характеризует точность регулирования в статическом режиме.

Под статической ошибкой регулирования  $\varepsilon_{ст}$  будем понимать отклонение регулируемой величины от заданного значения по окончании переходного процесса.

Покажем, как *пропорциональный* закон регулирования влияет на статическую ошибку переходного процесса замкнутой системы автоматического регулирования.

Уравнение (5.1) в этом случае приобретает вид

$$u = k_{н} \cdot \varepsilon,$$

где  $k_{н}$  – коэффициент передачи регулятора. Передаточная функция  $W_{пер}(p)$  может иметь и более сложный вид, например:

$$W_{пер}(p) = k_{н} \frac{A(p)}{B(p)},$$

где  $A(p)$  и  $B(p)$  – некоторые полиномы от оператора  $p$ . Однако цепь регулирования в этом случае представляет собой статическое звено, и

при  $p \rightarrow 0$  передаточная функция  $W_{\text{пер}}(p) \rightarrow k_{\text{п}}$  (т.е. установившийся режим, так как приравнивание оператора дифференцирования нулю означает приравнивание нулю всех производных).

Передаточная функция разомкнутой одноконтурной системы автоматического регулирования имеет вид

$$W_{\text{р.с}}(p) = W_{\text{пер}}(p)W_{\text{об}}(p) = k_{\text{п}}W_{\text{об}}(p),$$

где  $W_{\text{об}}(p)$  – передаточная функция объекта управления. В установившемся состоянии передаточная функция разомкнутой системы стремится к значению

$$\lim_{p \rightarrow 0} W_{\text{р.с}}(p) = k_{\text{п}} \cdot k_{\text{об}} = K.$$

Величина  $K$  называется обобщенным коэффициентом передачи разомкнутой системы и физически представляет собой отношение установившегося значения регулируемой величины к постоянному значению ошибки  $\varepsilon = \varepsilon_0$ , если цепь регулирования совместно с объектом рассматривать как некоторый усилитель, на входе которого действует сигнал в виде ошибки  $\varepsilon$ , а на выходе – усиленный сигнал  $y$ . В итоге для коэффициента передачи можно записать

$$K = \frac{y_{\text{уст}}}{\varepsilon_0}.$$

Запишем выражение в операторной форме для замкнутой системы автоматического регулирования относительно ошибки [3]:

$$\varepsilon = \frac{s}{1 + W_{\text{р.с}}(p)} + \frac{W_f(p)}{1 + W_{\text{р.с}}(p)} f, \quad (5.2)$$

где  $s$  – сигнал задания;  $f$  – возмущающее воздействие;  $W_f(p)$  – передаточная функция объекта управления по возмущающему воздействию.

Тогда для установившегося состояния замкнутой системы при постоянном задающем воздействии  $s = s_0 = \text{const}$  получаем

$$\varepsilon_{\text{ст}} = \frac{s_0}{1 + K} + \frac{\varepsilon_{f \text{ уст}}}{1 + K},$$

где  $\varepsilon_{f_{уст}}$  – установившееся значение ошибки от возмущающего воздействия в объекте без регулирования.

Таким образом, пропорциональное регулирование позволяет уменьшить установившиеся ошибки в объекте в  $1+K$  раз. Для сведения статической ошибки к нулю в случае пропорционального регулирования необходимо, чтобы  $k_{и} \rightarrow \infty$ , так называемый случай глубокой обратной связи, что ведет к затруднению в практическом использовании пропорционального регулирования для сведения  $\varepsilon_{ст}$  к нулю.

В случае *интегрального* регулирования уравнение (5.1) принимает вид

$$u = \frac{k_v}{p} \varepsilon,$$

где  $k_v = k_{и} / T_{и}$  – скорость разгона;  $k_{и}$  – коэффициент передачи;  $T_{и}$  – время интегрирования.

Аналогично предыдущему случаю передаточная функция разомкнутой системы принимает вид

$$W_{p.c}(p) = \frac{k_v}{p} W_{об}(p).$$

В установившемся режиме передаточная функция разомкнутой системы  $\lim_{p \rightarrow 0} W_{p.c}(p) \rightarrow \infty$ . В результате первая составляющая ошибки в уравнении (5.2), для установившегося состояния, при  $s = s_0 = \text{const}$  обращается в нуль. Вторая составляющая, определяемая наличием возмущающих воздействий, может не обращаться в нуль, так как в установившемся состоянии числитель ее может также стремиться к бесконечности. Поэтому при  $f = f_0 = \text{const}$

$$\varepsilon_{ст} = \lim_{p \rightarrow 0} \frac{W_f(p) f_0}{1 + W_{p.c}(p)}.$$

Это значение может быть как равным нулю, так и отличным от нуля.

Таким образом, при интегральном регулировании получается система, астатическая по отношению к задающему воздействию. По отно-



шению к возмущающему воздействию она может быть как статической, так и астатической.

Еще одним из важных показателей качества переходного процесса является время регулирования, время, по истечении которого регулируемая величина начинает отличаться от установившегося значения менее чем на заранее заданное значение. Интегральный закон регулирования обладает низким быстродействием. Это объясняется тем, что при  $t=0$  в системе не только регулирующее воздействие равно нулю, но равна нулю также и его первая производная, т.е. должно пройти некоторое время, пока не «накопится» интеграл.

Предположим, что ошибка  $\varepsilon$  изменяется скачкообразно (рис. 5.1). В этом случае выходная величина регулятора будет изменяться по закону  $u(t) = k_v \cdot t$ . Очевидно, чем больше величина  $k_v$ , тем больше будет угол  $\alpha$  и регулятор будет быстрее изменять выходную величину. Однако в системах автоматического регулирования это приведет к тому, что при малых изменениях ошибки регулирования выходная величина регулятора будет продолжать быстро меняться, вследствие чего у системы появится склонность к колебаниям.

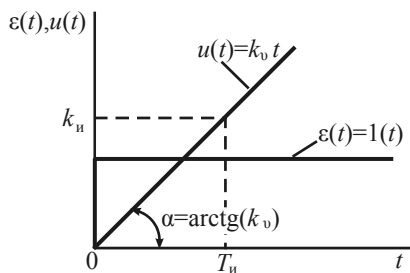


Рис. 5.1. Реакция интегрального регулятора  $u(t)$  на ошибку регулирования  $\varepsilon(t)$

Здесь уже приходится считаться с такими показателями качества, как перерегулирование  $\sigma$  и степень затухания  $\psi$ . Поэтому использование данного типа регулятора предполагает поиск компромисса между быстродействием и степенью затухания.

Пропорциональный регулятор имеет лучшее быстродействие, поэтому использование совместно пропорционального и интегрального регулирования дает некоторый выигрыш в быстродействии и сведении

статической ошибки к нулю. Такое регулирование получило название изодромного или пропорционально-интегрального (ПИ). Уравнение (5.1) в этом в этом случае принимает вид

$$u = k_n \left( 1 + \frac{1}{T_{из} p} \right) \varepsilon, \quad (5.3)$$

где  $k_n$  – коэффициент передачи регулятора;  $T_{из} = k_n / k_v = T_i \cdot k_n / k_n$  – время изодрома. В случае установившегося режима  $p = 0$  регулирование оказывается астатическим относительно задающего воздействия.

Изодромное регулирование сочетает в себе высокую точность интегрального регулятора и быстроту пропорционального. Это определяется первым слагаемым в уравнении (5.3). В дальнейшем система начинает работать как система интегрального регулирования, так как с течением времени преобладающее значение приобретает второе слагаемое (5.3), что приводит к дальнейшему «затягиванию» процесса.

Увеличить быстродействие можно с помощью введения производной в канал регулирования, которая будет учитывать тенденцию роста ошибки регулирования. Такой регулятор называется пропорционально-интегро-дифференциальным (ПИД). Уравнение (5.1) в этом случае примет вид

$$u = k_n \left( 1 + \frac{1}{T_y p} + T_{пр} p \right) \varepsilon,$$

где  $T_y = k_n T_i / k_n$  – постоянная времени удвоения;  $T_{пр} = T_d / k_n$  – постоянная времени предварения;  $T_d$  – постоянная времени дифференцирования.

Ввод производной дает повышение быстродействия по отношению к изодромному регулятору, но чтобы это быстродействие было ощутимым на практике, необходимо:

- точное описание модели объекта регулирования;
- точное выполнение операции дифференцирования.

Реальные же процессы всегда описываются уравнениями более высоких порядков, чем модель, а воспроизведение точной операции дифференцирования имеет трудности с физической реализуемостью [6].

Мы рассматривали случай, когда задающее воздействие является постоянным. При изменении задающего воздействия как функции времени ситуация существенно меняется. Так, например, если задание – полином по  $t$  степени  $m$ , т.е. в нуль тождественно обращаются все его производные, начиная с  $(m+1)$ -й, то для достижения астатизма в этом случае потребуется не менее чем  $m$  интегрирований ошибки регулирования. При  $m \rightarrow \infty$  порядок линейного регулятора растет до бесконечности, что, конечно, практически неприемлемо. В частности, таким путем нельзя построить астатическую систему для экспоненциально растущего задания  $s(t) = \exp(\alpha t)$ ,  $\alpha = \text{const} > 0$ , так как

$$\exp(\alpha t) = \sum_0^{\infty} \frac{(\alpha t)^m}{m!}.$$

Это потребовало бы бесконечно мерного регулятора, поэтому решение следует искать в классе систем с переменной структурой [5], для которых уравнение (5.1) имеет вид

$$u = \Psi \cdot \varepsilon,$$

где  $\Psi$  – логический закон регулирования. В таких системах удастся получить свойства устойчивости движения, сочетаемые с высоким быстродействием и точностью при использовании такой информации, которая была недостаточна для стабилизации линейной системы.

Однако в ряде случаев требуется изменение самого логического закона регулирования  $\Psi$ . В результате система должна не менять свою структуру, а перестраивать ее, т.е. изменять не только комбинационные связи, но также и функциональные.

Таким образом, постановка рассматриваемой задачи управления заключается в стабилизации регулируемой величины, при неполной информации об объекте управления, и получении свойств автоматической системы регулирования, инвариантных к внешним возмущающим воздействиям, без введения в канал управления производной, с конечным коэффициентом усиления обратной связи и возможностью перестройки внутренней структуры регулятора.

## 5.2. ТИПОВАЯ СИСТЕМА РЕГУЛИРОВАНИЯ

Наиболее широкое применение в практике нашли автоматические системы регулирования (АСР) по отклонению [9]. Типовая структурная схема такой системы представлена на рис. 5.2.

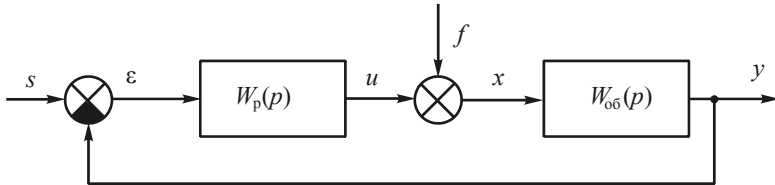


Рис. 5.2. Структурная схема системы автоматического регулирования

Здесь  $W_{об}(p)$  – передаточная функция объекта регулирования;  $W_p(p)$  – передаточная функция регулятора;  $s$  – задающее воздействие;  $\varepsilon$  – ошибка регулирования;  $f$  – внешнее возмущающее воздействие;  $u$  – управляющее воздействие;  $x = f + u$  – воздействие на объект с учетом внешних возмущений;  $y$  – выходная регулируемая величина.

Расчет такой системы регулирования сводится к определению оптимальных параметров выбранного регулятора.

Приведем динамические процессы, протекающие в подобных системах. Для этого произведем расчет одноконтурной системы регулирования, состоящей из ПИ-регулятора и объекта регулирования с передаточной функцией вида (4.5), при  $T_1 = T = 97,7$  с;  $T_2 = 0$ ;  $k = 6$ ;  $\tau = 23,4$  с:

$$W_{об}(p) = \frac{k \exp(-\tau p)}{Tp + 1} = \frac{6 \exp(-23,4p)}{97,7p + 1}. \quad (5.4)$$

Расчет оптимальных параметров настройки ПИ-регулятора произведем методом расширенных амплитудно-фазочастотных характеристик (РАФЧХ). Одним из существенных его достоинств является то, что метод наиболее оптимален с точки зрения предъявляемых требований к переходным характеристикам в промышленных системах [2, 13].

С помощью разработанной программы в среде MatLab (прил. 2) рассчитаны по методу РАФЧХ граница заданного запаса устойчиво-

сти (рис. 5.3) и переходный процесс замкнутой системы автоматического регулирования, а также произведена оценка его качества.

При определении настроек регуляторов в качестве показателя оптимальности системы регулирования обычно выбирают интегральный критерий качества (например, интегральный квадратичный критерий) при действии на объект наиболее тяжелого возмущения с учетом добавочного ограничения на запас устойчивости системы.

В дальнейшем под оптимальными будем понимать настройки регулятора, обеспечивающие заданную степень колебательности  $m$  процесса регулирования при минимуме интегрального квадратичного критерия  $I_{\text{КВ}}$ .

Точка на графике, соответствующая максимуму (т.е. минимуму первого интегрального критерия), определит параметры настройки ПИ-регулятора:

$$K_p = 0,47 \frac{\text{размерность управляющего воздействия}}{\text{размерность регулируемой величины}}; T_i = 52,6 \text{ с.}$$

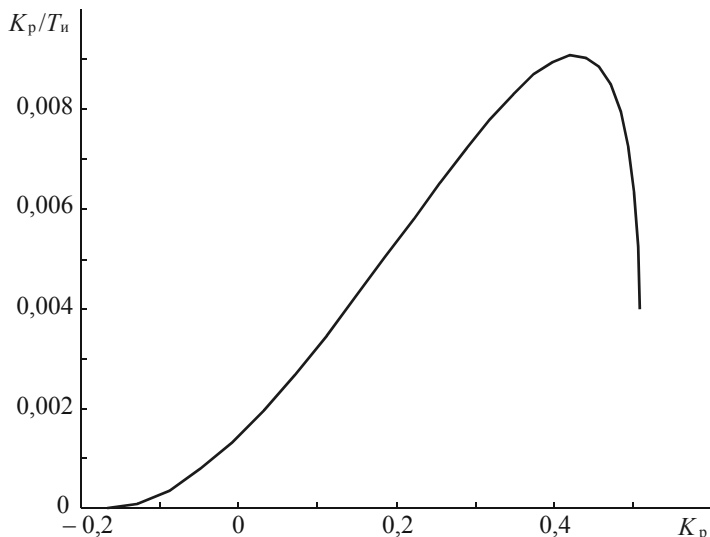


Рис. 5.3. Граница заданного запаса устойчивости

Тогда передаточную функцию ПИ-регулятора можно записать в следующем виде:

$$W_p(p) = K_p \frac{(T_{\text{и}}p + 1)}{T_{\text{и}}p} = 0,47 \frac{(52,6p + 1)}{52,6p}. \quad (5.5)$$

Характер переходного процесса системы автоматического регулирования, по каналу задающего воздействия, при оптимальных параметрах настройки регулятора показан на рис. 5.4 для различных постоянных времени объекта управления.

Любая промышленная АСР, кроме устойчивости, должна обеспечивать определенные качественные показатели процесса регулирования.

Качество процесса регулирования оценивают по переходной характеристике по отношению к единичному ступенчатому возмущающему воздействию [4].

Склонность системы к колебаниям, а следовательно, и запас устойчивости могут быть охарактеризованы максимальным значением регулируемой величины  $y_{\text{max}}$  или так называемым перегуливанием:

$$\sigma = (A_{\text{max}} - 1)/1 \cdot 100\% = 38\%.$$

Время регулирования  $t_p$  – время, по истечении которого отклонение регулируемой величины от установившегося состояния не будет превышать некоторой наперед заданной величины  $\delta$ . Обычно  $\delta = 0,05y(\infty) = 0,05 \cdot 1 = 0,05$ ;  $t_p = 136$  с.

Максимальная динамическая ошибка

$$A_{\text{max}} = 0,38.$$

Степень затухания называется отношение разности двух соседних амплитуд, одного знака, к большей из них:

$$\psi = (A_1 - A_2)/A_1 \approx 0,96.$$

Как видно из рис. 5.4, при различных постоянных времени объекта управления переходные процессы автоматической системы регулирования имеют различные качественные показатели, близкие к оптимальным только при рассчитанных, заранее фиксированных параметрах настройки регулятора.

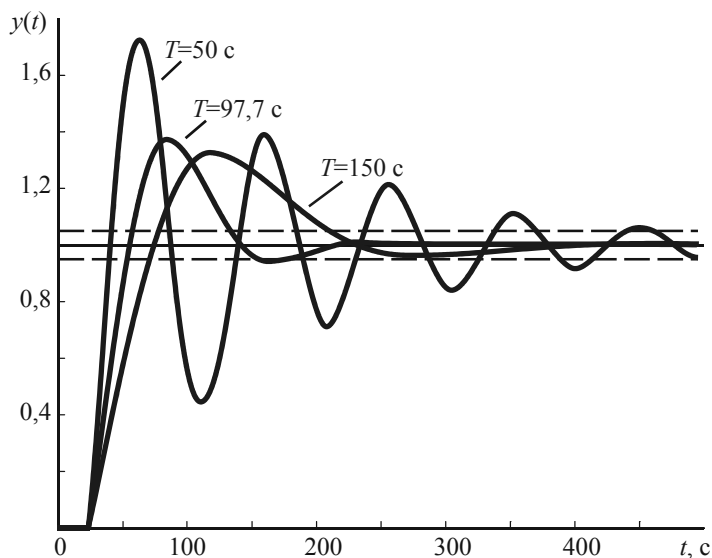


Рис. 5.4. Переходный процесс в замкнутой системе регулирования по каналу задающего воздействия

Обозначим диапазон изменения постоянной времени объекта через  $\Delta T$ , а соответствующее изменение степени затухания через  $\Delta \psi$ . Тогда обобщенный показатель качества системы относительно степени затухания будет характеризовать отношение [8]:

$$\psi' = \frac{|\Delta \psi|}{|\Delta T|}. \quad (5.6)$$

Чем меньше это отношение, тем больше обобщенный показатель качества системы относительно степени затухания. На рис. 5.5 представлены результаты моделирования для рассматриваемой системы автоматического регулирования при изменении постоянной времени объекта управления в диапазоне от 50 до 150 с.

Таким образом, при параметрических возмущениях в системе регулирования с линейным регулятором существенно снижаются показатели качества переходного процесса. В случае широкого диапазона изменения параметров объекта управления данный аспект может привести систему автоматического регулирования к неустойчивому состоянию.

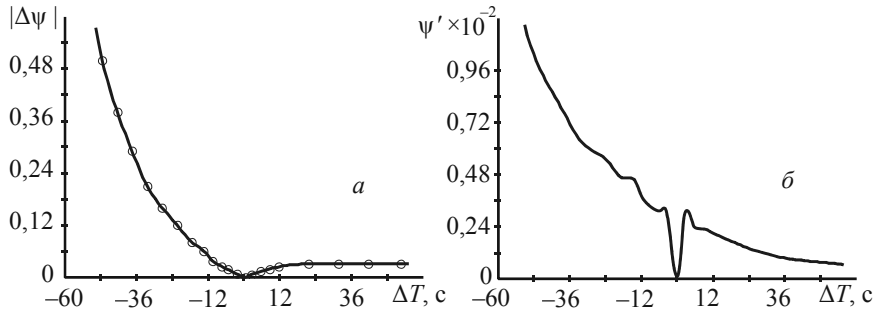


Рис. 5.5. Результаты моделирования системы автоматического регулирования при параметрическом возмущении: *а* – модуль изменения степени затухания; *б* – обобщенный показатель качества системы относительно степени затухания

### 5.3. АДАПТИВНАЯ СИСТЕМА АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ

Общепринятый порядок синтеза систем управления состоит в следующем:

- задается математическая модель объекта (на практике это обычно модель, полученная на основе экспериментальной переходной характеристики объекта);
- принимается критерий оптимальности системы управления;
- по модели объекта определяются структура и численные значения параметров алгоритма функционирования контроллера (регулятора), удовлетворяющие принятому критерию оптимальности.

Считается, что если модель достаточно близка к реальному объекту, а выбранный метод синтеза и расчеты выполнены безукоризненно, то спроектированная система заработает без какой-либо существенной доводки при пуске. Однако опыт свидетельствует о том, что такой оптимистический прогноз, как правило, не оправдывается. Объясняется это двумя причинами:

- системным характером задачи получения математической модели объекта; это значит, что для формулировки критерия приближения последней необходимо располагать алгоритмом функционирования контроллера, для определения которого, собственно, и нужна эта модель [15];



– практической невозможностью учета отклонения принимаемой в расчетах динамической модели контроллера от реальной (наличие широтно-импульсного преобразования сигнала на выходе контроллера, зоны нечувствительности, люфтов в механических сочленениях исполнительного механизма и т.п.).

Выход из сложившейся ситуации состоит в том, что системы управления даже с относительно стабильными объектами должны проектироваться как адаптивные (с автоматизированной настройкой). Эффективность таких систем определяется тем, что они оперируют всей системой в целом, причем при соответствующем выборе режима идентификации можно осуществлять автоматическую линеаризацию нелинейности в значительном для каждой конкретной системы диапазоне частот и отклонений сигналов [1].

В функции адаптации не входит подстройка параметров регуляторов к относительно быстро меняющимся свойствам объекта, вызванным контролируемыми возмущениями, прежде всего – изменениями нагрузки объекта. В этом случае должна применяться обычная коррекция настройки регуляторов по заранее заданным законам, реализуемым в соответствующих корректирующих блоках. Однако в функции адаптации входит настройка этих корректирующих блоков. Вообще возможности теории автоматического управления (как и любой другой теории) ограничены некоторыми пределами. При слишком быстрых изменениях свойств объекта и связанных с этим появлением нелинейных эффектов принципиальная возможность адаптации систем управления достаточно сложными в динамическом отношении объектами оказывается весьма проблематичной.

Структура адаптивной системы управления может быть представлена такой, как показана на рис. 5.6. К контуру регулирования, включающему объект Об и регулятор Рег, подсоединяется адаптирующее устройство Ад, на вход которого подаются входной  $u(t)$  и выходной  $y(t)$  сигналы объекта. В идентифицирующем устройстве Ид по полученным сигналам оценивается модель объекта, а в вычислительном устройстве ВУ определяются оптимальные параметры настройки регулятора, которые затем устанавливаются с помощью адаптирующего воздействия  $a(t)$ . Причем для реализации адаптирующего устройства Ад используется один из известных в настоящее время методов адаптации.

В работе применяется метод адаптации, использующий сигнальное гармоническое идентифицирующее воздействие (метод Циглера–Никольса) [18]. Достоинством такого метода является возможность обоснованного применения методов математической статистики в процессе проведения итерационной процедуры движения к оптимуму. Практическая значимость этого обстоятельства состоит в возможности уменьшения амплитуды воздействий до приемлемого уровня и, несмотря на это, получения удовлетворительных оценок параметров выходных колебаний благодаря увеличению продолжительности адаптации.

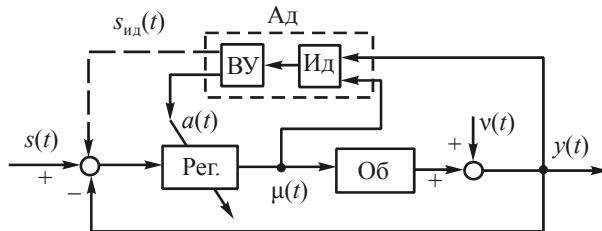


Рис. 5.6. Структура адаптивной системы

Наблюдение за состоянием объекта в процессе нормального функционирования без введения дополнительных поисковых составляющих не приводит к успеху. Объясняется это тем, что поскольку объект находится в составе системы, то и оперировать следует с воздействиями, являющимися входными сигналами всей системы; при этом входной сигнал следует выбрать таким образом, чтобы идентифицируемый канал системы зависел только от одного неизвестного оператора объекта.

Существующая проблема сводится не к тому, чтобы создать систему адаптации, функционирующую без вызванных идентифицирующими воздействиями добавочных отклонений регулируемой величины, а к тому, чтобы сделать эти отклонения достаточно малыми, приемлемыми для практики. Это достигается декомпозицией процедуры поиска с использованием в алгоритмах настройки нижнего уровня специально разработанных неэкстремальных критериев.

Такая процедура основана на использовании активных частотных методов идентификации объектов и расчета оптимальной настройки регулятора.

При формировании процесса оценки модели объекта структура адаптивной системы управления, приведенная на рис. 5.6, должна быть

дополнена еще одним идентифицирующим воздействием, которое должно оказывать адаптирующее устройство Ад на систему с целью идентификации объекта. Величина  $s_{ид}(t)$  показана пунктирной линией в виде сигнального воздействия, подаваемого на задатчик регулятора. Такое идентифицирующее воздействие не обязательно должно быть сигнальным. Оно может быть также алгоритмическим, параметрическим, структурным.

Зачастую автоматическая настройка осуществляется путем включения в канал сигнала ошибки двухпозиционного реле с малым выходным сигналом. Затем по параметрам автоколебаний, возникающих в замкнутой системе, определяются требуемые настройки регулятора. При этом при таком способе самонастройки происходит прекращение процесса регулирования объекта на время настройки, наблюдается высокая чувствительность к шумам в канале измерения, возникает опасность срыва автоколебаний при действии возмущений.

Имеется также алгоритм настройки регулятора в замкнутом контуре путем подачи на вход системы пробного синусоидального сигнала. Для этого алгоритма требуется достаточно большое время настройки (около 8–10 периодов колебаний на резонансной частоте замкнутой системы).

В работе используется метод Циглера–Никольса с частотным разделением каналов управления и самонастройки, что достигается включением двух режекторных цифровых фильтров в обратную связь контура регулирования.

На рис. 5.7 приведена структурная схема адаптивной системы управления. Основной контур состоит из настраиваемого регулятора Р, собственно объекта управления ОУ и двух режекторных фильтров (основного РФО и дополнительного РФД). Дополнительный режекторный фильтр при помощи переключателя  $\Pi_1$  включается только на первом этапе либо периодически для определения требуемых по методу Циглера–Никольса настроек. Блоки синхронного детектирования  $СД_1$ ,  $СД_2$  определяют установившиеся значения амплитуд  $A_6$ ,  $A_0$  и фаз  $\Phi_6$ ,  $\Phi_0$  пробных составляющих в выходных сигналах основного режекторного фильтра  $y_1$  и объекта управления  $y$ . Определение заданного фазового сдвига осуществляется с помощью блока фазовой автоподстройки частоты (БФАЧ).

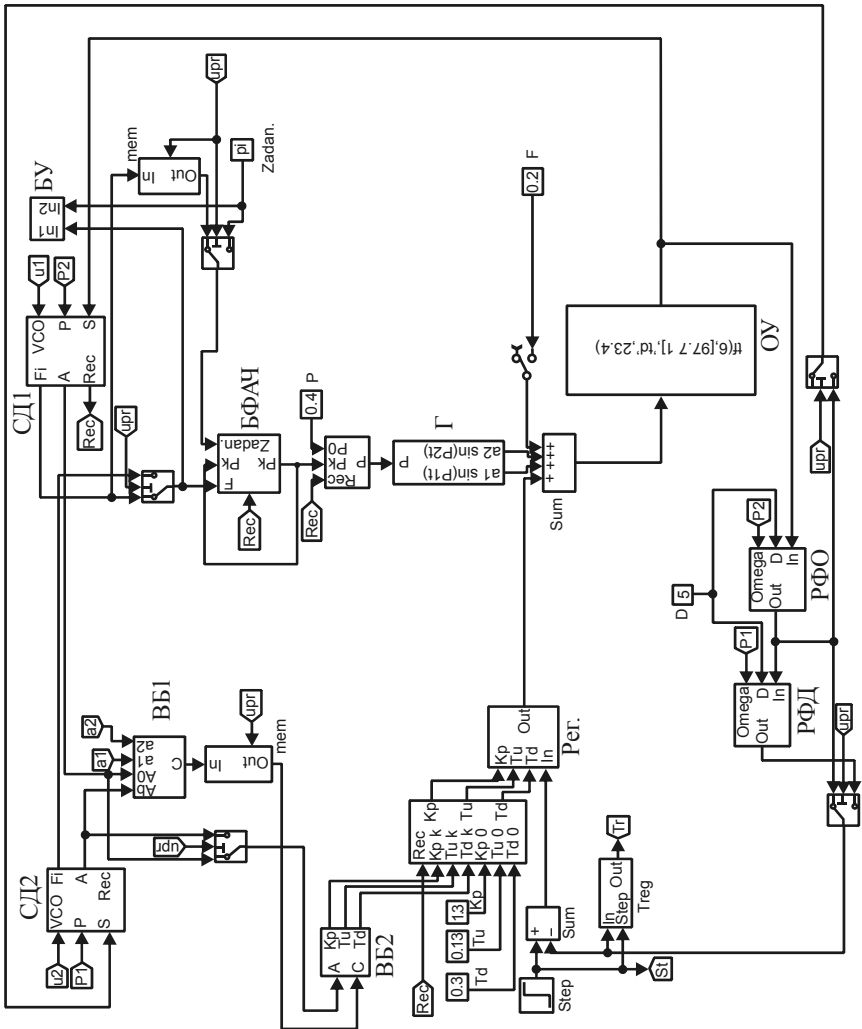


Рис. 5.7. Структурная схема адаптивной системы

Сравнивая установившиеся значения фазового сдвига  $\Phi_0$  с заданным  $\Phi_3$ , блок корректирует частоты пробных синусоидальных колебаний первого  $\Gamma_1$  и второго  $\Gamma_2$  генераторов. При этом частота генератора  $\Gamma_2$

выбрана вдвое выше частоты генератора  $\Gamma_1$ . Вычислительные блоки ВБ<sub>1</sub>, ВБ<sub>2</sub>, ВБ<sub>3</sub> по сигналам с блока управления БУ проводят вычисления поправочного коэффициента  $C$ , параметров настройки регулятора и коэффициентов режекторных фильтров. Значения коэффициента  $C$  и фазового сдвига  $\Phi_0$  в объекте управления в момент расчета требуемых настроек хранятся в запоминающих устройствах ЗУ<sub>1</sub>, ЗУ<sub>2</sub>. Коммутация входных сигналов соответствующих блоков на разных этапах работы системы происходит по командам БУ с помощью переключателей П<sub>3</sub>, П<sub>4</sub>, П<sub>5</sub>, П<sub>6</sub>. Последнюю операцию целесообразно осуществлять с применением многофункциональных логических модулей [7].

Процесс адаптации состоит из двух этапов: двухчастотного, в конце которого вычисляются требуемые настройки и коэффициент  $C$ ; основного, в котором процесс самонастройки происходит уже на одной повышенной частоте пробных колебаний с одним режекторным фильтром РФО в контуре обратной связи. Как раз определение критической частоты и соответствующей ей амплитуды колебаний для блока ОУ+РФО позволяет работать с нужными параметрами настроек на втором этапе. Осуществляется это с помощью поправочного коэффициента  $C$ , устанавливающего связь между требуемыми по методу Циглера–Никольса настройками на первом и втором этапах работы системы [12].

На рис. 5.8, 5.9 представлены переходные процессы рассматриваемой системы по каналу задающего и возмущающего воздействия соответственно. После процесса адаптации значительно уменьшились колебания, максимальная динамическая ошибка и время регулирования.

На рис. 5.10 представлены сравнительные динамические характеристики созданного адаптивного регулятора с настройкой по методу Циглера–Никольса (кривая 1), традиционный ПИД-регулятор, настраиваемый вручную на каждом этапе работы адаптивного регулятора (кривая 2), традиционный ПИД-регулятор с фиксированной настройкой по методу РАФЧХ на исходное значение постоянной времени объекта управления (кривая 3).

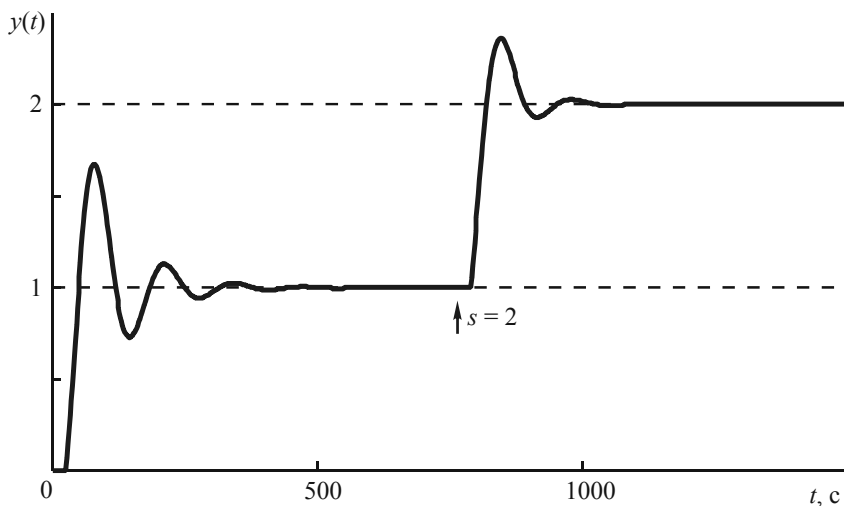


Рис. 5.8. Переходный процесс в адаптивной системе автоматического регулирования по каналу задающего воздействия

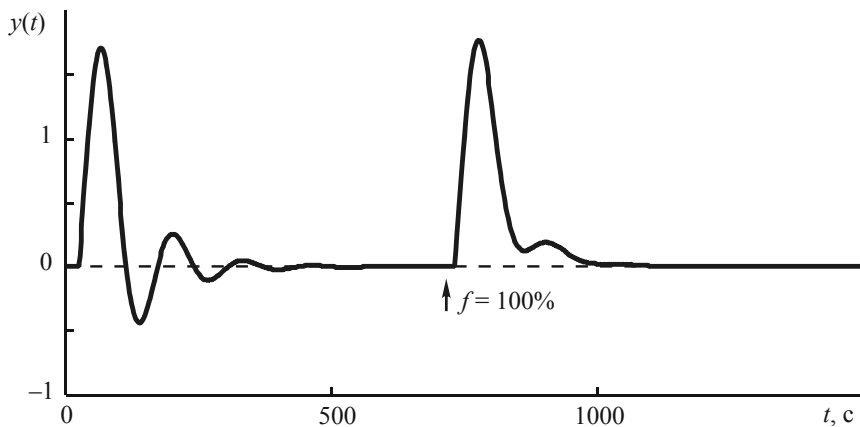


Рис. 5.9. Переходный процесс в адаптивной системе автоматического регулирования по каналу возмущения, идущего со стороны регулирующего органа

Для функционирования системы достаточно в разработанный адаптивный ПИД-регулятор ввести настройки, обеспечивающие устойчи-

вость замкнутой системы, поэтому воспользуемся найденными в предыдущем параграфе параметрами настройки ПИД-регулятора и примем коэффициент передачи  $k_p = 0,47$ , постоянную времени удвоения  $T_y = 52,6$  с и постоянную времени предварения  $T_{np} = 0$ .

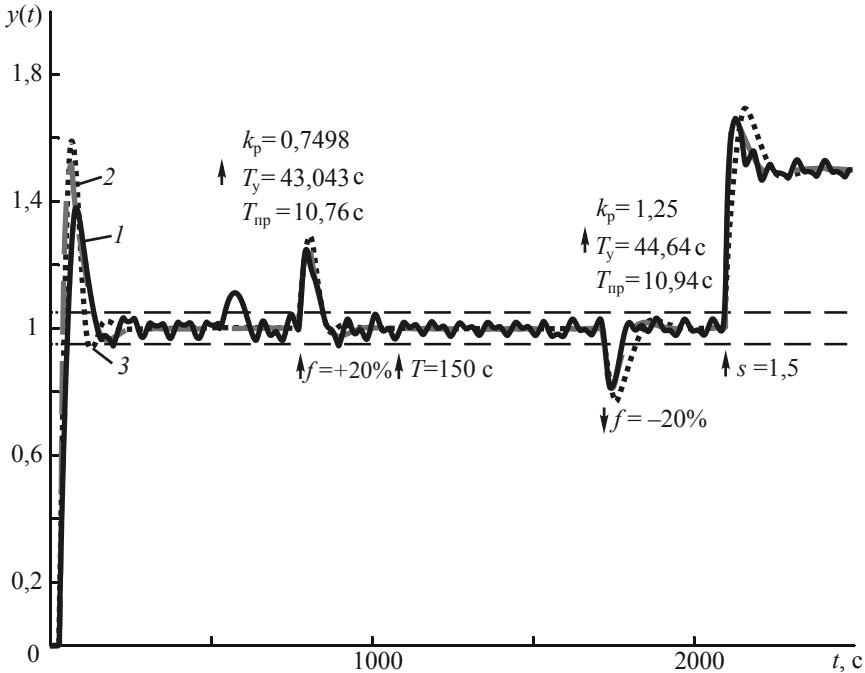


Рис. 5.10. Динамические характеристики: 1 – адаптивный ПИД-регулятор; 2 – классический ПИД-регулятор, настраиваемый вручную на каждом этапе работы адаптивного регулятора; 3 – ПИД-регулятор, настроенный на первоначальное значение постоянной времени объекта

Тогда передаточная функция регулятора на начальный момент регулирования примет вид

$$W_{\text{пер}}(p) = k_p \left( 1 + \frac{1}{T_y p} + T_{np} p \right) = 0,47 + \frac{0,47}{52,6 p}$$

Параметрический синтез ПИД-регулятора по методу РАФЧХ для объекта, описываемого передаточной функцией (5.4) при заданных  $\psi = 0,95$  и втором интегральном критерии качества, дал следующие результаты:  $k_p = 0,727$ ;  $T_y = 51,929$  с;  $T_{пр} = 5,77$  с. Синтез методом Циглера–Никольса позволил определить значения:  $k_p = 0,72$ ;  $T_y = 42,35$  с;  $T_{пр} = 5,57$  с.

Интегральные критерии качества по обоим методам принимают почти одинаковые значения:

$$I_2^{\text{РАФЧХ}} = \int_0^{\infty} |\varepsilon(t)| dt = 59,59; \quad I_2^{\text{Ц-Н}} = \int_0^{\infty} |\varepsilon(t)| dt = 62,43; \quad (5.7)$$

$$I_3^{\text{РАФЧХ}} = \int_0^{\infty} \varepsilon(t)^2 dt = 41; \quad I_3^{\text{Ц-Н}} = \int_0^{\infty} \varepsilon(t)^2 dt = 39,71. \quad (5.8)$$

Установлено, что переходный процесс рассматриваемой системы автоматического регулирования с регулятором, рассчитанным по методу Циглера–Никольса, обладает близкими к минимуму интегральными оценками качества (5.7), (5.8), однако во временной области переходный процесс имеет меньшее перерегулирование и меньшую максимальную динамическую ошибку по сравнению с процессами, настроенными по аналитическому методу РАФЧХ, что, несомненно, предпочтительней для технологических процессов.

Адаптивный ПИД-регулятор с автоматической подстройкой коэффициентов имеет близкие динамические характеристики с ПИД-регулятором, настроенным вручную, на каждом этапе адаптации, по методу Циглера–Никольса, и заметный выигрыш в качестве регулирования по сравнению с ПИД-регулятором, имеющим фиксированную настройку при параметрическом возмущении.

В табл. 5.1 приведены численные значения параметров настройки регуляторов, а в табл. 5.2 – показателей качества переходных процессов. Как видно, разработанный адаптивный регулятор характеризуется достаточно высокой скоростью коррекции параметров при сохранении настроек в окрестности их расчетных значений по методу Циглера–Никольса.



Таблица 5.1

**Параметры настройки ПИД-регулятора, полученные автоматически и аналитическим способом**

Постоянная времени $T, c$	Значение параметров настройки адаптивного ПИД-регулятора			Значение параметров настройки ПИД-регулятора, рассчитанных вручную на каждом этапе адаптации		
	$k_p$	$T_v, c$	$T_{np}, c$	$k_p$	$T_v, c$	$T_{np}, c$
97,7	0,7498	43,043	10,76	0,72	42,4	10,76
150	1,2521	44,641	10,94	1,07	44,58	11,05
50	0,4153	40,1816	6,02	0,4027	40,116	6,046

Таблица 5.2

**Оценки показателей качества переходного процесса системы автоматического регулирования с ПИД-регуляторами, настроенными разными способами**

Показатели качества переходного процесса	Адаптивный ПИД-регулятор			ПИД-регулятор с ручной настройкой на каждом этапе адаптации			ПИД-регулятор с фиксированной настройкой		
	$T, c$								
	50	97,7	150	50	97,7	150	50	97,7	150
$\psi$	0,94	0,99	0,99	0,95	0,99	0,98	0,06	0,95	0,99
$\sigma, \%$	56,3	53,9	68,9	53,1	51,7	55,3	115	59,2	45
$A_{max}$	0,56	0,54	0,69	0,53	0,51	0,55	1,15	0,59	0,45
$t_p, c$	128	134	133	127	135	138	$4 \cdot 10^3$	135	221

Таким образом, разработанный контроллер может широко использоваться при автоматизации многих технологических процессов, и прежде всего таких, как теплоэнергетический (регулирование уровня воды в барабане котлоагрегата), металлургический (регулирование температуры и скорости горячего дутья доменной печи), химический (регулирование температуры рабочего тела на выходе печи пиролиза) и др. К недостаткам регулятора следует отнести возможность адаптации только в установившемся состоянии системы и работы алгоритма только совместно с объектами, обладающими изменением фазового сдвига свыше  $-180^\circ$ . Проведенные исследования говорят о перспективном использовании регулятора при управлении нестационарными технологическими объектами.

#### 5.4. СИСТЕМЫ АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ СО СТРУКТУРНОЙ АДАПТАЦИЕЙ

Принцип действия таких систем связан с изменением структуры регулятора и соответственно закона адаптации [21]. При этом изменение той или иной структуры может быть сведено к изменению соответствующих параметров. Так, структура ПИД-регулятора с передаточной функцией вида

$$W_p(p) = K_p \left( 1 + \frac{1}{T_y p} + T_{np} p \right) \quad (5.9)$$

в зависимости от выбора параметров может быть модифицирована в структуру ПИ- ( $T_{np} = 0$  с) и даже П-регулятора ( $T_y \rightarrow \infty$  с,  $T_{np} = 0$  с). Другой способ реализации – использование перестраиваемых структур. Принцип построения системы со структурной адаптацией показан на рис. 5.11. Он заключается в выборе  $R_i$ -регулятора из  $n$  имеющихся в зависимости от состояния процесса. При этом проблема идентификации сводится к классификации, заключающейся в установлении соотношения между характеристиками входного и выходного сигналов основной системы и структурой регулятора.

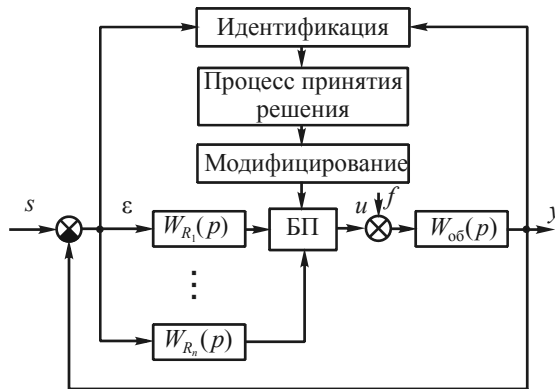


Рис. 5.11. Принцип построения структурно-адаптивной системы: БП – блок переключения

Приведем пример динамики работы такой системы. Пусть система состоит из объекта регулирования, описываемого передаточной функцией вида (5.4), и ПИ-регулятора с передаточной функцией

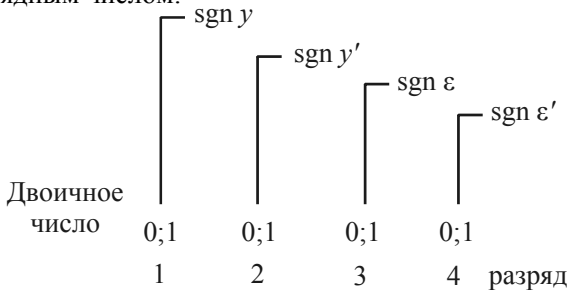
$$W_p(p) = K_p \left( 1 + \frac{1}{T_{и} p} \right) = \beta_0 + \frac{\beta_1}{p}, \tag{5.10}$$

где параметры  $\beta_1, \beta_0$  имеют по четыре значения.

Для выбора различных комбинаций этих значений применяется следующее двоичное кодирование знаков сигналов  $y, y', \varepsilon, \varepsilon'$ :

$$\text{sgn } y \begin{cases} > 0 \triangleq 1, \\ < 0 \triangleq 0, \end{cases} \quad \text{sgn } y' \begin{cases} > 0 \triangleq 1, \\ < 0 \triangleq 0, \end{cases} \quad \text{sgn } \varepsilon \begin{cases} > 0 \triangleq 1, \\ < 0 \triangleq 0, \end{cases} \quad \text{sgn } \varepsilon' \begin{cases} > 0 \triangleq 1, \\ < 0 \triangleq 0. \end{cases} \tag{5.11}$$

В результате состояние системы может быть описано следующим четырехразрядным числом:



С помощью этого числа возможно описать  $2^4 = 16$  комбинаций параметров  $[\beta_{1i}, \beta_{0i}]$ . Структурная схема системы автоматического регулирования, реализующей описанный принцип, приведена на рис. 5.12.

На рис. 5.13 представлен процесс выбора параметров настройки регулятором в период его функционирования. В момент времени  $t = 250$  с на объект действует параметрическое возмущение, в результате которого постоянная времени  $T = 150$  с.

Динамика изменения переходного процесса при изменении постоянной времени объекта управления системы автоматического регулирования, функционирующей совместно с регулятором со структурной адаптацией, представлена на рис. 5.14, а оценка показателей качества – в табл. 5.3.

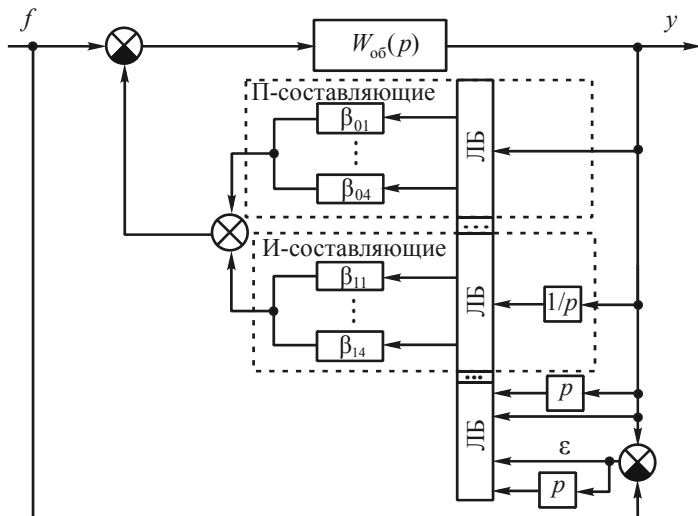


Рис. 5.12. Структурная схема АСР со структурной адаптацией ПИ-регулятора: ЛБ – логический блок

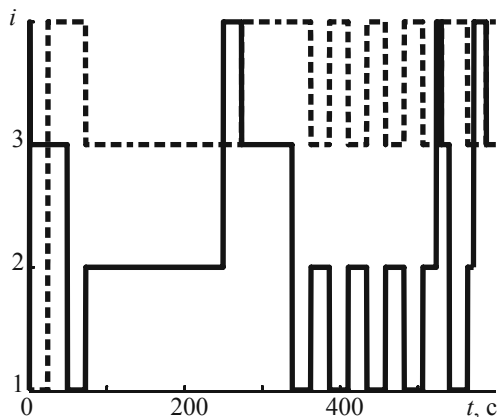


Рис. 5.13. Процесс автоматического выбора параметров настройки регулятора: сплошная линия –  $\beta_{0i}$ ; пунктирная –  $\beta_{1i}$

При сравнении табл. 5.2 и 5.3 установлено, что регулятор со структурной адаптацией позволяет получить лучшие показатели качества

переходного процесса, чем классические регуляторы. На первый взгляд, регулятор обладает более низкими показателями качества, чем регулятор с параметрической адаптацией по методу Циглера–Никольса, однако, как уже упоминалось в разделе 5.2, последний способен подстраивать свои параметры только в установившемся состоянии системы, а регулятор со структурной адаптацией приспосабливается к объекту непосредственно при протекании переходного процесса системы автоматического регулирования.

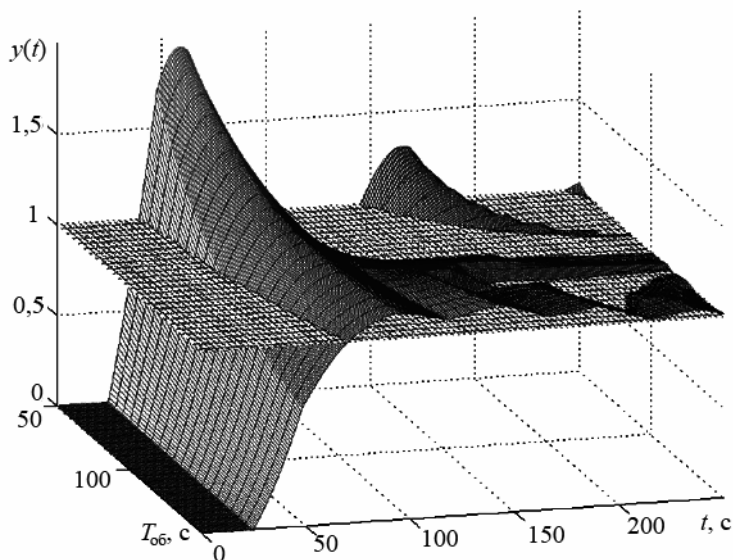


Рис. 5.14. Динамика изменения переходного процесса в рассматриваемой системе автоматического регулирования при изменении постоянной времени объекта

Таким образом, разработанный регулятор позволяет расширить область улучшенных показателей качества системы автоматического регулирования при действии параметрических возмущений. Исследования показали, что системы со структурной адаптацией могут применяться для объектов, динамические характеристики которых претерпевают существенные изменения за короткое время, исходя из чего получают существенное преимущество по сравнению с системами автоматического

регулирования с параметрической адаптацией, однако в ущерб обобщенным качественным показателям.

Таблица 5.3

**Оценки показателей качества АСР с регулятором со структурной адаптацией при действии параметрических возмущений**

Показатели качества	Постоянная времени объекта $T$ , с		
	50	97,7	150
$\psi$	0,67	0,99	0,9
$A_{\max}$	0,93	0,33	0,14
$\sigma$ , %	93,26	33,3	14,9
$t_p$ , с	260	115	205

От систем со структурной адаптацией следует отличать системы регулирования с перестраиваемой структурой, о которых речь пойдет далее. Специфика последних заключается в ориентации только на улучшение динамики регулирования. Однако следует отметить, что в некоторых случаях принципы построения систем со структурной адаптацией и перестраиваемой структурой совпадают.

## 5.5. СИСТЕМЫ АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ С ПЕРЕСТРАИВАЕМОЙ СТРУКТУРОЙ

### 5.5.1. Формирование логического закона управления

В общем случае закон управления для систем с перестраиваемой структурой (СПС) выглядит следующим образом:

$$u = \Psi \cdot \varepsilon, \quad (5.12)$$

где  $\varepsilon$  – сигнал ошибки регулирования;  $\Psi$  – логический закон, принимающий определенное значение в зависимости от типа объекта управления и его текущего состояния.

Устройство управления, реализующее (5.12), можно представить в виде заданного класса динамических корректирующих звеньев  $A_i$  ( $i = 1, 2, \dots, n$ ), логического устройства  $M$  и вспомогательных блоков формирования входных аргументов  $X$  и настройки структуры  $Z$ . Последние блоки служат для выработки логических переменных, зависящих от состояния системы в фазовом пространстве и настройки

логического устройства  $M$  на требуемый логический закон управления (рис. 5.15).

В системах с переменной структурой требуемые динамические свойства замкнутой системы обеспечиваются надлежащим выбором поверхности переключения, вид которой задается при синтезе [14]. В нашем случае поверхность переключения реализуется квазиизотропной средой, построенной на основе МЛМ двух типов (см. 2.4.1 и 2.4.2 –  $L$ - и  $V$ -структуры) [19, 20].

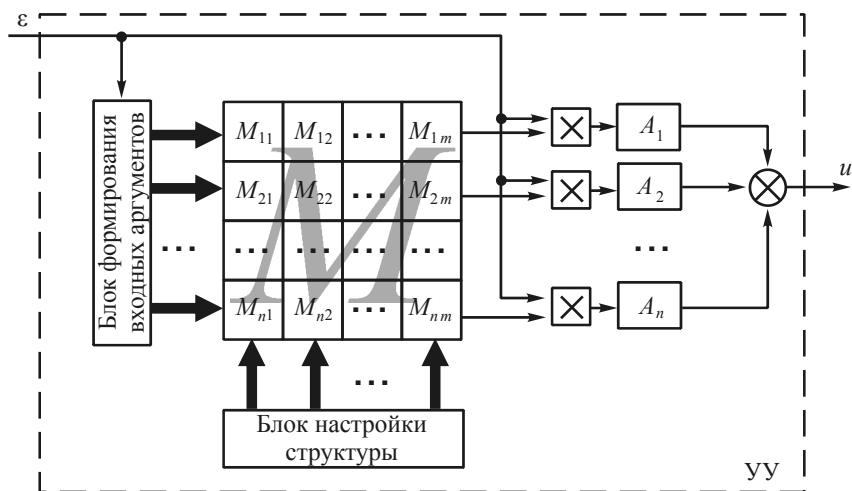


Рис. 5.15. Структурная схема устройства управления

Квазиизотропная среда представляет собой три грани пространственной шестинаправленной структуры (рис. 5.16) [11]. Центральная (первая) грань является линейной структурой из  $L$ - и  $V$ -ячеек с режимными боковыми входами ( $y_i, i=1,2,3$ ) и поданными аргументами  $X_j$  ( $j=1,2, \dots, n, n=6$  – число аргументов) системы, описывающими логический закон управления, на функциональные входы  $x$  ячеек. Вторая и третья грани представляют собой также линейные структуры из  $V$ -ячеек с режимными входами. На функциональные входы  $x$  поданы

промежуточные значения после соседних ячеек центральной грани первого и второго каналов соответственно для второй и третьей граней.

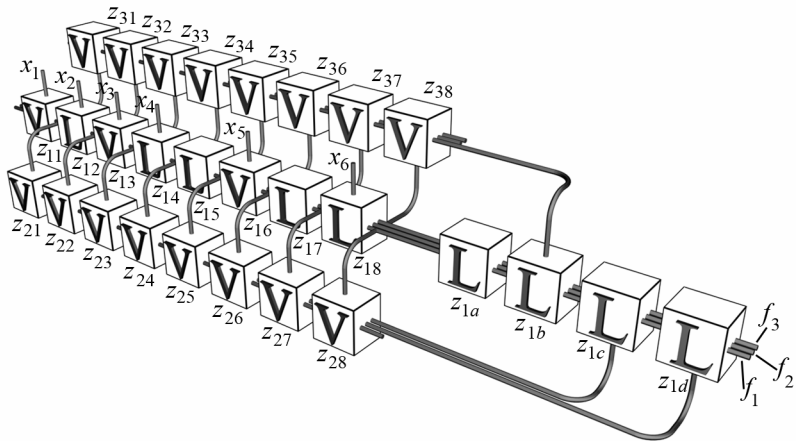


Рис. 5.16. Квазиизотропная среда

Для различных типов технологических объектов необходимо использовать разные законы управления. Так, например, для объектов с постоянными параметрами, в случае использования системы с переменной структурой, закон управления описывается выражением

$$\Psi = \begin{cases} \alpha \text{ при } \varepsilon s_1 > 0, \\ \beta \text{ при } \varepsilon s_1 < 0, \end{cases} \quad (5.13)$$

а для объекта с постоянными параметрами, обладающего транспортным запаздыванием, имеет вид [5]:

$$\Psi = \begin{cases} \alpha \text{ при } \varepsilon s_1 > 0, \\ \beta \text{ при } s_2 s_1 < 0, \\ 0 \text{ при } \varepsilon s_2 < 0. \end{cases} \quad (5.14)$$

Здесь  $\alpha$  и  $\beta$  – коэффициенты передачи первой и второй линейных структур соответственно ( $A_1 = \alpha$ ,  $A_1 = \beta$ , см. рис. 5.15);  $s_i = \text{sgn}(\varepsilon' + c_i \varepsilon)$ ,  $i = 1, 2$ , – информация о знаке линейной комбинации ошибки и ее



производной, характеризующая положение системы в фазовом пространстве относительно прямых переключений;  $c_i$  – коэффициент наклона прямых. Как видно, для функционирования системы не требуется точного значения производной от сигнала ошибки, а достаточно лишь информации о знаке ее линейной комбинации с величиной ошибки, которую можно получить сравнительно простыми техническими средствами, например, описанными в [5, 6].

В булевой форме стратегию выбора динамического закона управления для объектов без транспортного запаздывания можно представить следующим образом:

$$\begin{cases} f_1 = X_1, \\ f_2 = X_2, \\ f_3 = 0, \end{cases} \quad (5.15)$$

а для объектов с запаздыванием – в виде

$$\begin{cases} f_1 = (X_1 \vee X_4)X_2X_3, \\ f_2 = X_1X_3X_5, \\ f_3 = X_1X_6(X_2 \vee X_5), \end{cases} \quad (5.16)$$

где  $X_j$  ( $j=1,2, \dots, 6$ ) – логические аргументы, отражающие поведение рассматриваемой системы в фазовом пространстве, вырабатываемые блоком формирования входных аргументов (см. рис. 5.15).

Система булевых формул (5.15) или (5.16) формируется логическим устройством  $M$  (рис. 5.15), для этого каждый вход  $Z$   $ij$ -й ячейки квазиизотропной среды (см. рис. 5.16) необходимо настроить согласно табл. 5.4 или 5.5 соответственно.

Таблица 5.4

**Настроечные коды квазиизотропной среды для реализации алгоритма управления объектом без транспортного запаздывания**

$i$	$j$											
	1	2	3	4	5	6	7	8	$a$	$b$	$c$	$d$
1	0011	001	0110	010	010	0110	010	010	010	010	010	010
2	0101	1111	0110	0110	0110	0110	0110	0110	—	—	—	—
3	0101	1111	0110	0110	0110	0110	0110	0110	—	—	—	—

Таблица 5.5

**Настроечные коды квазиизотропной среды для реализации алгоритма управления объектом с транспортным запаздыванием**

<i>i</i>	<i>j</i>											
	1	2	3	4	5	6	7	8	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
1	1001	111	0101	010	100	1001	000	110	000	010	000	010
2	0101	0110	0110	0110	0110	0100	0110	0101	—	—	—	—
3	0110	1001	0110	0110	0110	0110	0110	0110	—	—	—	—

Следовательно, два закона управления можно реализовать на одной вычислительной среде, используя в качестве ее элементарных узловых элементов МЛМ. Количество реализуемых алгоритмов представленной квазиизотропной среды не ограничивается только двумя, которые мы используем. Оно зависит от числа фиксированных структур, реализуемых логическим устройством  $M$ , которое, в свою очередь, зависит от обобщенного настроечного кода логического устройства и составляет  $2^r$  структур, где  $r$  – разрядность этого кода. В нашем случае нетрудно подсчитать (по табл. 5.4 или 5.5), что  $r = 103$ . Кроме того, при различных фиксированных структурах, а значит, и различных настроечных кодах устройства  $M$  последнее может реализовывать одни и те же алгоритмы, поэтому их число  $m \leq 2^r$ .

Приведем методику синтеза логического устройства  $M$  на уже разработанных или вновь разрабатываемых МЛМ [17].

1. Записываются алгоритмы управления, удовлетворяющие требованиям, предъявляемым к системе.
2. Выбранные алгоритмы представляются в булевой форме.
3. Выбирается один алгоритм, полученный в п. 2, покрывающий больший класс булевых формул.
4. Выбираются множество МЛМ, на которых будет реализовываться алгоритм (это множество может быть и единичным – случай изотропной среды), и вид комбинационных связей между соседними МЛМ.
5. Из набора фиксированных структур, реализуемых выбранными МЛМ, выделяются те, которые обеспечивают реализацию соответствующей булевой формулы в логическом законе управления из п. 3.
6. На основе выделенного подмножества автоматных отображений выбранных МЛМ синтезируется изотропная или квазиизотропная среда, обеспечивающая реализацию логического закона управления (т.е. вы-

полняется процесс последовательного вложения каждого аргумента логического закона в квазиизотропную среду). Результатом являются синтезированная квазиизотропная среда и коды настройки каждого МЛМ среды на реализацию логического закона (см. п. 3; рис. 5.16 и табл. 5.5).

7. Настраиваем квазиизотропную среду на реализацию следующего алгоритма из п. 2, используя ее в качестве «каркаса» и изменяя автоматные отображения каждой МЛМ, путем подачи на него определенного настроечного кода таким образом, что аргументы реализуемого алгоритма последовательно вкладываются в квазиизотропную среду. Результатом является код настройки квазиизотропной среды на соответствующий алгоритм (см. табл. 5.4).

8. Повторяем п. 7  $k - 2$  раз, где  $k$  – число алгоритмов из п. 2.

Таким образом, количество входов  $X$  логического устройства  $M$  зависит только от сложности реализуемого алгоритма, удовлетворяющего требованиям системы, т.е. от количества аргументов, использующихся при его описании в булевой форме. Количество же настроечных входов  $Z$  зависит от выбранного варианта комбинационных связей между МЛМ, а значит, от формы квазиизотропной среды, количества МЛМ в среде и количества настроечных входов каждого МЛМ.

### 5.5.2. Пример синтеза системы без запаздывания в контуре управления

В качестве иллюстрации описанного подхода рассмотрим объект с передаточной функцией

$$W_{об}(p) = \frac{k}{Tp + 1} = \frac{6}{97,7p + 1}. \quad (5.17)$$

Обеспечим в системе переходный процесс с заданным качеством (время регулирования  $t_p \leq 15$  с, перерегулирование  $\sigma \leq 20\%$ ). Поскольку объект регулирования и линейные структуры регулятора являются статическими звеньями, то при конечных коэффициентах последнего в замкнутой системе будет возникать статическая ошибка. Для сведения ее к нулю придадим системе свойство астатизма, введя в контур управления интегрирующее звено с передаточной функцией вида

$$W_{и}(p) = \frac{1}{T_{и}p}.$$

Передаточная функция приведенного объекта в таком случае принимает вид

$$W_{об}^*(p) = \frac{k}{T_{и}T p^2 + T_{и}p}.$$

Из [28] известно, что для организации скользящего режима в системе второго порядка с регулятором, имеющим две линейные структуры, достаточно выбрать коэффициенты передачи  $\alpha$  и  $\beta$  выражения (5.13), пользуясь соотношением вида

$$\begin{cases} \alpha > \frac{-c_1^2 + c_1 \cdot a_2 - a_1}{a_1}, \\ \beta < \frac{-c_1^2 + c_1 \cdot a_2 - a_1}{a_1}, \end{cases}$$

где  $a_1 = k/(T_{и}T)$ ;  $a_2 = 1/T$ .

Из вышеприведенного находим, что  $\beta < -1,62 < \alpha$  (при  $T_{и} = 1$  и  $c_1 = 0,2$ ). Фазовые портреты линейных структур при  $\alpha$  и  $\beta$  представлены на рис. 5.17, *a* для начальных условий  $x_1^0 \in [-1; -0,5; \dots; 1]$ ,  $x_2^0 \in [-1; -0,5; \dots; 1]$ . Как видно, фазовые траектории в области переключения (IV четверть) направлены в разные стороны, что позволяет организовать скользящий режим.

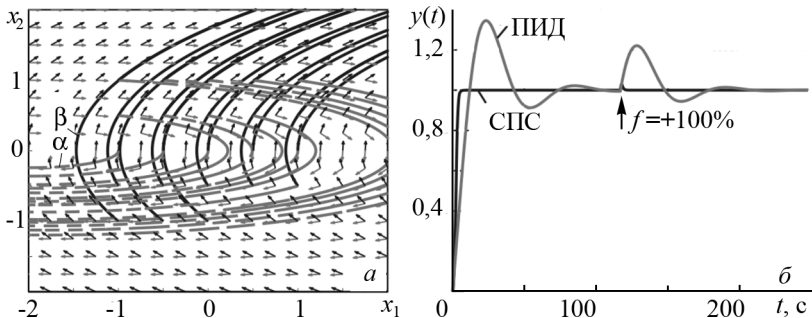


Рис. 5.17. Результаты моделирования: *a* – фазовые портреты и поля направлений для линейных структур; *б* – динамические характеристики автоматической системы регулирования с разными типами регуляторов

На рис. 5.17, б приведены сравнительные переходные характеристики для системы с ПИД- и СПС-регулятором, из которых видно, что последний значительно лучше справляется с двумя основными задачами системы автоматического регулирования – программным управлением и стабилизацией технологического параметра (в данном случае при координатном возмущении  $f$ ), по сравнению с традиционными системами. При действии параметрических возмущений на объект регулирования, в результате которых происходит изменение постоянной времени объекта, СПС-регулятор позволяет существенно расширить улучшенные показатели качества переходного процесса (рис. 5.18, а). Количественные оценки показателей качества переходного процесса для постоянной времени  $T \in \{50; 97,7; 150\}$  с приведены в табл. 5.6.

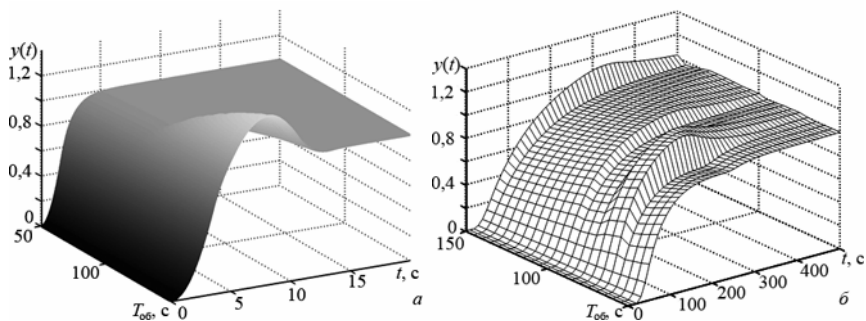


Рис. 5.18. Динамика изменения переходного процесса в системе при действии параметрического возмущения: а – для объекта без запаздывания; б – для объекта, обладающего транспортным запаздыванием

Таблица 5.6

**Оценки показателей качества переходного процесса в АСП с СПС- и ПИД-регуляторами**

Показатели качества	Тип регулятора					
	СПС			ПИД		
	$T, c$					
	50	97,7	150	50	97,7	150
$\psi$	1	1	0,999	0,98	0,96	0,9
$\sigma, \%$	0	0	30	24	30	35
$A_{max}$	0	0	0,3	0,24	0,3	0,35
$t_{ps}, c$	3,8	4,2	11,2	38	70	86

Таким образом, применение принципов перестраиваемости структур позволяет решить задачу высококачественного управления без ввода производной в канал управления и наделить систему автоматического управления свойством инвариантности к возмущающим воздействиям. Данные типы регуляторов могут широко применяться совместно с различными типами объектов, в частности для управления электродвигателями, электронасосами и т.п.

### 5.5.3. Пример синтеза системы с запаздыванием в контуре управления

При построении систем автоматического регулирования технологическими процессами весьма часто оказывается, что контур управления обладает чистым запаздыванием. Во многих случаях динамические системы высоких порядков с достаточной степенью точности аппроксимируются системой первого или второго порядка с запаздыванием [5, 6].

В качестве примера рассмотрим объект регулирования с передаточной функцией

$$W_{об}(p) = \frac{k \exp(-\tau p)}{Tp + 1} = \frac{6 \exp(-23,4p)}{97,7p + 1}.$$

Обеспечим переходный процесс в системе продолжительностью, близкой величине  $t_p = 150$  с, и  $\sigma \leq 15\%$  за счет сочетания конечного числа структур, отличных друг от друга лишь уровнем воздействия по координате ошибки. Как и в предыдущем примере, придадим всей системе свойство астатизма, тогда для приведенного объекта управляющее воздействие  $u$  необходимо вырабатывать с учетом системы (5.14). Найдем параметр  $\alpha = 0,0074$  из уравнения гармонического баланса, тогда величину  $\beta = -0,0006$  выберем из условия, что система в момент переключения на вторую линейную структуру должна быть охвачена положительной обратной связью и ее фазовая траектория направлена в направлении, противоположном движению траектории первой линейной структуры (при коэффициенте  $\alpha$ ).

Система автоматического регулирования должна за время протекания переходного процесса трижды сменить структуру: сначала объект должен быть охвачен отрицательной обратной связью, затем на интервале продолжительностью  $(t_p - \tau) - t_1$ , причем  $t_1 < (t_p - \tau)$ , – положительной обратной связью, и, наконец, с момента  $t_p - \tau$  система становится разомкнутой. Чем величина  $t_1$  ближе к  $t_p - \tau$ , тем меньше переходный процесс будет отличаться от первой полуволны переходного процесса системы с линейной структурой при коэффициенте  $\alpha$  [5].

Для построения системы с перестраиваемой структурой, в которой реализуется описанный переходный процесс, необходимо найти закон изменения структуры в функции координат системы. При произвольном постоянном воздействии строим переходный процесс, выбрав моменты переключения  $t_1$  и  $t_p - \tau$ .

В силу того что каждая из рассматриваемых структур является линейной, для любого постоянного воздействия отношение  $(d\varepsilon/dt)/\varepsilon$  для различных процессов в моменты переключения будет неизменным. Следовательно, коэффициенты наклона прямых переключений (при  $t_1 = 120$  с) находятся из графиков ошибки регулирования и ее производной по формулам

$$c_1 = -\frac{d\varepsilon(t_1)/dt}{\varepsilon(t_1)} = 0,0028, \quad c_2 = -\frac{d\varepsilon(t_p - \tau)/dt}{\varepsilon(t_p - \tau)} = 0,017.$$

При этом квазиизотропную среду настроим в соответствии с табл. 5.5.

В результате СПС-регулятор, функционируя совместно с объектами, обладающими транспортным запаздыванием, позволяет добиться переходных процессов (см. рис. 5.18, б) практически без перерегулирования с быстроедействием, близким к системам, которые функционируют совместно с ПИД-регулятором. Количественные оценки показателей качества переходного процесса замкнутой системы автоматического регулирования с СПС-регулятором, при различных постоянных времени объекта управления, приведены в табл. 5.7.

Таблица 5.7

## Оценки показателей качества АСР с СПС-регулятором

Показатели качества	Постоянная времени объекта $T$ , с		
	50	97,7	150
$\psi$	1	1	1
$A_{\max}$	0	0,023	0,04
$\sigma$ , %	0	2,3	4
$t_p$ , с	120	152	237

Таким образом, используя одну и ту же квазиоднородную структуру регулятора, можно добиться повышения качества управления для объектов как с транспортным запаздыванием, так и без него путем подачи определенного настроечного кода на каждую ячейку среды, изменяя тем самым внутреннюю структуру и закон функционирования регулятора. Разработанный регулятор может применяться на разнообразных технологических объектах управления, например для управления температурой перегретого пара парогенератора.

### 5.6. ИНТЕГРАЛЬНЫЙ РЕГУЛЯТОР С ПЕРЕСТРАИВАЕМОЙ СТРУКТУРОЙ

При управлении технологическими процессами с взаимосвязанными параметрами лучшие технологические показатели достигаются при таких изменениях регулирующих воздействий, которые необходимы, согласно статическим характеристикам объектов, только для компенсации возмущений [10].

Интегральный регулятор с перестраиваемой структурой (прил. 3) обеспечивает управление минимальными изменениями регулирующих воздействий инерционными объектами. Регулятор состоит из следящего контура (рис. 5.19), вырабатывающего вспомогательную координату  $s$ , и логического устройства, которые в зависимости от сочетания знаков вспомогательных координат  $\varepsilon^*$ ,  $s$ ,  $\sigma$  формируют логический закон управления  $\chi$  с целью изменения структуры системы. При определенных сочетаниях знаков координат  $\varepsilon^*$ ,  $s$ ,  $\sigma$  канал управления открывается для передачи сигнала ошибки  $\varepsilon$  на интегратор.



Дифференциальное уравнение регулятора имеет вид

$$T_s \mu' = \rho,$$

где

$$\rho = \begin{cases} 1 & \text{при } \chi > \frac{e}{2}, \\ 0 & \text{при } -\frac{e}{2} \leq \chi \leq \frac{e}{2}, \\ -1 & \text{при } \chi < -\frac{e}{2}. \end{cases}$$

Здесь  $e$  – зона нечувствительности релейного элемента;

$$v = U_1(\varepsilon^*, s)\varepsilon,$$

где

$$U_1(\varepsilon^*, s) = \begin{cases} 0 & \text{при } \varepsilon^* \cdot s \leq 0, \\ 1 & \text{при } \varepsilon^* \cdot s > 0. \end{cases}$$

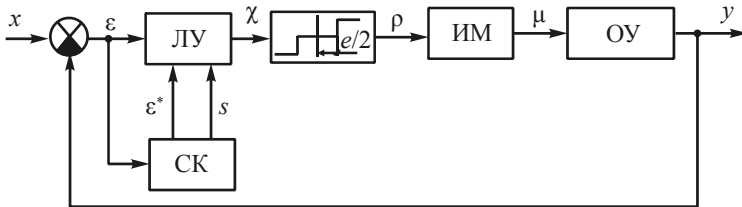


Рис. 5.19. Обобщенная структурная схема системы с интегральным регулятором с переменной структурой: ЛУ – логическое устройство; СК – следящий контур; ИМ – исполнительный механизм; ОУ – объект управления

Здесь  $\varepsilon^* = \text{sgn } \varepsilon$  – знак сигнала ошибки регулирования;  $s = \varepsilon' + c \cdot \varepsilon$  – прямая переключения;  $c$  – коэффициент наклона прямой;

$$\chi = U_2(\sigma)v,$$

где

$$U_2(\sigma) = \begin{cases} 1 & \text{при } \sigma = 0, \\ 0 & \text{при } \sigma = 1, \end{cases} \quad \sigma = \begin{cases} 1 & \text{при } \lambda > e, \\ 0 & \text{при } \lambda \leq e; \end{cases}$$

$$\lambda = \alpha + \beta, \quad \alpha = U_4(s, \xi)\psi;$$

$$U_4(s, \xi) = \begin{cases} 1 & \text{при } s = 0, \xi = 1, \\ 0 & \text{при } \begin{cases} s = 0, \xi = 0, \\ s \neq 0, \xi = 0, \end{cases} \end{cases} \quad \xi = \begin{cases} 1 & \text{при } \vartheta > e, \\ 0 & \text{при } \vartheta \leq e. \end{cases}$$

Здесь  $\vartheta$  – решение линейного дифференциального уравнения

$$T_1 \vartheta' + \vartheta = \gamma,$$

$$\gamma = U_3(v, \sigma)V_1, \quad \text{где } V_1 = \text{const};$$

$$U_3(v, \sigma) = \begin{cases} 0 & \text{при } \begin{cases} v = 0, \sigma = 0, \\ v \neq 0, \sigma = 1, \end{cases} \\ 1 & \text{при } v \neq 0, \sigma = 0, \end{cases} \quad \beta = U_5(\sigma)\psi, \quad U_5(\sigma) = \begin{cases} 1 & \text{при } \sigma = 1, \\ 0 & \text{при } \sigma = 0. \end{cases}$$

Здесь  $\psi$  – решение нелинейного дифференциального уравнения

$$T(\sigma)\psi' + \psi = v,$$

$$T(\sigma) = \begin{cases} T_0 & \text{при } \sigma = 0, \\ T_1 & \text{при } \sigma = 1, \end{cases} \quad v = U_2(\sigma)V_2, \quad \text{где } V_2 = \text{const}.$$

В регуляторе использованы три МЛМ  $L$ -структуры для формирования выходного логического закона управления, представленного функцией  $F$ , зависящей от четырех аргументов и настроенного кода  $Z = 001$ :

$$F = f(\bar{A}, A, \bar{B}, B, Z).$$

На рис. 5.20 приведен пример обработки задающего воздействия рассматриваемой системой. В момент времени  $t = 25$  с на объект действует внешнее координатное возмущающее воздействие  $f$ , равное +100% от задающего воздействия.

При действии параметрического возмущения, в результате которого происходит увеличение постоянной времени объекта управления, переходный процесс в рассматриваемой замкнутой системе автоматического

регулирования приобретает большее перерегулирование и увеличивается время регулирования (табл. 5.8, рис. 5.21).

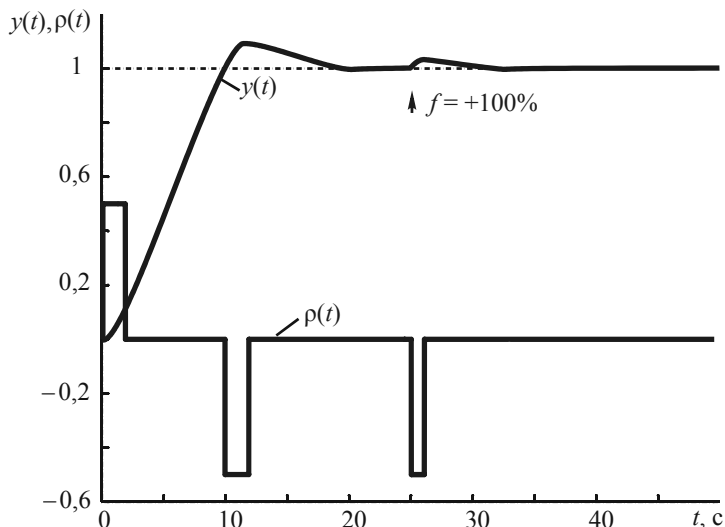


Рис. 5.20. Переходный процесс в системе автоматического регулирования с интегральным регулятором с перестраиваемой структурой по каналу задающего воздействия

Т а б л и ц а 5.8

**Оценки показателей качества переходного процесса рассматриваемой АСР при параметрических возмущениях**

Показатели качества	Постоянная времени объекта $T$ , с		
	50	97,7	150
$\psi$	1	1	1
$A_{\max}$	0,04	0,09	0,13
$\sigma$ , %	4,8	9,5	13,8
$t_p$ , с	8,9	15,3	18,5

Таким образом, за счет модификации закона управления с перестраиваемой структурой и ухода от ввода непосредственно разрывной функции от ошибки в канал управления разработанный регулятор имеет практическую направленность на использование в системах автоматического регулирования совместно с объектами, имеющими электроме-

ханические и механические узлы в канале управления. При этом сохраняются все основные достоинства систем с перестраиваемой структурой.

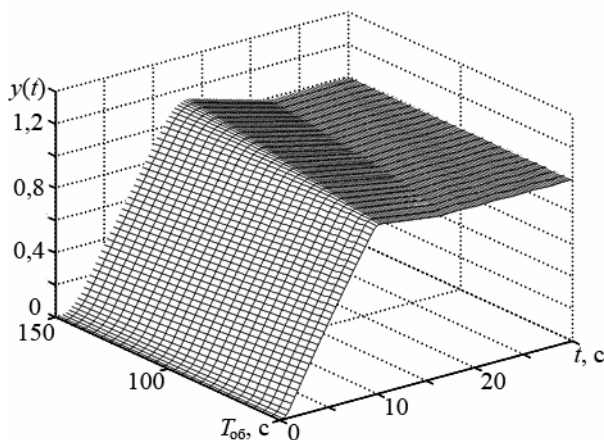


Рис. 5.21. Динамика изменения переходного процесса при различных значениях постоянной времени объекта управления

### 5.7. ИНТЕГРАЛЬНЫЙ ДИСКРЕТНЫЙ РЕГУЛЯТОР С ПЕРЕСТРАИВАЕМОЙ СТРУКТУРОЙ

Управление некоторыми непрерывными технологическими процессами с применением интегральных дискретных регуляторов с постоянным циклом затруднительно, поскольку, как известно, в них периодически оценивается сигнал ошибки регулирования и на основании этой информации формируется регулирующее воздействие с целью компенсации возмущений [10]. Это воздействие формируется в промежутке между двумя импульсами без учета предыстории. Однако возмущения, возникающие в технологических процессах, носят зачастую вероятностный характер и могут быть заданы лишь статистически. Кроме того, на сигнал ошибки регулирования может быть наложена помеха, существенно искажающая информацию о процессе. В итоге это может привести к резкому ухудшению качества процесса регулирования, а в отдельных случаях и к потере устойчивости системы. Следовательно, в

этих случаях целесообразно использовать для управления информацию за более длительное время.

В синтезируемом регуляторе использован периодический режим работы с постоянным циклом, начинающимся после появления возмущения. В каждом цикле контур регулирования замыкается на некоторое достаточно короткое время. При формировании регулирующего воздействия, в отличие от обычного дискретного интегрального регулятора, учитывается предыстория процесса, т.е. его характер в то время, когда система разомкнута. В этот период происходит оценка изменений сигнала ошибки относительно нулевого уровня.

В момент времени, когда система разомкнута и сигнал ошибки пересечет нулевое значение хотя бы один раз, очередной управляющий импульс не поступит на вход интегрирующего звена, хотя в момент замыкания контура регулирования этот импульс не равен нулю. Если за время размыкания системы сигнал ошибки не пересечет нулевое значение, то регулятор работает как обычный дискретный интегральный регулятор.

Синтезируемый регулятор с перестраиваемой структурой (рис. 5.22) состоит из узла памяти, вырабатывающего вспомогательные координаты  $\eta$  и  $\vartheta$ , и логического устройства, которое в зависимости от сочетания знаков вспомогательных координат формирует логический закон управления [16]:

$$\begin{aligned} \varepsilon^* &= \varepsilon \cdot U, \\ U &= F \& H, \end{aligned}$$

где  $H$  – значение импульса окончания счета длительностью  $t = T_{\text{зам}}$ ;

$$F = (M_{\eta} \bar{M}_{\vartheta} \vee \bar{M}_{\eta} M_{\vartheta}) \& R.$$

Здесь  $R$  – величина, характеризующая окончание цикла

$$M_{\eta} = (\eta \vee M_{\eta}) \& R; M_{\vartheta} = (\vartheta \vee M_{\vartheta}) \& R,$$

где  $\eta = \text{sgn}(\text{sgn}(E - \varepsilon) + 1)$ ;  $\bar{\vartheta} = \text{sgn}(\text{sgn}(\varepsilon + E) + 1)$ .

При наличии возмущения, когда величина сигнала ошибки превышает пороговое значение  $E$ , одна из вспомогательных координат  $\eta$  или  $\vartheta$  узла памяти воздействует на логическое устройство, которое по истечении некоторого наперед заданного времени  $T_{\text{ц}}$  замыкает канал

регулирования на время  $t = T_{зам}$ . Канал регулирования оказывается замкнутым, и сигнал ошибки  $\varepsilon(t)$  воздействует на интегрирующее звено, компенсируя возмущающее воздействие.

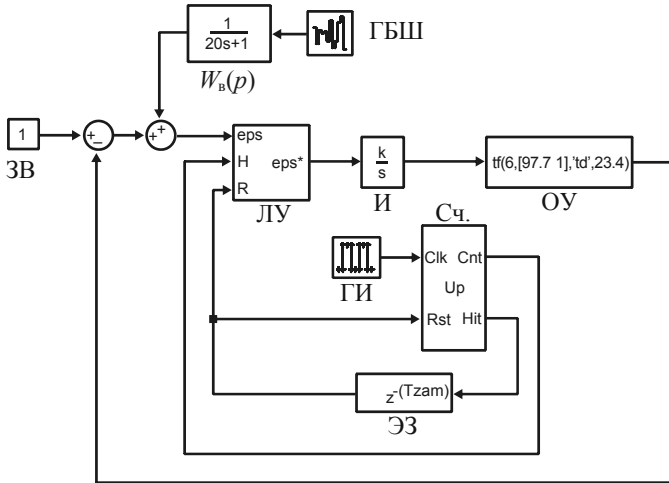


Рис. 5.22. Структурная схема системы с дискретным интегральным регулятором с переменной структурой: ОУ – объект управления; И – интегратор; ЛУ – логическое устройство; ГИ – генератор импульсов; Сч. – счетчик; ЭЗ – элемент задержки; ЗВ – задающее воздействие;  $W_b(p)$  – передаточная функция по каналу возмущения; ГБШ – генератор белого шума

Если за промежуток времени  $t = T_{ц}$  сигнал ошибки  $\varepsilon(t)$  изменит знак, то появление второй вспомогательной координаты приведет к тому, что в момент окончания цикла канал регулирования по-прежнему будет разомкнут и сигнал ошибки не поступит на интегрирующее звено. Иными словами, это приведет к сбросу счетчика времени цикла. Циклы будут повторяться до тех пор, пока величина сигнала ошибки не станет меньше порогового значения  $E$ .

На рис. 5.23 приведен переходный процесс рассматриваемой системы  $y(t)$  по каналу задающего воздействия при действии на ошибку регулирования не поддающихся измерению случайных возмущающих воздействий.

Применение рассматриваемого регулятора целесообразно, когда на сигнал ошибки налагается не поддающаяся измерению помеха, существенно искажающая информацию о процессе. Динамика изменения характеристик системы автоматического регулирования с интегральным дискретным регулятором с перестраиваемой структурой при действии возмущений приведена на рис. 5.24, а количественные показатели качества переходного процесса даны в табл. 5.9.

Т а б л и ц а 5.9

Оценки показателей качества переходного процесса в АСР с интегральным дискретным регулятором с перестраиваемой структурой

Показатели качества	Постоянная времени объекта $T$ , с		
	50	97,7	150
$\psi$	1	1	1
$A_{\max}$	0,02	0,04	0,13
$\sigma$ , %	2	4	13
$t_p$ , с	352	402	703

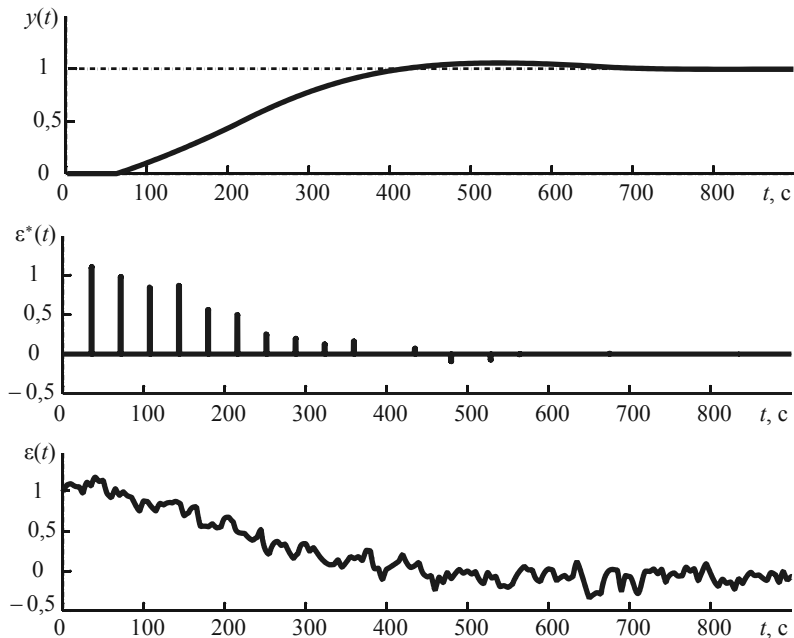


Рис. 5.23. Переходный процесс по каналу задающего воздействия

Таким образом, применение описанного регулятора снижает затраты на регулирование, поскольку регулирующее воздействие на объект поступает только в фиксированные промежутки времени, равные длительности импульсов, разрешающих регулирование, причем формирование закона регулирования происходит с учетом протекания переходного процесса в интервале между двумя импульсами. Исследования показали целесообразность применения разработанного регулятора при воздействии на сигнал ошибки помех, существенно искажающих информацию о технологическом процессе.

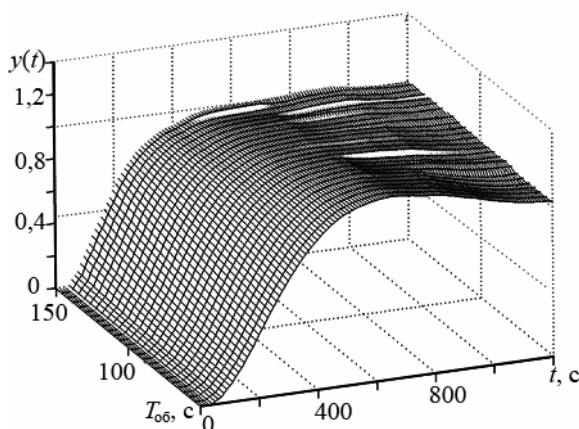


Рис. 5.24. Динамика изменения переходного процесса в системе автоматического регулирования при действии параметрических и высокочастотных координатных возмущений

## 5.8. РЕГУЛЯТОР КАЧЕСТВА ПЕРЕХОДНОГО ПРОЦЕССА

В системах с зависимостью динамики свойств объекта регулирования от внутренних координат технологического процесса могут существенно ухудшиться показатели качества переходного процесса. Это объясняется тем, что в системах автоматического регулирования с фиксированными настройками для подобных технологических процессов



качество переходного процесса изменяется в зависимости от возмущения и технологических режимов [10]. При некоторых технологических режимах изменение качества переходных процессов может существенно ухудшить показатели процесса.

Системы автоматического управления, которые в процессе своей работы изменяли бы слабо (в допустимых пределах) переходные процессы при изменении в широких пределах характеристик объекта, наиболее приемлемы для рассмотренных выше технологических процессов.

Требуемое качество переходного процесса характеризуется величиной затухания процесса регулирования. Оценка величины колебательности переходного процесса в регуляторе проводится путем сравнения сигнала ошибки за определенный фиксированный интервал времени (цикл исследования  $T_{ц}$ ) с двумя последовательно включаемыми, противоположными по знаку, заданными пороговыми значениями  $E$ . Цикл исследования в рассматриваемом регуляторе начинается после превышения сигналом ошибки первого порогового значения.

Предлагаемый регулятор качества переходного процесса (см. рис. 5.25) работает следующим образом. Сигнал ошибки регулирования  $\varepsilon$  поступает на вход МЛМ, на основе которых построено логическое устройство и блок памяти, в задачи которого входит генерация внутренних переменных  $\eta$  и  $\vartheta$ , а также запоминание их значений ( $M_{\eta}$ ,  $M_{\vartheta}$ ) на время цикла исследования. Этот блок настроен так, что при пересечении сигналом ошибки первого (положительного) порогового значения возводится (становится равной 1) переменная  $\eta$ , а при пересечении второго (отрицательного) – переменная  $\vartheta$ .

Цикл работы регулятора начинается после того, как переменная  $\eta$  станет равной единице. Сигнал поступает на вход логического устройства, которое подает команду для отсчета времени исследования. В течение этого времени определяется факт пересечения или непересечения сигналом ошибки второго порогового значения. Логическое устройство на весь период цикла исследования формирует выходной сигнал, равный логической единице, а по окончании цикла переходит в состояние

логического нуля. Сразу же после окончания исследования начинается цикл управления.

Характер управления определяет также логическое устройство. На его входы поступают переменные  $M_\eta$  и  $M_\vartheta$ . В зависимости от их сочетания на выходе формируются значения 1, 0, либо  $-1$ .

Передаточная функция замкнутой системы автоматического регулирования по каналу задающего воздействия выглядит следующим образом:

$$W_{з.с}(p) = \frac{W_{об}(p)W_n(p, k)}{W_{об}(p)W_n(p, k) + 1}. \quad (5.18)$$

Здесь  $W_n(p, k) = k/p$  – передаточная функция интегрирующего звена в канале регулирования;  $k = k_0 + \varphi$  – коэффициент передачи;  $k_0$  – параметр начальной настройки;  $\varphi$  – решения дифференциального уравнения вида

$$T \cdot \varphi' = v, \quad (5.19)$$

где  $T$  – постоянная времени интегрирования;  $v$  – выходная величина логического устройства, принимающая к концу цикла исследования значение из множества  $\{1, 0, -1\}$ :

$$v = A + B \cdot (-1); \quad (5.20)$$

$$A = (M_\eta \vee M_\vartheta) \& \overline{(M_\eta \& M_\vartheta)}; B = M_\eta \& M_\vartheta; \quad (5.21)$$

$$M_\eta^{t+1} = [(\eta \vee M_\eta) \& R]^t; M_\vartheta^{t+1} = [(\vartheta \vee M_\vartheta) \& R]^t, \quad (5.22)$$

где  $\eta = \text{sgn}(\text{sgn}(E - \varepsilon) + 1)$ ;  $\bar{\vartheta} = \text{sgn}(\text{sgn}(E + \varepsilon) + 1)$ ;  $R$  – величина, характеризующая окончание цикла.

При разрешении управления сигнал с логического устройства поступает на интегрирующее звено. В зависимости от того, что пришло на его вход  $-1$ ,  $-1$  или  $0$ , он увеличивает, уменьшает либо не изменяет коэффициент передачи интегрирующего звена в контуре регулирования системы.

По истечении времени, равного времени цикла исследования и времени замыкания, для изменения коэффициента передачи интегрирующего звена в контуре регулирования величины  $M_\eta$  и  $M_\vartheta$  обнуляются ( $R = 0$ ) и цикл повторяется заново.

Реализация описываемой системы автоматического регулирования, разработанной на базе МЛМ  $L$ - и  $V$ -структур, представлена на рис. 5.25.

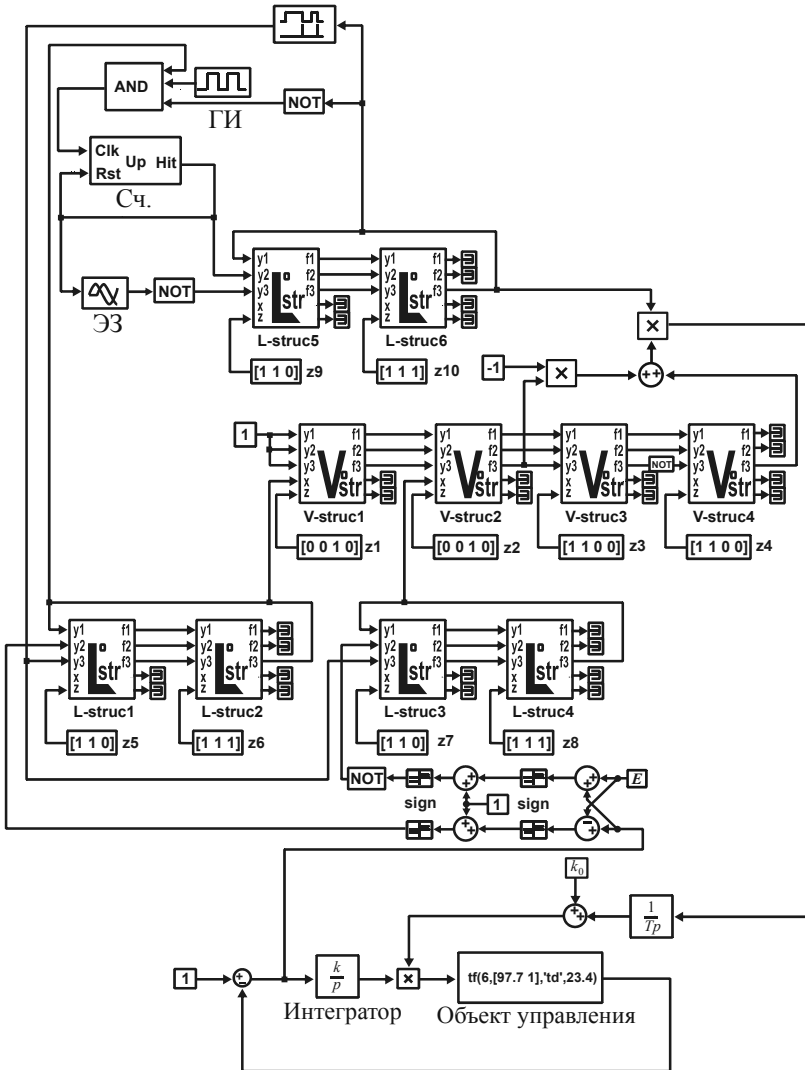


Рис. 5.25. Реализация в среде MatLab системы автоматического регулирования с регулятором качества переходного процесса

На рис. 5.26 приведена динамика процессов, протекающих в рассмотренной системе автоматического регулирования качества переходного процесса, при подаче на вход системы задающего воздействия  $s(t)$ :

$$s(t) = \begin{cases} 1 & \text{при } \{0 \leq t < 3500, 5600 \leq t < 7500, 9500 \leq t < 11700, t \geq 14100\}, \\ 0 & \text{при } \{t < 0\}, \\ 1,5 & \text{при } \{3500 \leq t < 5600, 7500 \leq t < 9500, 11700 \leq t < 14100\}. \end{cases}$$

В процессе функционирования системы коэффициент передачи интегрирующего звена в контуре управления подстраивается под технологический объект регулирования исходя из заданных критериев качества. После процесса подстройки, при постоянных параметрах объекта регулирования, изменение коэффициента передачи  $k$  сопровождается пульсациями, связанными с протеканием периодических процессов исследования в регуляторе.

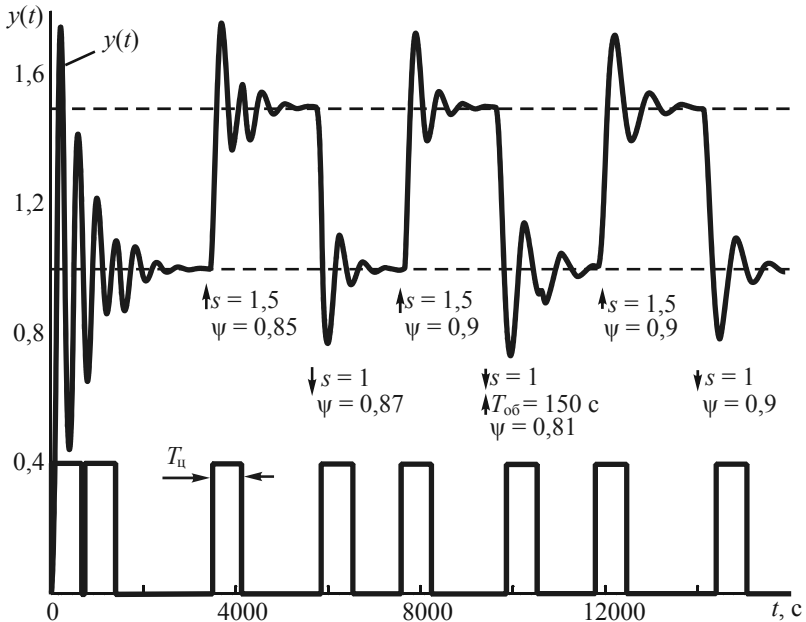


Рис. 5.26. Кривая разгона замкнутой системы при подаче задающего воздействия  $s(t)$

Система автоматического регулирования качества переходного процесса показала хорошие результаты в процессе исследования, которые выявили, что он обладает способностью поддерживать на заданном уровне степень затухания и максимальную динамическую ошибку переходного процесса (рис. 5.27, табл. 5.10).

Т а б л и ц а 5.10

Оценка показателей качества переходного процесса в АСР  
с регулятором качества при  $\psi = 0,9$

Показатели качества	Постоянная времени объекта $T$ , с		
	50	97,7	150
$\psi$	0,9	0,9	0,9
$A_{\max}$	0,314	0,31	0,313
$\sigma$ , %	31,4	31	31,3
$t_{ps}$ , с	560	770	1056

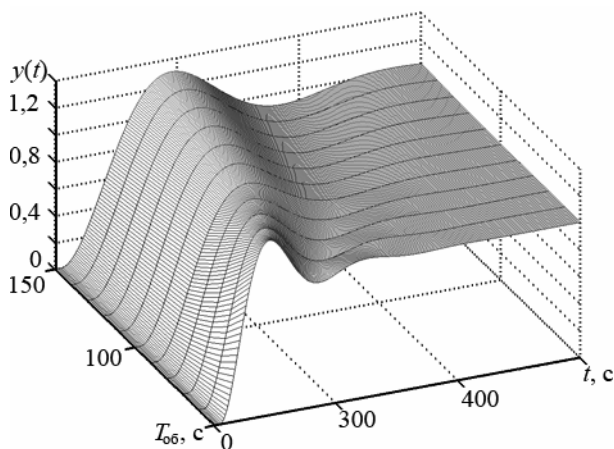


Рис. 5.27. Динамика изменения переходного процесса в АСР с регулятором качества при заданном  $\psi = 0,9$

Таким образом, разработанный регулятор позволяет поддерживать показатели качества переходного процесса (степень затухания, перерегулирование) системы автоматического регулирования на заданном уровне, а также удерживать ошибку регулирования в заранее заданном диапазоне. Регулятор может применяться совместно с различными технологическими объектами, динамические характеристики которых ме-

няются как в узких, так и в широких пределах, в частности для регулирования давления газозводшной смеси в магистральном трубопроводе.

## 5.9. ВЫВОДЫ

Основной результат исследований, изложенных в данной главе, заключается в решении проблемы построения систем автоматического управления с перестраиваемой структурой на основе МЛМ. Устройства управления, базирующиеся на принципах перестраиваемых структур, реализуемых с помощью МЛМ, обладают следующими особенностями:

1. Системы автоматического регулирования, построенные на основе перестраиваемых структур, наделяют всю систему новыми качествами, такими как быстродействие; понижение порядка уравнения движения для всех траекторий, кроме асимптот; нечувствительность к вариациям параметров объекта и действию внешних сил.

2. Системы с перестраиваемой структурой, построенные на основе МЛМ, получают выигрыш по времени в процессе перестройки своей внутренней структуры; в простоте реализации на однотипных элементах; в построении адаптивных систем путем добавления контура адаптации, реализуемого достаточно просто благодаря специфике СПС и МЛМ.

3. Реальный скользящий режим сопровождается высокочастотными колебаниями в окрестности прямой скольжения  $s$ , что негативно сказывается на работе механических и электромеханических приводов, которые часто используются на практике в качестве силовых установок. Это обстоятельство делает ограниченным практическое использование классических СПС.

4. Применение модифицированных алгоритмов управления с перестраиваемой структурой позволяет использовать СПС совместно с механическими и электромеханическими приводами, при этом решается задача высококачественного управления.

5. Управляющее устройство, базирующееся на принципах перестраиваемости своей внутренней структуры, придает всей системе автоматического регулирования способность поддерживать на заданном уровне не только технологический параметр, но и динамику изменения его во времени, т.е. регулировать качество технологического процесса.

## ЛИТЕРАТУРА

1. *Автоматизация* настройки систем управления / В.Я. Ротач, В.Ф. Кузищин, А.С. Ключев и др. – М.: Энергоатомиздат, 1984. – 272 с.
2. *Андык В.С.* Теория автоматического управления: Учебное пособие к практическим занятиям. – Томск: Изд-во ТПУ, 2004. – 108 с.
3. *Бесекерский В.А., Попов Е.П.* Теория систем автоматического регулирования. – М.: Наука, 1972. – 768 с.
4. *Дорф Р., Бишоп Р.* Современные системы управления / Пер. с англ. Б.И. Копылова. – М.: Лаборатория базовых знаний, 2002. – 832 с.
5. *Емельянов С.В.* Системы автоматического управления с переменной структурой. – М.: Наука, 1967. – 336 с.
6. *Емельянов С.В., Коровин С.К.* Новые типы обратной связи. – М.: Наука, 1997. – 352 с.
7. *Каляев А.В.* Многопроцессорные системы с программируемой архитектурой. – М.: Радио и связь, 1984. – 240 с.
8. *Козлов Ю.М., Юсупов Р.М.* Беспорисковые самонастраивающиеся системы. – М.: Наука, 1969. – 456 с.
9. *Кориков А.М.* Основы теории управления: Учебное пособие. – 2-е изд. – Томск: Изд-во НТЛ, 2002. – 392 с.
10. *Математические модели технологических процессов и разработка систем автоматического регулирования с переменной структурой: Сборник трудов Гинцветмета № 21 / Под общ. ред. Б.Н. Петрова.* – М.: Металлургия, 1964. – 468 с.
11. *Прангшивили И.В., Абрамова Н.А., Бабичева Е.В., Игнатуценко В.В.* Микроэлектроника и однородные структуры для построения логических и вычислительных устройств. – М.: Наука, 1967. – 228 с.
12. *Спицын А.В., Мазуров В.М.* Высококачественная адаптивная система управления с ПИ-регулятором // Изв. ТулГУ. – 1997. – Т. 1, вып. 2. – С. 11–17.
13. *Стефани Е.П.* Основы расчета настройки регуляторов теплоэнергетических процессов. – М.: Энергия, 1972. – 376 с.
14. *Уткин В.И.* Скользящие режимы в задачах оптимизации и управления. – М.: Наука, 1981. – 368 с.
15. *Филипс Ч., Харбор Р.* Системы управления с обратной связью. – М.: Лаборатория базовых знаний, 2001. – 616 с.
16. *Шидловский С.В.* Автоматизация технологических процессов и производств: Учебное пособие. – Томск: Изд-во НТЛ, 2005. – 100 с.
17. *Шидловский С.В.* Логическая система с перестраиваемой структурой в задачах управления технологическими процессами // Автотметрия. – 2005. – № 4. – С. 104–113.
18. *Шидловский С.В.* Теория автоматического управления: Учебное пособие. – Томск: Изд-во НТЛ, 2005. – 40 с.

19. Шидловский С.В. Ячейка однородной среды. Пат. РФ на изобретение № 2251140 // Б.И. – 2005. – № 12. – 9 с.

20. Шидловский С.В. Ячейка однородной среды. Пат. РФ на изобретение № 2251141 // Б.И. – 2005. – № 12. – 13 с.

21. Шульце К.П., Реберг К.Ю. Инженерный анализ адаптивных систем / Пер. с нем. – М.: Мир, 1992. – 280 с.



## АСПЕКТЫ ТЕОРИИ НЕЧЕТКИХ МНОЖЕСТВ В ЗАДАЧАХ КАЧЕСТВЕННОГО АНАЛИЗА ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

---

---

### 6.1. ВВОДНЫЕ ПОНЯТИЯ

Содержательная и ценностная стороны объекта могут быть воспроизведены в его модели введением понятия пространства качественных различий  $(X, \tau, \varepsilon(\cdot))$  как *некоторая тройка*, где  $X$  – множество,  $\tau$  – топология и  $\varepsilon(\cdot)$  – *семейство классов качественных различий* (класс определяется из условий методов исследования системы).

Существует три типа подходов к выделению качественных различий в исследуемой системе [6].

1. Физическая неразличимость элементов класса, когда все элементы  $y \in E(\cdot)$ ,  $E(\cdot)$  – *класс качественных различий* в пространстве наблюдаемых переменных или модельном пространстве  $X$ , имеет одно и то же качественное содержание.

2. Сенсорная неразличимость, когда все элементы  $y \in E(\cdot)$  в пространстве наблюдаемых переменных или модельном пространстве *не различимы* по проявлению качества  $q$  чувствительными элементами (рецепторами, датчиками наблюдений и т.п.).

3. Прагматическая неразличимость, когда все элементы  $y \in E(\cdot)$  определенного на пространстве наблюдаемых переменных или модельном пространстве имеют *несущественное* различие по проявлению качества  $q$  для функционирования конкретного наблюдателя, вследствие чего все  $y \in E(\cdot)$  в этом смысле неразличимы.

Все вышеперечисленные *подходы* объединяет выделение определенным способом *максимального множества-класса*  $E(\cdot)$ , элементы которого качественно или содержательно равнозначно интерпретируются. Как раз интерпретация элементов множества с качественной и содержательной стороны закладывается в основу методов анализа раз-

личий состояния системы [8]. Качественные свойства всех элементов, входящих в выделенный класс  $E(\cdot)$ , оцениваются характеристической функцией  $\mu_q(y)$ , определенной на  $X$ . Выбор значений подобной функции может проводиться по принципам накопления качественных различий. При пороговом принципе  $\mu_q \in \{0,1\}$  ( $\mu_q(y) = 1$ , если  $y \in E(q)$ , и  $\mu_q(y) = 0$ , если  $y \notin E(q)$ ). Если используется принцип относительной численной оценки накопления качественных различий, функция  $\mu$  может принимать весь ряд значений из замкнутого интервала  $[0, 1]$ .

## 6.2. НЕКОТОРЫЕ ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ ТЕОРИИ НЕЧЕТКИХ МНОЖЕСТВ

Произвольное подмножество  $A$  универсального множества  $X$  однозначно определяется

$$I_A(x) = \begin{cases} 1 & \text{для } x \in A, \\ 0 & \text{для } x \notin A. \end{cases}$$

Л. Заде ввел новое понятие **нечеткого множества**, тем самым расширил класс подмножеств  $X$  [5].

Нечеткое множество может представляться с помощью числовой функции со значениями из всего отрезка  $[0, 1]$  – обобщенного индикатора. Эти обобщенные индикаторы в теории нечетких множеств названы **функциями принадлежности**. Для некоторого нечеткого множества  $A$  значение функции принадлежности  $\mu_A(x)$  трактуется как *степень принадлежности точки  $x$  определенному множеству  $A$* .

Для нечетких множеств можно определить следующие стандартные теоретико-множественные операции и отношения:

$$\begin{aligned} A = B &\Leftrightarrow \mu_A(x) = \mu_B(x), \quad \forall x \in X, \\ A \subset B &\Leftrightarrow \mu_A(x) \leq \mu_B(x), \quad \forall x \in X, \\ C = A \cup B &\Leftrightarrow \mu_C(x) = \max[\mu_A(x), \mu_B(x)], \quad \forall x \in X, \\ C = A \cap B &\Leftrightarrow \mu_C(x) = \min[\mu_A(x), \mu_B(x)], \quad \forall x \in X, \\ E = A' &\Leftrightarrow \mu_E(x) = 1 - \mu_A(x), \quad \forall x \in X. \end{aligned} \tag{6.1}$$

Всякое нечеткое множество  $A$  может быть представлено как объединение нечетких подмножеств  $A$  с функцией принадлежности

$$\mu_A(x)/x = \begin{cases} \mu_A(x^*) & \text{для } x = x^* \in X, \\ 0 & \text{для } x \neq x^* \in X. \end{cases}$$

Это выражение можно записать для  $X$ -мощности континуума в виде

$$A = \int_X \mu_A(x) / x,$$

и для счетных  $X$ :

$$A = \sum_{x \in X} \mu_A(x) / x.$$

**Пример.** Пусть  $H = \{\langle a \rangle, \langle b \rangle, \langle c \rangle, \langle d \rangle\}$ , тогда множество  $A = \{\langle a \rangle, \langle b \rangle, \langle c \rangle\}$  является обычным (не нечетким) подмножеством  $H$ , а множество  $G = \{\langle 0, 4a \rangle, \langle b \rangle, \langle 0, 7d \rangle\}$  – нечетким.

Если, например,  $H = \{\langle 0 \rangle, \langle 0, 1 \rangle, \langle 0, 2 \rangle, \dots, \langle 1 \rangle\}$ , то нечеткое подмножество  $H$  может быть выражено в виде

$$G = \{\langle 0, 4/0, 3 \rangle, \langle 0, 5/0, 6 \rangle, \langle 0, 1/0, 8 \rangle, \langle 1/1 \rangle\}$$

или

$$J = \{\langle 0, 6/0, 3 \rangle, \langle 0, 5/0, 6 \rangle, \langle 0, 9/0, 8 \rangle, \langle 0/1 \rangle\}.$$

Причем в данном случае

$$G' = J \text{ и } \mu_G(h) = 1 - \mu_J(h).$$

Некоторые важные результаты теории нечетких множеств основаны на так называемом принципе расширения. Он заключается в следующем. Пусть  $f$  – отображение  $X \rightarrow Y$ , т.е.  $y = f(x)$ . Если  $A$  – нечеткое подмножество  $X$ :

$$A = \int_X \mu_A(x) / x,$$

то образ  $A$  при  $f$  будет

$$f(A) = \int_A \mu_A(x) / f(x).$$

Если  $f$  – бинарное отображение  $X \times Y \rightarrow Z$  и  $A, B$  – нечеткие подмножества  $X, Y$ , то

$$f(A \times B) = \int_{X \times Y} (\mu_A(x) \wedge \mu_B(y)) / f(x, y), \quad (6.2)$$

где  $\wedge$  обозначает операцию  $\min$ .

**Нечеткий автомат** представляет собой развитие понятия стохастического автомата переменной структуры и приспособлен к работе с входами «нечеткого» характера, по Заде.

Нечеткая величина принадлежит к категории величин, описывающих объекты с неопределенными границами, характеризуемыми функцией принадлежности, заданной на некотором множестве.

Таким образом, нечеткий автомат определяется шестеркой величин  $[Z, Q, U, F, H, \mathfrak{Z}]$ , в которой, кроме конечного множества нечетких входов  $Z = \{z\}$ , конечного множества состояний  $Q = \{q\}$ , конечного множества выходов  $U = \{u\}$ , функции переходов между состояниями  $F$  и выходной функции  $H$ , фигурирует еще вектор принадлежности  $\mathfrak{Z}$ , определяющий нечеткое состояние системы. Для изменения значений функций принадлежности  $\mathfrak{Z}$  каждому нечеткому входу  $z^k$  можно поставить в соответствие матрицу пересчета векторов принадлежности  $\Xi^k(n)$ . Такой нечеткий автомат может скоординировать  $p$  элементарных движений, соответствующих отдельным подпроцессам так, чтобы они составили требуемое сложное движение, переводящее систему из некоторого начального состояния в требуемое.

### 6.3. НЕЧЕТКАЯ ЛОГИКА И ПРИБЛИЖЕННЫЕ ВЫВОДЫ

Если исходные данные, цели и ограничения плохо поддаются формальному описанию в силу своей природы, то исследование таких систем с применением строгих математических методов вызывает значительные трудности.

В отличие от предыдущих логических систем, системы, базирующиеся на теории нечеткой логики, являются моделью приближенных рассуждений и выводов. Очень важное место в проблемах принятия решений занимает анализ ситуаций, в которых определяющими являются не количественные, а *качественные* характеристики. Основным

назначением и преимуществом нечеткой логики как раз и является построение формализаций качественных методов исследования.

Нечеткую логику можно рассматривать как расширение логики Лукасевича [6], которая образует для нечеткой логики базовую логику. Значениями истинности в логике Лукасевича являются действительные числа на отрезке  $[0, 1]$  такие, что

$$u(z') = 1 - u(z), \quad u(z \vee m) \triangleq \max(u(z), u(m)),$$

$$u(z \wedge m) \triangleq \min(u(z), u(m)), \quad u(z \Rightarrow m) \triangleq \min(1, 1 - u(z) + u(m)),$$

где  $u(z)$  – значение истинности предложения  $z$ ;  $'$ ,  $\wedge$ ,  $\vee$ ,  $\Rightarrow$  – операции отрицания, дизъюнкции, конъюнкции и импликации.

Изложенные выше значения истинности в нечеткой логике представляют нечеткие подмножества  $[0, 1]$ , которым обычно приписываются лингвистические метки:

$$\begin{aligned} \aleph = & (\text{истинно, ложно, не истинно, очень истинно,} \\ & \text{не очень истинно, более или менее истинно,} \\ & \text{не очень истинно и не очень ложно, ...). \end{aligned} \quad (6.3)$$

Всякой метке  $x \in \aleph$  соответствует функция принадлежности  $\mu_x(u)$  такая, что

$$x = \int_0^1 \mu_x(u) / u. \quad (6.4)$$

Пусть, например, значение *истинно* определено как

$$\mu_{\text{ист}}(u) = \begin{cases} 0 & \text{для } 0 \leq u \leq \zeta, \\ 2 \left( \frac{u - \zeta}{1 - \zeta} \right)^2 & \text{для } \zeta \leq u \leq (\zeta + 1)/2, \\ 1 - 2 \left( \frac{u - \zeta}{1 - \zeta} \right)^2 & \text{для } (\zeta + 1)/2 \leq u \leq 1, \end{cases}$$

где  $\zeta \in [0, 1]$ .

Согласно (6.4) получим

$$\text{истинно} \triangleq \int_{\zeta}^{(\zeta+1)/2} 2 \left( \frac{u-\zeta}{1-\zeta} \right)^2 / u + \int_{(\zeta+1)/2}^1 \left( 1 - 2 \left( \frac{u-\zeta}{1-\zeta} \right)^2 \right) / u$$

и, следовательно,

$$\text{не истинно} = \int_0^1 (1 - \mu_{\text{ист}}(u)) / u .$$

Значение *ложно*, выраженное через значение *истинно*, может выглядеть следующим образом:

$$\text{ложно} \triangleq \int_0^1 \mu_{\text{ист}}(1-u) / u .$$

В целом подразумевается, что множество  $\mathfrak{S}$  значений истинности порождается контекстно-свободной грамматикой  $G$ , состав которой, кроме первичных терминов *истинно*, *ложно* и операций *и*, *или*, *не*, расширяется модификаторами *очень*, *более или менее*, *почти*, *около* и т.д.

Значения этих модификаторов являются отображением в себя алгебры нечетких подмножеств  $[0, 1]$ . Например, аппроксимацией для *очень* и *более или менее* служат операции концентрирования и растяжения:

$$\text{con}(A) = \int_U \mu_A^2(u) / u, \quad \text{dil}(A) = \int_U \mu_A^{1/2}(u) / u,$$

где  $A = \int_U \mu_A(u) / u$ .

Применение операции концентрирования к нечеткому множеству означает уменьшение нечеткости или неопределенности в задании этого множества. Часто это может быть связано с поступлением дополнительной полезной информации.

Применение же операции растяжения, наоборот, означает усиление неопределенности в задании нечеткого множества. Это может быть следствием либо потери части информации, либо неучета в исходной нечеткой модели информации, поступающей о дополнительных факторах [2].

**Пример.** Пусть  $\mathfrak{N}$  имеет вид (6.3). Нетрудно проверить, что  $\mathfrak{N}$  порождается контекстно-свободной грамматикой  $G$ , система продукций которой определяется следующим образом:

$$\begin{array}{ll}
 T \rightarrow A; & C \rightarrow D; \\
 T \rightarrow T \text{ или } A; & C \rightarrow E; \\
 A \rightarrow B; & D \rightarrow \text{очень } D; \\
 A \rightarrow A \text{ и } B; & E \rightarrow \text{не очень } E; \\
 B \rightarrow C; & D \rightarrow \text{истинно}; \\
 B \rightarrow \text{не } C; & E \rightarrow \text{ложно}. \\
 C \rightarrow (T); &
 \end{array} \quad (6.5)$$

Здесь каждая продукция индуцирует соответствующие отношения в  $H$   $[0, 1]$ , например, продукция  $A \rightarrow A \text{ и } B$  соответствует  $A_{\text{л}} = A_{\text{п}} \cap B_{\text{п}}$  (индексы «л» и «п» служат для различения левой и правой частей соответственно).

Таким образом, таблица отношений, двойственная к (6.5), будет

$$\begin{array}{ll}
 T \rightarrow A, & T_{\text{л}} = A_{\text{п}}; \\
 T \rightarrow T \text{ или } A, & T_{\text{л}} = T_{\text{п}} \cup A_{\text{п}}; \\
 A \rightarrow B, & A_{\text{л}} = B_{\text{п}}; \\
 A \rightarrow A \text{ и } B, & A_{\text{л}} = A_{\text{п}} \cap B_{\text{п}}; \\
 B \rightarrow C, & B_{\text{л}} = C_{\text{п}}; \\
 B \rightarrow \text{не } C, & B_{\text{л}} = (C_{\text{п}})'; \\
 C \rightarrow (T), & C_{\text{л}} = T_{\text{п}}; \\
 C \rightarrow D, & C_{\text{л}} = D_{\text{п}}; \\
 C \rightarrow E, & C_{\text{л}} = E_{\text{п}}; \\
 D \rightarrow \text{очень } D, & D_{\text{л}} = D_{\text{п}}^2; \\
 E \rightarrow \text{не очень } E, & E_{\text{л}} = (E_{\text{п}}^2)'; \\
 D \rightarrow \text{истинно}, & D_{\text{л}} = \text{истинно}; \\
 E \rightarrow \text{ложно}, & E_{\text{л}} = \text{ложно}.
 \end{array} \quad (6.6)$$

Учитывая (6.5) и (6.6), можно получить, что значение  $z = \text{не очень истинно}$  и  $\text{не очень ложно}$  является

$$\left[ \underbrace{\left( \underbrace{\text{истинно}}_{\text{очень}} \right)^2}_{\text{не}} \right] \underset{\text{и}}{\cap} \left[ \underbrace{\left( \underbrace{\text{ложно}}_{\text{очень}} \right)^2}_{\text{не}} \right].$$

Покажем теперь, как производятся операции над лингвистическими значениями истинности.

Пусть  $z$  и  $m$  – нечеткие предложения и их значения истинности выглядят следующим образом:

$$u(z) = \int_U \mu_A(u) / u = A, \quad u(m) = \int_V \mu_B(v) / v = B. \quad (6.7)$$

Следуя принципу расширения (6.2) для произвольного отображения (\*)  $V \times U \Rightarrow W$ ,

$$A * B = \int_W (\mu_A(v) \wedge \mu_B(u)) / u * v.$$

Подставляя формулы (6.1) для операций  $u$ , или,  $\Rightarrow$  в (6.7), получим

$$u(z \text{ и } m) = \sum_{i,j} (\mu_i \wedge u_j) / v_i \wedge u_j,$$

$$u(z \text{ или } m) = \sum_{i,j} (\mu_i \vee u_j) / v_i \vee u_j,$$

$$u(z \Rightarrow m) = \sum_{i,j} (\mu_i \wedge u_j) / (1 \wedge (1 - (v_i - u_j))).$$

Причем результаты этих операций могут выходить за пределы  $\mathfrak{N}$ , поэтому для произвольного значения истинности  $\Omega$ , полученного из (6.3), требуется аппроксимация лингвистическими значениями истинности  $\Omega^*$  из  $x: \Omega^* = LA[\Omega]$ .

Операция  $LA$  в общем случае не единственная, но в большинстве приложений  $\Omega^*$  находят по критерию  $\min$  его лингвистической длины [6].

На практике особую ценность имеет используемое в нечеткой логике композиционно правило вывода.

Правило логического вывода состоит в следующем. Пусть булевой переменной  $s$  присвоено значение  $A$ , т.е. « $s$  есть  $A$ » ( $s \in A$ ), и зависи-



мость переменных  $s$ ,  $U$  определяется отношением « $U$  есть: если  $A$ , тогда  $B$ , иначе  $C$ », которое понимается как  $U = (A \cap B) \cup (\neg A \cap C)$ .

Тогда из этих соотношений по правилам булевой логики можно вывести

$$A \cup [(A \cap B) \cup (\neg A \cap C)] = B$$

или

$$A \cup (\text{если } A, \text{ тогда } B, \text{ иначе } C) = B. \quad (6.8)$$

Для нечетких множеств  $A$ ,  $B$  и  $C$  формула (6.8) заведомо неверна, так как интерпретация « $s$  есть  $A$ », как  $u \in A$ , становится в этом случае бессмысленной. Естественный выход из этого затруднения состоит в том, чтобы интерпретировать « $s$  есть  $A$ » как назначение лингвистического значения  $A$  тому свойству  $s$ , которое  $A$  определяет. Например, предложение «Андрей высокий» может быть описано как

$$\text{Рост (Андрей)} = \text{высокий}.$$

Здесь Рост – название переменной; Андрей – аргумент; высокий – присваиваемое логическое значение.

Подобно нечеткое предложение

$$p \stackrel{\Delta}{=} \text{Саша намного выше Тани}$$

следует интерпретировать как

$$(\text{Рост (Саша)}; \text{Рост (Таня)}) = \text{намного выше}.$$

В правой части стоит бинарное нечеткое отношение присваиваемой переменной, стоящей в левой части.

В общем случае пусть  $U_1, \dots, U_n$  – некоторые универсальные множества, а  $R = R(U_1, \dots, U_n)$  – нечеткое ограничение в  $U_1 \times U_2 \times \dots \times U_n$ .

Нечеткое предложение  $p \stackrel{\Delta}{=} (u_1, \dots, u_n)$  есть  $A$  может быть представлено как уравнение назначения ограничению  $R$  значения  $A$ :  $R(u_1, \dots, u_n) = A$ .

Предположим, например, что

$$P_1 \stackrel{\Delta}{=} u_1 \text{ есть маленький,}$$

$$P_2 \stackrel{\Delta}{=} u_1 \text{ и } u_2 \text{ есть приблизительно равны,}$$

тогда предложения переводятся как уравнения назначения

$$R(u_1) = \text{маленький} , \quad (6.9)$$

$$R(u_1, u_2) = \text{приблизительно равны} . \quad (6.10)$$

Решение системы (6.9) – (6.10) относительно  $R(u_2)$  задается как

$$R(u_2) = R(u_1) \circ R(u_1, u_2) , \quad (6.11)$$

где  $\circ$  – знак композиции унарного ограничения  $u_1$  и бинарного на  $(u_1, u_2)$ .

Таким образом,

$$R(u_2) = LA[\text{мал} \circ \text{приблизительно равны}] .$$

Правило вывода, базирующееся на решении отношения  $R(u_2)$  в (6.9) – (6.11), называется композиционным правилом.

В общем случае его можно представить в виде

$$\begin{array}{l} A_1 : u_1 \text{ и } u_2 \text{ есть } A \\ A_2 : u_2 \text{ и } u_3 \text{ есть } B \end{array} \Rightarrow A_3 : u_1 \text{ и } u_3 \text{ есть } LA[A \circ B] ,$$

где  $A$  и  $B$  – нечеткие отношения, выраженные в лингвистических терминах.

Одним из важных нечетких утверждений является предложение

$$\text{если } A, \text{ тогда } B, \text{ иначе } C , \quad (6.12)$$

соответствующее бинарному отношению

$$A \times B + (\neg A) \times C . \quad (6.13)$$

Применяя комбинационный вывод, можно показать, что если

$$R(u) = A , \quad R(u, v) = \text{если } A, \text{ тогда } B, \text{ иначе } C ,$$

то

$$R(v) = A \circ (A \times B + \neg A \times C) = B + A \circ (\neg A \times C) \neq B .$$

В частном случае обычных (не нечетких)  $A$ ,  $B$  и  $C$

$$A \circ (\text{если } A, \text{ тогда } B, \text{ иначе } C) = B .$$

Выражения (6.12) – (6.13) отражают наиболее простые отношения между нечеткими переменными. Для описания более сложных отноше-

ний используются так называемые нечеткие алгоритмы. **Нечеткий алгоритм** есть упорядоченное множество нечетких предложений:

1) назначающие инструкции:

$x = \text{малый}$  ;

$x = \text{не малый и не очень большой}$  ;

$x = \text{приблизительно } 5$  ;

2) условные нечеткие высказывания:

**если**  $X$  – *малый*, **тогда**  $y$  – *большой*, **иначе**  $y$  – *небольшой* ,

**если**  $X$  – *положителен*, **тогда** *увеличивать*  $y$  *незначительно* ;

3) безусловные инструкции:

«умножить  $X$  на  $X$ », «идти к 7» и т.д.

Выполнение этих предложений происходит в соответствии с композиционным правилом вывода. В предложениях нечеткие алгоритмы бывает удобно классифицировать на некоторые категории, например: алгоритмы управления, принятия решения, идентификации и т.д.

#### 6.4. РЕГУЛЯТОР КАЧЕСТВА

Рассмотрим замкнутую систему автоматического регулирования (рис. 6.1) с передаточной функцией вида

$$W_{з.с}(p) = \frac{W_{об}(p)W_{и}(p, k)}{W_{об}(p)W_{и}(p, k) + 1}.$$

Здесь  $W_{об}(p) = k_{об} \exp(-\tau p) / (T_{об}p + 1)$  – передаточная функция объекта управления;  $W_{и}(p, k) = k / p$  – передаточная функция интегрирующего звена в канале регулирования;  $k = k_0 + \varphi$  – коэффициент передачи;  $k_0$  – параметр начальной настройки;  $\varphi$  – решение дифференциального уравнения вида

$$T \cdot \varphi' + 1 = v_j,$$

где  $T$  – постоянная времени;  $v_k$  – выходная величина блока 3, принимающая к концу  $j$ -го цикла исследования значение

$$v_j = v_{j-1} + \Delta z.$$

Здесь  $v_{j-1}$  – значение выходной величины блока 3 на предыдущем цикле исследования;  $\Delta z$  – корректирующий сигнал с выхода нечеткого логического регулятора (НЛР).

В системе задаются максимальный верхний  $E_1$  и нижний  $E_2$  уровни отклонения ошибки регулирования, с помощью которых на каждом цикле исследования блоком 1 вычисляются величины [9]:

$$\sigma_1 = \max(\varepsilon - E_1), \quad \sigma_2 = \min(|E_2| - |\varepsilon|),$$

на основании которых НЛР оценивает, насколько необходимо изменить коэффициент передачи регулятора основного контура регулирования.

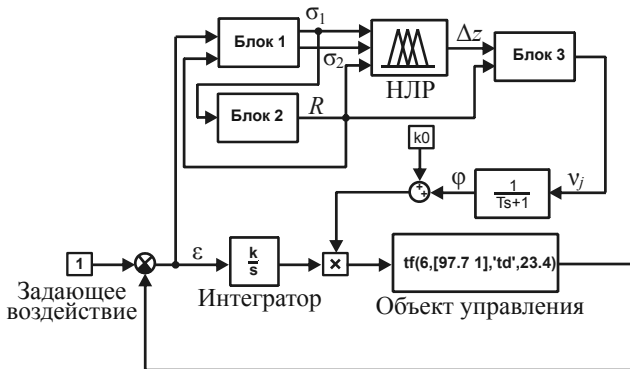


Рис. 6.1. Структурная схема системы автоматического регулирования

На НЛР возлагается задача выработки корректирующего воздействия в диапазоне изменения динамической ошибки регулирования относительно ее пороговых значений, поэтому при синтезе нечеткого регулятора используем алгоритм вывода по Мамдани [2] как наиболее простой и интуитивный. Несмотря на то что алгоритм Мамдани рекомендуют применять в других областях [10] (экспертные системы и системы принятия решений), он может эффективно работать и при разработке нечетких систем управления технологическим процессом.

Как показали исследования [7], при включении НЛР с алгоритмом Мамдани в анализируемую систему стабилизации целесообразно использовать его в качестве дополнительного канала управления, так как для работы НЛР с данным алгоритмом в прямом канале управления потребовалось бы ввести в схему дополнительные динамические звенья.

Таким образом, НЛР с алгоритмом Мамдани выступает в качестве корректирующего элемента к основному закону регулирования.

Нечеткий регулятор, включенный на параллельную коррекцию параметра И-регулятора, имеет три входа, один выход и вырабатывает сигнал коррекции  $\Delta z$  в зависимости от состояния входных сигналов  $\sigma_1$ ,  $\sigma_2$  и  $R$ . С помощью сигнала коррекции нечеткий регулятор изменяет коэффициент передачи основного регулятора.

Нечеткий регулятор состоит из трех основных блоков: блока фаззификации, блока нечеткого вывода и блока дефаззификации.

В первом блоке необходимо произвести фаззификацию входных сигналов, т.е. перевести четко определенные сигналы в нечеткую форму. Для этого сначала определимся с количеством нечетких высказываний для каждого входного сигнала, которыми будет оперировать база знаний НЛР, являющаяся содержательным компонентом блока нечеткого вывода. Для заполнения первоначально пустой базы знаний необходимо определить цели управления.

В данном случае целью управления является поддержание технологического параметра в заданном диапазоне отклонений от требуемого значения при возникновении контролируемых и неконтролируемых возмущений. При этом чем сильнее увеличивается отклонение технологического параметра от порогового значения, тем больше должно быть приращение, изменяющее коэффициент передачи регулятора.

Как уже упоминалось, НЛР имеет три входных сигнала. Для сигнала  $\sigma_1$  запишем следующие высказывания: «*сильно большой*» (СБ); «*большой*» (Б); «*средний*» (С); «*малый*» (М); «*сильно малый*» (СМ). Для сигнала  $\sigma_2$ : «*положительный большой*» (ПБ); «*положительный средний*» (ПС); «*положительный малый*» (ПМ); «*нулевой*» (Н); «*отрицательный малый*» (ОМ); «*отрицательный средний*» (ОС); «*отрицательный большой*» (ОБ). Наконец, для  $R$ : «*разрешено*» (Р).

Тогда для выходного сигнала  $\Delta z$  примем следующие выражения: «*положительный большой*» (ПБ); «*положительный малый*» (ПМ); «*нулевой*» (Н); «*отрицательный малый*» (ОМ); «*отрицательный большой*» (ОБ).

Опирируя полученными входными и выходными лингвистическими переменными и поставленной целью управления, составляем базу знаний НЛР (табл. 6.1).

Т а б л и ц а 6.1

База правил НЛР

$\sigma_1$	$\sigma_2$						
	ПБ	ПС	ПМ	Н	ОМ	ОС	ОБ
СБ	ОБ	ОБ	ОБ	Н	ПМ	ПБ	ПБ
Б	ОБ	ОБ	ОМ	Н	ПМ	ПМ	ПБ
С	ОМ	ОМ	ОМ	Н	Н	ПМ	ПБ
М	ОМ	ОМ	Н	Н	Н	ПМ	ПБ
СМ	Н	Н	Н	Н	Н	ПМ	ПБ

Выбрав лингвистические переменные и составив базу знаний НЛР, можно перейти к этапу фазификации, целью которого является установление соответствия между конкретными значениями отдельной входной переменной НЛР и значениями функций принадлежности соответствующего ей терма входной лингвистической переменной. По завершении этого этапа должны быть определены конкретные значения функций принадлежности по каждому из лингвистических термов, которые используются в табл. 6.1, для всех входных переменных.

Итак, будем использовать множества  $A_1 = \{\langle\text{СБ}\rangle, \langle\text{Б}\rangle, \langle\text{С}\rangle, \langle\text{М}\rangle, \langle\text{СМ}\rangle\}$ ,  $A_2 = \{\langle\text{ПБ}\rangle, \langle\text{ПС}\rangle, \langle\text{ПМ}\rangle, \langle\text{Н}\rangle, \langle\text{ОМ}\rangle, \langle\text{ОС}\rangle, \langle\text{ОБ}\rangle\}$ ,  $A_3 = \{\langle\text{Р}\rangle\}$  в качестве терм-множеств первой, второй и третьей входных лингвистических переменных соответственно, кусочно-линейные функции принадлежности которых изображены на рис. 6.2, а, б, в.

В качестве терм-множеств выходной лингвистической переменной будем использовать множество  $V = \{\langle\text{ПБ}\rangle, \langle\text{ПМ}\rangle, \langle\text{Н}\rangle, \langle\text{ОМ}\rangle, \langle\text{ОБ}\rangle\}$  с функциями принадлежности, изображенными на рис. 6.2, г.

Далее используется алгоритм вывода Мамдани, формально заключающийся в следующем [2]:

– агрегирование подусловий в нечетких правилах. Для нахождения степени истинности условий каждого из правил используются нечеткие логические операции. Те правила, степень истинности которых отлична от нуля, считаются активными и используются для дальнейших расчетов;

– активизация подзаключений в нечетких правилах. Осуществляется по формуле

$$\mu'(y) = \min\{c_i, \mu(y)\},$$

где  $\mu(y)$  – функция принадлежности терма, который является значением некоторой входной переменной;  $c_i$  – степень истинности подзаключений для каждого из правил. Для сокращения времени вывода учитываются только активные правила;

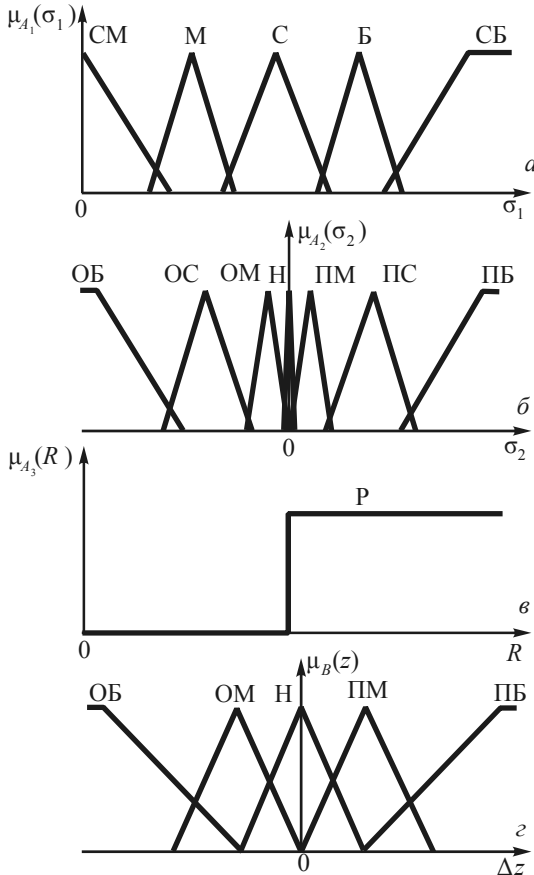


Рис. 6.2. Функции принадлежности для:  $a$  – первой,  $b$  – второй,  $v$  – третьей входной переменной;  $z$  – выходной переменной

– аккумуляция заключений нечетких правил продукции. Осуществляется объединение нечетких множеств, соответствующих термам под-

заклучений, принадлежащим к одним и тем же выходным лингвистическим переменным:

$$\mu_D(x) = \max \{ \mu_A(y), \mu_B(y) \},$$

где  $\mu_A(y)$ ,  $\mu_B(y)$  – функции принадлежности объединяемых нечетких множеств;

– дефазификация выходных переменных осуществляется по методу центра тяжести:

$$y = \left[ \int_{\min}^{\max} x \mu(x) dx \right] / \left[ \int_{\min}^{\max} \mu(x) dx \right],$$

где  $y$  – результат дефазификации;  $x$  – переменная, соответствующая входной лингвистической переменной;  $\mu(x)$  – функция принадлежности нечеткого множества;  $\min$ ,  $\max$  – левые и правые точки интервала носителя нечеткого множества рассматриваемой входной переменной.

Опишем работу регулятора (рис. 6.3). При превышении ошибки регулирования первого порогового значения  $E_1$  начинается отсчет времени блоком 2 и вычисление величин  $\sigma_1$  и  $\sigma_2$  блоком 1. По истечении времени, равного циклу исследования  $T_{\text{ц}}$ , блок 2 формирует импульс  $R$  длительностью  $T_{\text{зам}}$ , разрешающий изменение коэффициента передачи регулятора. При появлении импульса  $R$  НЛР на основе значений  $\sigma_1$  и  $\sigma_2$ , а также базы знаний формирует импульс длительностью  $T_{\text{зам}}$  и амплитудой  $\Delta z$ , на величину которой необходимо изменить коэффициент передачи регулятора. Сигнал с НЛР поступает на блок 3, который, в свою очередь, изменяет свой выходной сигнал  $v_j$  на величину  $\Delta z$  и фиксирует его до прихода следующего импульса  $R$ . При появлении координаты  $v_j$  коэффициент передачи регулятора  $k$  начинает изменяться на ее величину в течение некоторого времени  $T$  [3]. По истечении времени, равного  $T_{\text{ц}} + T_{\text{зам}}$ , величины  $\sigma_1$  и  $\sigma_2$  обнуляются и цикл повторяется заново.



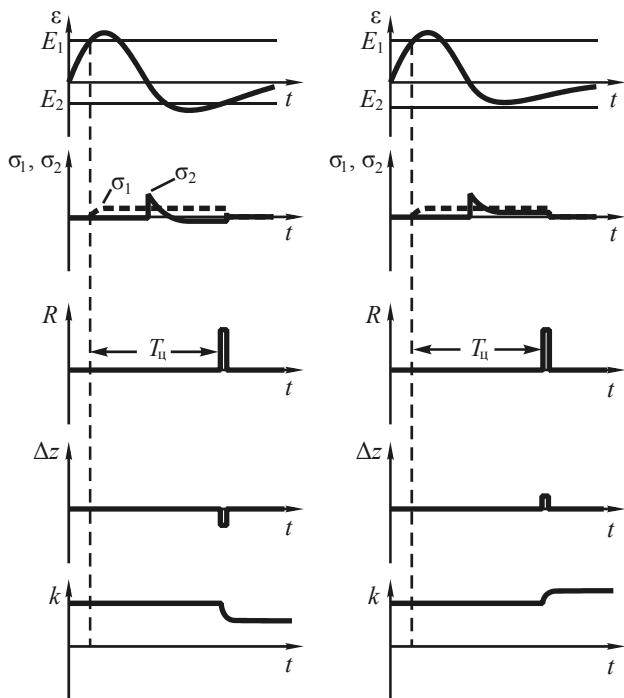


Рис. 6.3. Временная диаграмма работы регулятора

В табл. 6.2 приведены сравнительные оценки качества переходных процессов в рассматриваемой системе регулирования, в системах с И- и ПИ-регуляторами при их фиксированных настройках и действии параметрических возмущений на объект управления. Параметры И- и ПИ-регуляторов рассчитывались по методу расширенных амплитудно-фазочастотных характеристик при заданной степени затухания переходного процесса в замкнутой системе, равной 0,95, при втором интегральном критерии качества и параметрах объекта регулирования  $k_{ог} = 6$ ;  $T_{ог} = 97,7$  с;  $\tau = 23,4$  с.

Как видно из табл. 6.2, при различных постоянных времени объекта управления переходные процессы автоматической системы регулирования с линейными регуляторами имеют различные качественные показатели.

Таблица 6.2

Сравнительные показатели качества переходного процесса

Показатели качества переходного процесса	Нечеткий регулятор качества			И-регулятор с фиксированной настройкой			ПИ-регулятор с фиксированной настройкой		
	$T_{об}, c$								
	50	97,7	150	50	97,7	150	50	97,7	150
$\psi$	0,95	0,95	0,95	0,98	0,95	0,9	0,42	0,96	0,96
$\sigma, \%$	22,9	22,5	22,6	9,8	22,5	31,1	79	38	40
$t_p, c$	470	585	1035	432	585	1070	517	136	295

Для оценки качества работы синтезированного регулятора воспользуемся формулой (5.6) для обобщенного показателя качества системы относительно степени затухания [1]. На рис. 6.4 представлены результаты моделирования для рассматриваемой системы автоматического регулирования и системы с ПИ-регулятором при изменении постоянной времени объекта управления в диапазоне от 50 до 150 с.

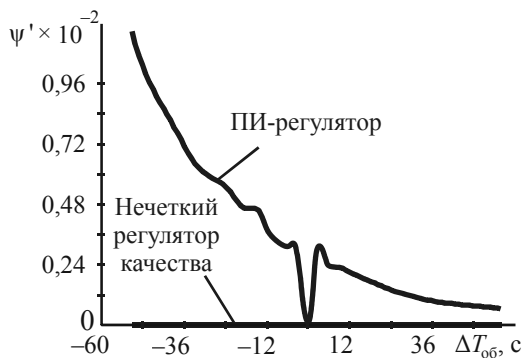


Рис. 6.4. Результаты моделирования системы автоматического регулирования при параметрическом возмущении: обобщенный показатель качества системы относительно степени затухания

Таким образом, рассмотренное управляющее устройство придает всей системе автоматического регулирования способность поддерживать на заданном уровне не только технологический параметр, но и динамику изменения его во времени, т.е. регулировать качество технологического процесса. Разработанный регулятор придает всей системе инвариантность по отношению к параметрическим возмущени-

ям (см. рис. 6.4.). Применение теории нечеткой логики позволило оценивать величину адаптирующего воздействия, которую необходимо ввести в систему автоматического регулирования, что позволило достигнуть поставленную цель управления за один цикл исследования. Регулятор может применяться совместно с различными технологическими объектами, динамические характеристики которых меняются как в узких, так и в широких пределах.

## 6.5. ВЫВОДЫ

Применение устройств управления, базирующихся на принципах теории нечетких множеств, позволяет использовать для целей управления информацию качественного характера, которую трудно формализовать при реализации традиционных законов регулирования. Полученные системы оказываются малочувствительными к возмущениям в определенном диапазоне и отличаются улучшенными характеристиками по сравнению с системами, функционирующими совместно с классическими регуляторами.

## ЛИТЕРАТУРА

1. *Козлов Ю.М., Юсупов Р.М.* Бесписковые самонастраивающиеся системы. – М.: Наука, 1969. – 456 с.
2. *Леоненков А.* Нечеткое моделирование в среде MatLab и fuzzyTECH. – СПб.: БХВ–Петербург, 2003. – 763 с.
3. *Математические модели технологических процессов и разработка систем автоматического регулирования с переменной структурой:* Сб. тр. Гинцветмета № 21 / Под общ. ред. Б.Н. Петрова. – М.: Металлургия, 1964. – 468 с.
4. *Мелихов А.Н., Бернштейн Л.С., Коровин С.Я.* Ситуационные советующие системы с нечеткой логикой. – М.: Наука, 1990. – 272 с.
5. *Нечеткие множества в моделях управления и искусственного интеллекта* / Под ред. Д.А. Поспелова. – М.: Наука, 1986. – 312 с.
6. *Петров Б.Н., Уланов Г.М., Гольденблат И.И., Ульянов С.В.* Теория моделей в процессе управления. – М.: Наука, 1978. – 224 с.
7. *Соловьев В.А., Владыко А.Г., Легенкин В.С.* Применение нечеткой логики в устройствах регулирования энергетическими объектами // *Электротехника и энергосберегающие технологии: Межвуз. сб. науч. тр.* – Владивосток: Изд-во ДВГТУ, 1998. – С. 125–133.
8. *Шидловский С.В., Светлаков А.А.* Нечеткая классификация признаков в системе контроля знаний // Тез. докл. региональной науч.-методич. конф. «Со-

временное образование: массовость и качество». – Томск: ТУСУР, 2001. – С. 57–59.

9. Шидловский С.В., Светлаков А.А. Перестраиваемые структуры в системах автоматического управления технологическими процессами, инвариантные к изменению динамических характеристик объекта // Электронные средства и системы управления: Материалы Междунар. науч.-практ. конф. – Томск: Изд-во Института атмосферы СО РАН, 2004. – Ч. 2. – С. 103–106.

10. *Fuzzy Logic Toolbox User's Guide* // The MathWorks, Inc., 1998.

# ИНФОРМАЦИОННО-ПОИСКОВЫЕ АСПЕКТЫ ПРИ СИНТЕЗЕ СТРУКТУРНОГО АВТОМАТА

---

---

### 7.1. ВВОДНЫЕ ПОНЯТИЯ

Важной частью любой системы управления является ее информационная составляющая. Информационные системы зачастую работают в условиях широкого изменения полезных составляющих входных сигналов. Использование фильтров с неизменной настройкой не всегда дает удовлетворительное выделение полезных сигналов в связи с переменностью параметров или недостаточно полным использованием априорных сведений о закономерностях изменения полезных сигналов.

Естественным выходом из сложившейся ситуации является применение принципов самонастройки с поисковыми составляющими; последние могут быть как регулярными, так и случайными. Ниже рассмотрим информационно-поисковую составляющую на примере информационно-поискового автомата (ИП-автомат).

### 7.2. РЕЖИМЫ ПОИСКА

Режимы поиска зависят от диапазона логических и вычислительных операций, выполняемых в данной информационно-поисковой системе (ИПС).

*Критерий выдачи* – это формальное правило, в соответствии с которым в информационном массиве определяются документы, подлежащие выдаче в ответ на поступивший в систему запрос.

Различают три вида критериев выдачи, формулируемых в терминах теории множеств (теоретико-множественные критерии). При этом поисковые образы документов и предписания запросов понимаются как множества лексических единиц информационно-поискового языка (ИПЯ).

*Критерий «на совпадение»*. Для выдачи требуется, чтобы лексические единицы поискового образа документа и поискового предписания совпали. Такое совпадение возможно в автоматизированных информа-

ционно-поисковых системах (АИПС) в частном случае (поиск по формальным признакам документов – автор, язык и т.д.), в ручных ИПС этот критерий применяется очень широко.

*Критерий «на включение»* (или «на вхождение»). В ответ на запрос выдаются те документы, поисковые образы которых включают целиком поисковое предписание запроса. Если поисковый образ документа представить в виде множества  $M_d$ , а поисковое предписание –  $M_q$ , то сообщение выдается, когда  $M_q \subset M_d$ . Этот критерий используется многими АИПС дескрипторного типа

*Критерий «на пересечение»* требует не полного, а частичного совпадения лексических единиц документа и запроса, т.е. пересечения поисковых образов. Математически этот критерий записывается в виде формулы  $M_q \cap M_d$ .

### 7.3. АНАЛИЗ КРИТЕРИЕВ ВЫДАЧИ ИНФОРМАЦИОННО-ПОИСКОВЫХ СИСТЕМ

Любой поисковый образ документа (ПОД) во всякой дескрипторной ИПС можно рассматривать как набор кодов, называемых поисковыми характеристиками. В ИПС, где применяются информационно-поисковые языки «без грамматики» [4, 5], поисковыми характеристиками являются коды дескрипторов. Если в ИПС системы имеются некоторые грамматические средства [1, 6], то поисковые характеристики представляются сложными кодами, включающими коды одного или нескольких дескрипторов, и кодами указателей роли и связи, идентификаторов, меток и т.п.

Поисковый массив любой дескрипторной ИПС можно рассматривать как множество  $N$ , элементами которого является ПОД.

Чтобы из множества  $N$  выделить только один элемент, необходимо задать информационный запрос, указывающий, какие поисковые характеристики должны содержаться в требуемом ПОД и какие – отсутствовать.

Согласно [10] математическая модель информационного запроса, требующего выдачи только одного ПОД, представляется выражением

$$f = \varphi \cdot \psi,$$

где  $\varphi$  – конъюнкция неинверсных аргументов, соответствующих поисковым характеристикам требуемого ПОД;  $\psi$  – конъюнкция инверсных аргументов, соответствующих поисковым характеристикам, не вошедшим в требуемый ПОД. Отсюда следует, что функция отдельного ПОД представляет собой член СДНФ  $n$  аргументов, где  $n$  – число различных возможных поисковых характеристик, использующихся при индексировании документов.

В общем случае любое подмножество элементов множества  $N$  может быть представлено дизъюнкцией тех членов СДНФ, которым соответствуют искомые ПОД.

Функцию, описывающую некоторое подмножество элементов множества  $N$ , условимся называть поисковой функцией. Критерий выдачи большинства ИПС характеризуется величиной  $K$ , где  $K$  – максимальное число различных поисковых характеристик одной заданной в системе поисковой функции. Если поисковая функция некоторого запроса содержит число аргументов, превышающее величину  $K$ , то эту функцию необходимо представить в виде нескольких самостоятельных поисковых функций, где каждая функция может быть задана критерием выдачи и реализована системой за одно обращение к поисковому массиву. Под выражением «одно обращение к поисковому массиву» будем подразумевать цикл операций, который необходимо выполнить ИПС при поиске документов (номеров документов) по информационному запросу, заданному в системе одной поисковой функцией. Для ИПС, реализованной на однопроцессорной ЭВМ, в указанный цикл входят операции, выполняемые процессором машины при сравнении поисковых характеристик ПОД с поисковыми характеристиками одной из заданных поисковых функций, и операции, связанные с анализом результатов сравнения (без учета операций вывода результатов поиска).

В многопроцессорных вычислительных системах число поисковых функций, по которым выполняются одновременные поисковые операции (за одно обращение к поисковому массиву), может быть соизмеримо с числом используемых в системе основных процессоров.

Булевы функции являются универсальным и одним из наиболее распространенных и удобных на практике математических средств для моделирования информационных запросов, содержащих произвольные сочетания числовых и нечисловых характеристик.

#### 7.4. ДИНАМИЧЕСКИЕ РЕЖИМЫ РАБОТЫ ИНФОРМАЦИОННО-ПОИСКОВЫХ СИСТЕМ

Критерий выдачи ИПС, работающий в динамическом режиме, условимся называть динамическим критерием выдачи (ДКВ).

В [11] выбран метод изменения запросов, известный под названием «обратная связь по релевантности», так как запросы автоматически корректируются на основе поступающей от пользователей информации о релевантности ранее выданных документов. В частности, при использовании обратной связи по релевантности предполагается, что по каждому поступившему в систему запросу производится предварительный поиск. Наибольшее количество документов, в наибольшей мере отвечающих запросу, выдается потребителю, который должен оценить некоторые из этих документов и разделить их на релевантные  $R$ , т.е. отвечающие его информационным потребностям, или нерелевантные  $S$ . Эти оценки релевантности возвращаются в систему и используются для изменения запросов таким образом, чтобы термины запросов, присутствующие в релевантных документах, «продвигались» (путем увеличения их веса), а термины, присутствующие в нерелевантных документах, одновременно «понижались».

В дальнейшем  $R$  ранее признанных релевантных документов и  $S$  нерелевантных используется для формулировки нового запроса  $q'$ , который предположительно будет более схож с релевантными и менее – с нерелевантными документами, чем первоначальный запрос  $q$  [8].

Можно выделить две разновидности ДКВ.

*Первая* характеризуется тем, что абонент обязан до поиска на основе своей информационной потребности полностью сформулировать исходный запрос (т.е. указать перечень поисковых признаков и логику поиска), который в дальнейшем, благодаря обратной связи по pertinентности, будет корректироваться и на основе которого будет осуществляться вторичный поиск. Такой ДКВ получил наибольшее распространение [2].

*Вторая* разновидность ДКВ характеризуется неполным формулированием запроса. Абонент перечисляет только поисковые характери-



ки и не указывает поисковую логику. В этом случае поиск осуществляется по заранее принятому критерию, одинаковому для всех запросов, а по результатам анализа выдачи определяется поисковая логика. Такой критерий предполагает гораздо бóльшую практическую ценность по сравнению с первой разновидностью ДКВ, поскольку при его использовании уменьшается нагрузка на абонента.

Обеим разновидностям ДКВ присущи недостатки, из которых как основные можно отметить следующие.

Первоначальный (пробный) поиск ведется в малой части массива, а результат корректировки запроса распространяется на весь массив. В общем случае структура массива не является однородной ни для какого запроса, поэтому предположение о том, что полнота и точность при просмотре массива по скорректированному запросу значительно возрастут, является сомнительным.

Пробная выдача может содержать незначительное число пертинентных документов, вследствие чего пробную выдачу потребуется увеличить. Анализ большой выдачи представляет собой неоправданную (и значительную) нагрузку на абонента.

При выполнении профиля запроса необходимо снова анализировать пробную выдачу и снова корректировать запрос.

При корректировке запросов, логика поиска которых представлена динамическим критерием, основанным на весовых коэффициентах, на каждую величину необходимо увеличить или уменьшить вес того или иного признака.

Таким образом, в связи с тем, что динамический критерий, основанный на принципах корректировки логики запросов (поисковой функции), ориентирован лишь на уточнение самого запроса и практически не учитывает динамику поискового массива, возникает необходимость разработки такого ДКВ, который обеспечивал бы выдачу наиболее релевантных документов после однократного просмотра поискового массива с учетом его структуры.

## 7.5. ИНФОРМАЦИОННО-ПОИСКОВЫЙ АВТОМАТ

В [15] приведена классическая схема ИП-автомата. Напомним, что логическим называется управление с помощью сигналов, принимающих

конечное число фиксированных значений, осуществляемое в соответствии с заданным алгоритмом [12]. В работах [3, 7, 9] отражены модернизированные варианты классической схемы ИП-автомата. Далее рассмотрим построение ИП-автомата на основе МЛМ. Во всех указанных вариантах присутствует поисковая составляющая, а сам алгоритм поиска информации состоит из двух последовательностей:

- 1) операции сравнения всех признаков поисковых образов документа со всеми признаками поисковых образов заданий (ПОЗ);
- 2) операции нахождения значения булевых функций по результатам сравнения.

Обобщенная структурная схема такого ИП-автомата приведена на рис. 7.1.

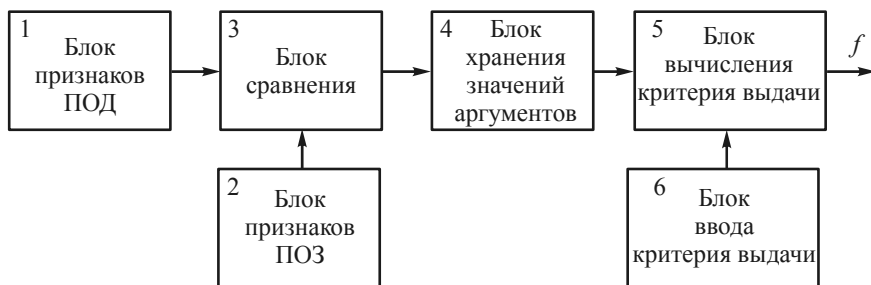


Рис. 7.1. Обобщенная структурная схема ИП-автомата: 1 – триггерный регистр, в который поступают коды признаков ПОД из внешнего запоминающего устройства (ЗУ); 2 – триггерный регистр для хранения кодов признаков ПОЗ; 3 – схема равенства; 4 – триггеры хранения результатов сравнения; 5 – логическая схема, моделирующая критерий выдачи; 6 – коммутационная среда либо триггерный регистр ввода булевой формулы

Наиболее важным в приведенной схеме является блок 5, поскольку он определяет функциональные возможности ИП-автомата относительно реализации того или иного критерия выдачи. Некоторое раскрытие функциональных блоков приведено на рис. 7.2, а на рис. 7.3 показана реализация ИП-автомата с помощью МЛМ [14].

Особенности работы рассматриваемого устройства покажем на следующих примерах.

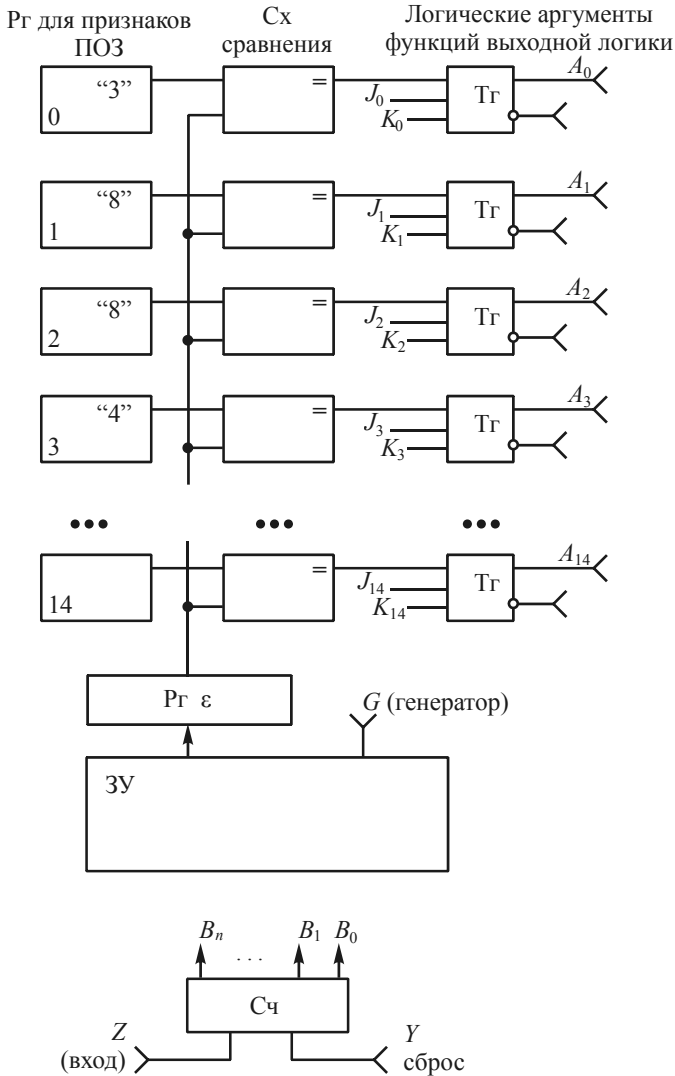


Рис. 7.2. Структурная схема ИП-автомата

Пример 1. На регистр  $\epsilon$  (Рг  $\epsilon$ ) последовательно поступают цифры десятичной системы. Схема должна выдавать сигнал, когда во

входной последовательности окажется число, например, «3884» (для выдачи этого числа один символ должен четко идти за другим).

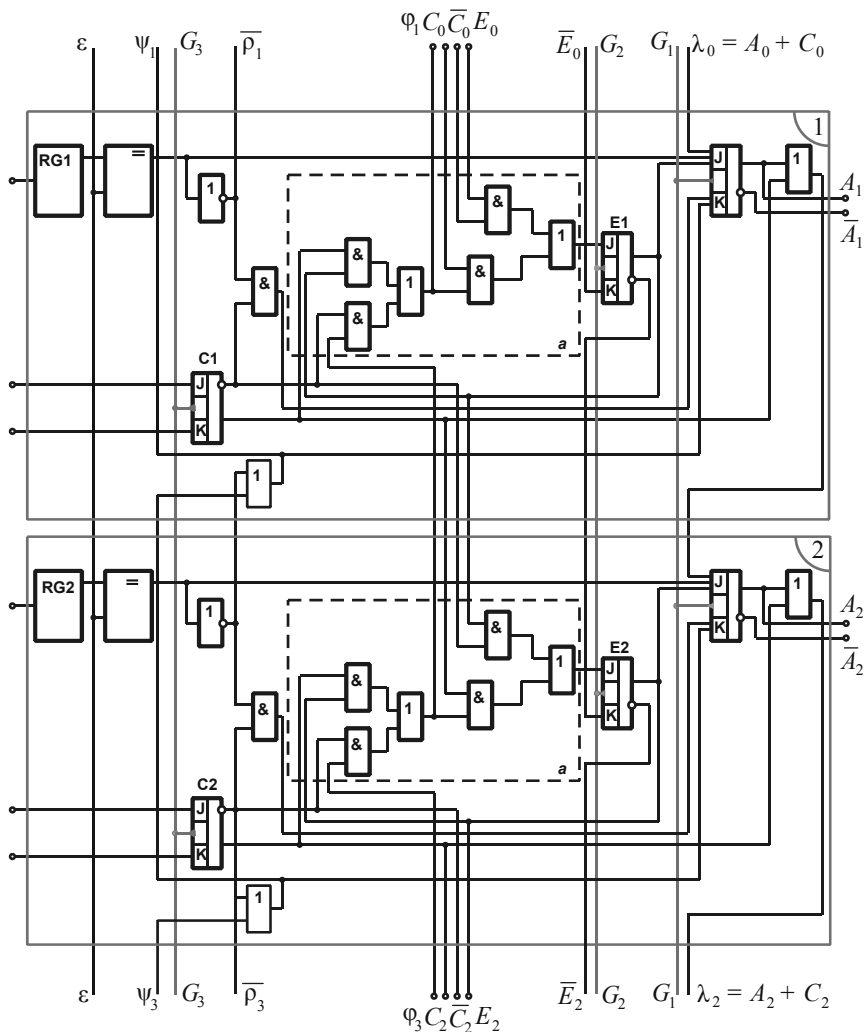


Рис. 7.3. Фрагмент функциональной схемы ИП-автомата, построенного на основе МЛМ

Поскольку первая цифра последовательности «3», то при поступлении ее из ЗУ в Рг  $\varepsilon$  счетчик (СЧ) должен получить импульс генератора ( $G$ ) (эти импульсы идут симметрично, с периодом смены чисел в регистре Рг  $\varepsilon$ ), следовательно,  $Z = \rho_0 G$ .

Если вторая цифра «8», то СЧ должен получить еще один импульс, следовательно,

$$Z = \rho_1 G \vee \rho_0 G = G(\rho_1 \vee \rho_0),$$

если же вторая цифра не «8», то СЧ необходимо сбросить в нуль, для этого заметим, что после прихода «3»  $A_0 = 1$ , следовательно,  $Y = A_0 \bar{\rho}_1$ .

Если третья цифра тоже «8», то

$$Z = \rho_1 G \vee \rho_0 G \vee \rho_2 G = G(\rho_1 \vee \rho_0 \vee \rho_2),$$

и если последняя цифра «4», то  $Z = G(\rho_1 \vee \rho_0 \vee \rho_2 \vee \rho_3)$ .

Предположим, что цифры «3» и «8» совпали ( $A_0 = 1$ ,  $A_1 = 1$ ), а третья цифра не «8», тогда

$$Y = A_0 A_1 \bar{\rho}_2 \vee A_0 \bar{\rho}_1,$$

если не совпала последняя цифра, то

$$Y = A_0 A_1 A_2 \bar{\rho}_3 \vee A_0 A_1 \bar{\rho}_2 \vee A_0 \bar{\rho}_1.$$

Если не совпала вторая цифра, то Тг  $A_1$  не перейдет в «1», а Тг  $A_0$  необходимо сбросить в «0», это можно сделать подачей импульса на вход  $K_0$ , т.е.  $K_0 = A_0 \bar{\rho}_1$ .

Если не совпала третья цифра, то необходимо сбросить Тг  $A_0$  и Тг  $A_1$ , т.е. подать импульс на вход  $K_1$  и  $K_0$ , т.к.  $K_0 = A_0 \bar{\rho}_1 \vee A_1 \bar{\rho}_2$ ,  $K_1 = A_1 \bar{\rho}_2$ .

Если не совпала четвертая цифра, то на входы  $K_0$ ,  $K_1$ ,  $K_2$  надо подать импульсы

$$K_0 = A_0 \bar{\rho}_1 \vee A_1 \bar{\rho}_2 \vee A_2 \bar{\rho}_3,$$

$$K_1 = A_1 \bar{\rho}_2 \vee A_2 \bar{\rho}_3,$$

$$K_2 = A_2 \bar{\rho}_3.$$

Функция выходной логики имеет вид  $f = A_3$  (при совпадении всех последовательностей цифр «3884»).

Анализ решения примера 1. Фактически автомат описывается тремя функциями, которые имеют вид

$$\begin{aligned} K_0 &= A_0\bar{\rho}_1 \vee A_1\bar{\rho}_2 \vee A_2\bar{\rho}_3, \\ K_1 &= A_1\bar{\rho}_2 \vee A_2\bar{\rho}_3, \\ K_2 &= A_2\bar{\rho}_3. \end{aligned} \quad (7.1)$$

Исходное состояние:

$$A_0 = A_1 = A_2 = A_3 = 0, \quad f = A_3 = 0.$$

1. Пришла не цифра «3» (и не «8», и не «4»), изменений в схеме нет.

2. Пришла цифра «8» (т.е. первая), тогда

$$\begin{cases} \rho_1 = 1; \\ \rho_2 = 1, \end{cases} \quad K_1 = A_1\bar{\rho}_2 \vee A_2\bar{\rho}_3 = 0 \quad \bar{1} \vee 0 \quad \bar{0} = 0.$$

3. Тг  $A_1$ , Тг  $A_2$  перейдут в «1».

4. Следующая цифра, допустим, «3», при этом

$$\begin{aligned} K_1 &= A_1\bar{\rho}_2 \vee A_2\bar{\rho}_3 = 1 \quad \bar{0} \vee 1 \quad \bar{0} = 1 \quad (\text{Тг } A_1 \text{ сбросился в «нуль»}), \\ K_2 &= A_2\bar{\rho}_3 = 1 \quad \bar{0} = 1 \quad (\text{Тг } A_2 \text{ тоже сбросился в «нуль»}). \end{aligned}$$

Тг  $A_0$  установился в «1», т.е. устанавливается начало нужной нам последовательности.

На синхровходы Тг одновременно поступают импульсы  $G$ . Тг меняет свое состояние после ухода синхроимпульса, так как срабатывает по заднему фронту.

5. Рассмотрим случай, когда первая цифра «4» (это последняя цифра заданной последовательности):

а)  $A_0 = A_1 = A_2 = A_3 = 0$ , до прихода «4» в Рг  $\varepsilon$ , так как считаем, что в начале Рг  $\varepsilon$  был нуль;

б) до прихода синхроимпульса  $\rho_3 = 1$ ,  $K_3 = 0$  (выясним, почему и как), т.е.  $A_3 = 1$ , а это равно  $f$ , что неверно!!!).

Для выхода из создавшегося положения необходимо сбрасывать Тг  $A_3$ , т.е. найти формулу  $K_3 = (\bar{A}_0 \vee \bar{A}_1 \vee \bar{A}_2)G$ .

**Пример 2.** Рассмотрим случай, когда ведется поиск по последовательности «8888», и пришла в Рг  $\epsilon$  «8»; все Тг  $A_0 \div A_3 = 1$ . Допустим, что опять пришла еще одна «8».  $K_0 = K_1 = K_2 = 0$  (см. по установке в «1» при значении первой «8»  $A_0 = A_1 = A_2 = A_3 = 1$ ,  $\rho_0 = \rho_1 = \rho_2 = \rho_3 = 1$ ).

Состояние не изменяется с приходом любого числа восьмерок.

Рассмотрим теперь случай, когда ведется поиск последовательности «8888», по второй цифре (первой была цифра «8») пришла любая цифра, отличная от восьмерки.

В этом случае все  $\rho = 0$ , а согласно системе (7.1) при  $A_0 \div A_3 = 1$ ,  $K_0 = K_1 = K_2 = 1$ , получим  $A_0 = A_1 = A_2 = 0$ , а  $A_3 = 1$ , при приходе третьей цифры не «8»,  $A_3$  установится в нуль.

Таким образом, получаем следующий результат (случай 1 и 2): если последовательность состоит из одинаковых цифр, то рассмотренная схема на нее реагирует *неправильно*.

**Пример 3.** Рассмотрим второй подход (с использованием аргументов  $A_0 \div A_4$ ). Берем последовательность примера 1 (т.е. «3884»). Для распознавания первой цифры, т.е. «3»,  $J_0 = 1$ ,  $K_0 = 0$ , и если она («3») пришла первой, Тг  $A_0$  установится в «1» ( $A_0 = 1$ ).

После второго синхроимпульса Тг  $A_1$  должен перейти в «1», если  $A_0 = 1$  и если  $\rho_1 = 1$ , следовательно,  $J_1 = A_0$ ; если после «3» не «8», т.е.  $\rho_1 = 0$ , то Тг  $A_0$  надо сбросить в нуль, это можно сделать импульсом  $K_0 = \bar{\rho}_1 A_0$ .

Аналогично рассуждая, записываем  $J_2 = A_1$ ,  $K_1 = \bar{\rho}_2 A_1$ , если после прихода двух цифр (т.е. «3» и «8») третья придет не «8», а, скажем, «5», то необходимо сбросить Тг  $A_0$  и Тг  $A_1$ , тогда

$$K_1 = K_0 = \bar{\rho}_2.$$

Рассмотрим случай, когда первой пришла цифра «4» (последняя из последовательностей), при этом  $\rho_3 = 1$ ,  $J_3 = A_2 = 0$ , Тг  $A_3$  остался в нуле. В случае прихода любой последовательности, отличной от «3884», автомат на нее не реагирует. В случае прихода первых трех цифр (т.е. «388»), когда четвертая не «4»,  $A_0 = A_1 = A_2 = 1$ ;  $A_2 = J_3 = 1$ ;  $\rho_3 = 0$ ;

Тг  $A_3 = 0$ ; при этой ситуации цифра, отличная от «4», проигнорируется: автомат ждет «4», и все цифры из последовательности ПОД, кроме «4», игнорируются, т.е. автомат работает на «выборке» «3884» из поступающих ПОД, т.е. автомат описывается следующей системой выходных аргументов (чтобы обойти этот момент, необходимо предыдущую информацию сбросить всю, т.е. после второго синхроимпульса  $K_2 = K_1 = K_0 = \bar{\rho}_3$ ):

$$\begin{cases} J_0 = 1, & K_0 = \bar{\rho}_1 \vee \bar{\rho}_2 \vee \bar{\rho}_3; \\ J_1 = A_0, & K_1 = \bar{\rho}_2 \vee \rho_3; \\ J_2 = A_1, & K_2 = \rho_3; \\ J_3 = A_2; \\ f = A_3. \end{cases}$$

Изъян наступает при поступающей последовательности «3883884», когда с приходом второй «3» мы теряем первую, так как  $\rho_3 = 0$  (мы ищем последовательность «3884»), а с приходом четвертой цифры, отличной от «4», все сбрасывается в «ноль». Чтобы обойти этот случай, необходимо на один из трех входов  $K_0$  через инвертор подать  $\rho_0$ .

### Замечания

*Замечание 1.* Для триггеров  $E$  (на 6 ячеек):

$$\begin{aligned} J_0 &= C_0 E_0 \vee \bar{C}_0 \Phi_1, & K_0 &= 1; \\ J_1 &= E_0 \bar{C}_0 \vee C_0 (C_1 E_1 \vee \bar{C}_1 \Phi_2), & K_1 &= \bar{E}_0; \\ J_2 &= E_1 \bar{C}_1 \vee C_1 (C_2 E_2 \vee \bar{C}_2 \Phi_3), & K_2 &= \bar{E}_1; \\ J_3 &= E_2 \bar{C}_2 \vee C_2 (C_3 E_3 \vee \bar{C}_3 \Phi_4), & K_3 &= \bar{E}_2; \\ J_4 &= E_3 \bar{C}_3 \vee C_3 (C_4 E_4 \vee \bar{C}_4 \Phi_5), & K_4 &= \bar{E}_3; \\ J_5 &= E_4 \bar{C}_4 \vee C_4 (C_5 E_5 \vee \bar{C}_5 \Phi_6), & K_5 &= \bar{E}_4; \end{aligned}$$

$$\begin{cases} J_{E_i} = E_{i-1} \bar{C}_{i-1} \vee C_{i-1} (C_i E_i \vee \bar{C}_i \Phi_{i+1}); \\ K_{E_i} = \bar{E}_{i-1}. \end{cases}$$



*Замечание 2.*

Для триггеров  $A$  (на 6 ячеек);  
единичные входы:

$$J_0 = \rho_0 E_0;$$

$$J_1 = (A_0 \vee C_0) \rho_1 E_1;$$

$$J_2 = (A_1 \vee C_1) \rho_2 E_2;$$

$$J_3 = (A_2 \vee C_2) \rho_3 E_3;$$

$$J_4 = (A_3 \vee C_3) \rho_4 E_4;$$

$$J_5 = (A_4 \vee C_4) \rho_5 E_5;$$

...

$$J_i = (A_{i-1} \vee C_{i-1}) \rho_i E_i.$$

Для триггеров  $A$ . Нулевые входы:

$$K_0 = \bar{\rho}_0 \bar{C}_0 (\psi_1 \vee \bar{\rho}_1);$$

$$K_1 = \bar{\rho}_1 \bar{C}_1 (\psi_2 \vee \bar{\rho}_2);$$

$$K_2 = \bar{\rho}_2 \bar{C}_2 (\psi_3 \vee \bar{\rho}_3);$$

$$K_3 = \bar{\rho}_3 \bar{C}_3 (\psi_4 \vee \bar{\rho}_4);$$

$$K_4 = \bar{\rho}_4 \bar{C}_4 (\psi_5 \vee \bar{\rho}_5);$$

$$K_5 = \bar{\rho}_5 \bar{C}_5 (\psi_6 \vee \bar{\rho}_6);$$

для последней ячейки  $\psi_6 = \rho_6 = 1$ ;

$$K_i = \bar{\rho}_i \bar{C}_i (\psi_{i+1} \vee \bar{\rho}_{i+1}).$$

*Замечание 3.* Формула  $J_{E_0}$  имеет вид

$$J_{E_0} = \bar{C}_0 \{C_1 E_1 \vee \bar{C}_1 \{C_2 E_2 \vee \bar{C}_2 [C_3 E_3 \vee \bar{C}_3 (C_4 E_4 \vee \bar{C}_4 C_5 E_5)]\}\}.$$

*Замечание 4.* Для реализации режима работы информационно-поисковой машины «Квантор» [13] необходимо:

а) в регистр  $C$  ввести число «111111» – все единицы располагаются на триггерной части (см. рис. 7.3);

б) в регистр  $C$  ввести число «000000». Во всех регистрах записалась одна последовательность (в данном случае «012345»);

в) в регистр  $C$  ввести число

$$\begin{array}{cccccc} C5 & C4 & C3 & C2 & C1 & C0 \\ \langle 0 & 0 & 0 & 1 & 0 & 0 \rangle; \end{array}$$

г) в регистр  $C$  ввести код

$$\langle 101010 \rangle.$$

## 7.6. ВЫВОДЫ

Представленные варианты ИП-автомата показывают многоцелевую направленность разработанного устройства, осуществляющего реализацию конкретного случая с помощью перестройки выбранной структуры. На основе выбранных МЛМ можно построить различные проблемно-ориентированные комплексы, настраивающиеся на реализацию конкретного алгоритма из заданного класса.

## ЛИТЕРАТУРА

1. *Алексенко А.Г.* Основы микросхемотехники. Элементы морфологии микроэлектронной аппаратуры. – М.: Сов. радио, 1971. – 352 с.
2. *Баранов С.И.* Синтез микропрограммных автоматов. – Л.: Энергия, 1979. – 136 с.
3. *Васильев С.Н. и др.* Интеллектуальное управление динамическими системами. – М.: Физматлит, 2000. – 352 с.
4. *Визирев И.С.* и др. А.с. 807270 (СССР). Ячейка вычислительной среды // Б.И. – 1981. – № 7.
5. *Высоцкий В.Ф., Алексеев В.И., Панчин В.Н. и др.* Цифровые фильтры и устройства обработки сигналов на интегральных схемах: Справочное пособие / Под ред. В.Ф. Высоцкого. – М.: Радио и связь, 1984. – 216 с.
6. *Глушков В.М.* Синтез цифровых автоматов. – М.: Наука, 1962. – 467 с.
7. *Латко А.В., Ченцов С.В.* Непараметрические системы обработки информации. – М.: Наука, 2000. – 350 с.
8. *Малашевич Б.М.* Ассоциативные параллельные процессоры и микропроцессоры // Микроэлектроника и полупроводниковые приборы / Под ред. А.А. Васенкова и Я.А. Федотова. – М.: Сов. радио, 1979. – Вып. 4. – С. 70–82.
9. *Методы робастного, нейро-нечеткого и адаптивного управления: Учебник* / Под ред. Н.Д. Егупова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. – 744 с.
10. *Михайлов А.И., Черный А.И., Гиляревский Р.С.* Основы информатики. – М.: Наука, 1968. – 756 с.
11. *Солтон Дж.* Динамические библиотечно-информационные системы. – М.: Мир, 1979. – 560 с.
12. *Шалыто А.А.* Логическое управление. Методы аппаратной и программной реализации алгоритмов. – СПб.: Наука, 2000. – 780 с.
13. *Шевелев Ю.П.* Логика поиска документальной информации. – Томск: Изд-во Том. ун-та, 1980. – 123 с.
14. *Шидловский С.В.* Перестраиваемые структуры на многофункциональных логических модулях // Информационные системы: Труды постоянно действующей науч.-техн. школы-семинара студентов, аспирантов и молодых специалистов «Информационные системы мониторинга окружающей среды». Вып. 2. – Томск: ТУСУР, 2003. – С. 105–117.
15. *Энциклопедия кибернетики* / Под ред. В.М. Глушкова и др. – Киев, 1975. – Т. 1, 2.

## КОРПОРАТИВНЫЙ ПОРТАЛ НА ОСНОВЕ ПРИНЦИПА ПЕРЕСТРАИВАЕМЫХ СТРУКТУР

---

---

### 8.1. ВВОДНЫЕ ПОНЯТИЯ

Развитие наукоемких технологий, включая Web-технологии, на современном этапе требует нового подхода к созданию современных управляющих сред, одним из вариантов которого является создание корпоративного портала. Под корпоративным порталом далее будем понимать корпоративный Web-сайт, предназначенный для внутреннего пользования и предоставляющий сотрудникам организации доступ к корпоративной информации, площадкам аппаратно-программного обеспечения, а также к ограниченному количеству внешних Web-сайтов.

Для взаимодействия конечных пользователей с порталом необходимо организовать обмен информацией между ними. Здесь существует два наиболее ярких способа обмена информацией:

- в реальном масштабе времени;
- в виде подачи запросов и протоколов отчетности.

В первом случае, для проведения комфортной работы, пользователь должен обладать информационными каналами связи с большой пропускной способностью. Как правило, это возможно только при работе в определенной части сети, если не вводить дополнительные функциональные возможности корпоративной среды.

В случае второго подхода необходимо создание специального программного обеспечения и оснащение им вычислительных средств всех пользователей. С этим связан ряд неудобств как для пользователя, так и для рабочей группы создания корпоративного портала.

Существенным достоинством первого способа является возможность доступа к корпоративному portalу стандартными средствами, поддерживающими протокол HTTP (Hyper Text Transport Protocol – протокол для передачи данных в виде гипертекста), которыми оснащены все современные операционной системы.

Как показывает мировой опыт работы с электронными документами [4], пользователю, как правило, необходима какая-либо часть запрашиваемого документа, но приложения, функционирующие в гиперпространстве, как правило, возвращают на запрос весь документ с избыточными данными, что и приводит к повышенной загрузке информационных каналов.

Данная проблема решается при использовании технологии XML (eXtensible Markup Language – расширяемый язык разметки). Далее рассмотрим более подробно, какие дополнительные возможности при разработке автоматических систем, функционирующих в гиперпространстве, дает вышеупомянутая технология.

## **8.2. XML КАК ИНСТРУМЕНТАРИЙ СОЗДАНИЯ ГИПЕРМОДЕЛЕЙ ОДНОРОДНЫХ СТРУКТУР**

В настоящее время большинство систем автоматического управления сталкивается с рядом трудностей, которые зачастую связаны с невозможностью приспособляться к окружающим условиям. Применение специализированных процессоров, способных вывести систему из сложившейся ситуации, получило довольно широкое распространение. В работах [10, 12] предлагается использовать для построения этих спецпроцессоров многофункциональные логические модули с однородной структурой, позволяющие создавать универсальные электронные устройства с перестраиваемой структурой, которые за счет программной настройки в процессе или перед началом работы могут эффективно решать различные задачи.

Системам автоматического управления, функционирующим в гиперпространстве, для создания подобных вычислительных сред требуется некая платформа, способная впитать всю простоту и мощь однородных структур.

Язык разметки XML описывает и структурирует содержание документа XML или пакета данных, более известного как сущность XML. Эта разметка состоит из тегов (ячеек) и содержит ссылки на специальные символы или текстовые макросы, а также передает специальные инструкции программному обеспечению приложения и комментарии редакторам документа [5].

Правильно оформленные документы XML определяются в форме иерархического дерева, причем каждый документ имеет один и только

один корневой узел, называемый сущностью (корнем) документа. Этот узел может содержать команды обработки и/или комментарии и всегда содержит ветвь элементов, корень которой называется элементом документа. Этот элемент порождает все остальные элементы документа и не может содержаться ни в каком другом элементе. Таким образом, XML-документ можно представить в виде однородной структуры элементарных ячеек (тегов), соединенных иерархическим образом в соответствии с определенными правилами.

Обобщенную структуру объекта данных XML можно представить в виде двухуровневого обобщенного каскада (рис. 8.1, а). Вкладывая каскад, например в однородную линейную структуру, мы при определенной настройке соответствующих ячеек можем добиться реализации желаемой функции. На рис. 8.1, б приведен пример реализации исходного обобщенного каскада, описываемый формулой [7]:

$$f = [(x_1 * x_2) * (x_3 * x_4)] * (x_5 * x_6),$$

где \* – необходимые логические операции.

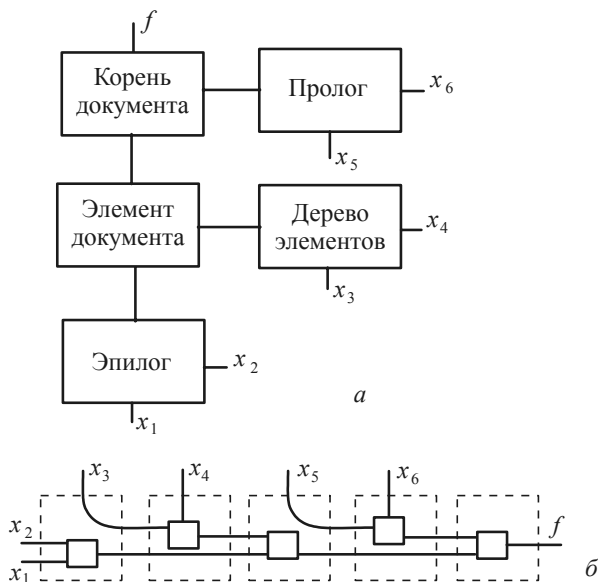


Рис. 8.1. Пример реализации структуры объекта данных XML в двухканальной однородной структуре

Рассмотренные выше принципы проиллюстрируем на примере образовательных технологий (рис. 8.2).

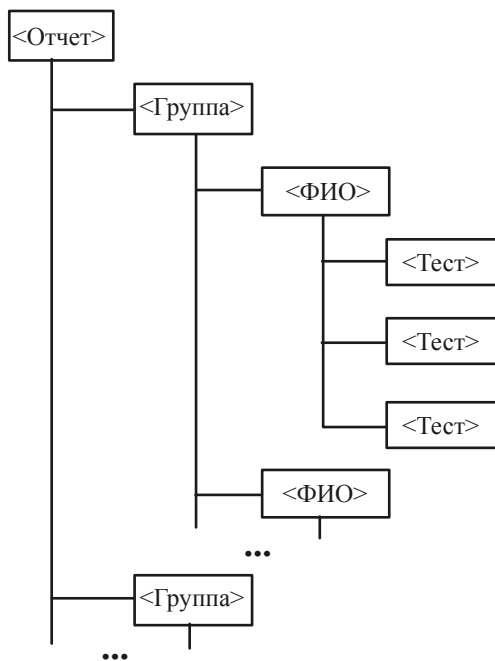


Рис. 8.2. Пример журнала отчетности в XML-форме

Результаты тестирования принадлежат обучаемым, прошедшим этот тест, а они, в свою очередь, записаны в соответствующую группу, которая входит в список групп, прошедших тестирование.

Физически выполненное тестовое задание не может частично принадлежать одному обучаемому, а частично другому. Кроме того, обучаемый может заниматься только в одной группе, а не в двух одновременно. Наконец, список группы должен быть размещен в журнале отчетности.

Данный пример показывает чрезвычайно важное ограничение, налагаемое на элементы языком XML, – их правильную вложенность. Здесь можно провести аналогию с физическим объектом, более того, элементы XML представляют собой объекты, которые должны подчиняться тем же правилам, что и физический объект [5].

Немаловажным достоинством XML-технологии является способность трансформировать. При работе на различных платформах и взаимодействиях с другими приложениями данный аспект очень важен, так как не все имеют возможность использовать для своих данных ту же структуру, что и мы.

Сгенерированный приложением или сохраненный в текстовом файле XML-документ содержится в фиксированном формате. Хотя XML-документ не зависит от платформы и может передаваться между различными частями приложения, в некоторых случаях требуется информация из других структур. Кроме того, может потребоваться трансформировать динамически структуру документа в интерактивный документ, например, для того, чтобы привести ее в соответствие с запросом пользователя.

Выделим три основные категории трансформации:

- структурные – преобразование структуры данных из одного словаря XML в другой;
- создание динамических документов – у пользователя появляется возможность изменять порядок, сортировать и фильтровать части документа XML;
- трансформация в язык формирования изображения – подготовка документа для визуального представления в какой-либо форме браузера пользователя.

Схема структуры трансформации части документа XML представлена на рис. 8.3.

Информация при попадании в ячейки XML, соединенные определенным образом по наборам правил трансформации, осуществляет настройку и перенастройку модели на все заданные алгоритмы функционирования.

Проиллюстрируем, как данный подход может отразиться на вышеприведенном примере (см. рис. 8.2). При желании пользователь может произвести сортировку, фильтрацию результатов тестирования в таблице и/или отобразить их в виде графика успеваемости за промежуток времени. В данном случае достигается интерактивность, т.е. пользователю предоставляется возможность выбрать и влиять на конечный результат функционирования системы. При этом нет необходимости обращаться к серверу, так как вся необходимая информация уже находится у пользователя портала, нужно только выбрать форму представления

данных, в которой заложена информация о выборе стратегии изменения структуры гипермодели. Последняя информация перестраивает структуру на один из заранее определенных алгоритмов управления. В итоге у пользователя появляется требуемая информация в желаемой форме представления. Этот нюанс также позволяет снизить нагрузку на информационные каналы.

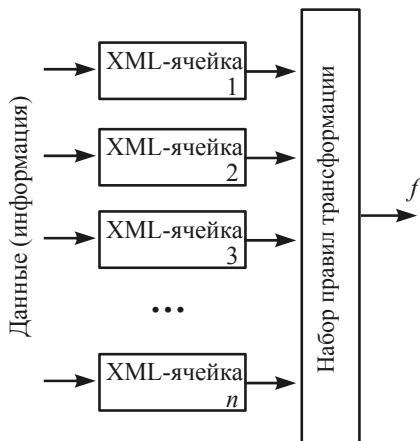


Рис. 8.3. Структура трансформации данных в XML-технологии

Таким образом, при построении аппаратно-программной площадки рассматриваемая технология позволяет создавать полноценные приложения, способные функционировать в гиперпространстве и взаимодействовать с другими приложениями, использующими иные структуры при передаче данных.

### 8.3. СХОДСТВО ПРОЦЕССОВ В СИСТЕМАХ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ И ОБУЧЕНИЯ

В настоящее время планирование процесса обучения осуществляется главным образом на основе эмпирически накопленных знаний, экспертных оценок, формировавшихся на основе самооценок и самоанализа. Сложный и многогранный характер обучения не позволяет делать какие-либо однозначные утверждения относительно его результата.

Для организации обучения необходимо иметь информацию об: исходных или начальных знаниях и умениях обучаемого; особенностях и



характере процесса их формирования; знаниях и умениях, приобретаемых в процессе обучения; критериях и способах контроля результатов обучения, а также способах коррекции процесса при отклонении от цели. Особое значение в этой системе имеет обратная связь, которая тесно связана с мотивацией и активизацией обучения. Обратная связь в процессе обучения, реализуя контроль и самоконтроль, направлена на выявление соответствия достигнутых результатов намеченным целям.

Наиболее важным компонентом системы организации обучения являются критерии обучения, благодаря которым соблюдается постоянство в оценке знаний и умений. При формулировании критериев необходимо учитывать условия протекания реального процесса обучения и ограничения, основывающиеся на существующих теоретических представлениях, экспериментальных данных и т.п.

Применение новых технологий и технических средств в организации обучения видоизменяет характер деятельности обучаемых; требует более глубокого изучения самого процесса обучения.

Использование новых технологий и технических средств требует определенной стандартизации, т.е. преподаватель, управляющий процессом обучения с помощью технических средств, должен иметь перечень указаний, в соответствии с которыми он организует свою деятельность и деятельность обучаемых. Индивидуальный, эмпирически сформировавшийся опыт преподавателя уступает место осознанным закономерностям учебно-воспитательной деятельности. В связи с этим возникла задача представления образовательного процесса в виде динамической системы.

Динамическая система в общем случае представляет собой совокупность организационных, технических и педагогических мероприятий, направленных на оптимальную индивидуализацию и управляемость процессом обучения. Система позволяет создавать учебные программы следующих режимов работы: обучение, контроль, справочно-информационный поиск, диалоговые вычисления, моделирование, программирование, управление системой исследования и др.

Когда говорят об обучении, то всегда имеют в виду существование той или иной цели, которая в результате обучения должна быть достигнута. В общей форме цель обучения представляет собой то состояние, к которому должна прийти обучающаяся система в результате обучения.

Необходимость в обучении возникает всякий раз, когда имеющаяся в нашем распоряжении априорная информация неполна. Тот или иной вид обучения зависит от степени неполноты априорной информации.

Различают два вида обучения: обучение с поощрением и обучение без поощрения [6].

При обучении с поощрением предполагается, что в каждый момент времени мы заранее знаем желаемую реакцию обучающейся системы и используем разность между желаемой и действительной реакцией, т.е. ошибку обучающейся системы, для изменения в нужном направлении ее свойств.

При обучении без поощрения заранее не известна желаемая реакция обучающейся системы, и в явном виде ее невозможно сформулировать, а значит, и использовать ошибку этой системы для изменения ее свойств.

Обобщая вышесказанное, можно провести следующую аналогию. Обучение с поощрением соответствует очному обучению, при котором преподаватель может ответить на все вопросы, возникающие у обучаемого. Обучение без поощрения соответствует заочному, т.е. обучению по неким методическим указаниям, учебным пособиям, составленным преподавателем, и обучаемый не имеет непосредственной возможности выяснить неясные вопросы.

Покажем сходства между процессом обучения и технологическим процессом, описываемым с помощью аппарата теории автоматического управления. Представим обучаемого в качестве объекта управления, а преподавателя – в качестве устройства управления (рис. 8.4, где  $s$  – величина, соответствующая в теории автоматического управления сигналу задания, в процессе обучения – цели обучения). В дальнейшем по тексту в процессе установления соответствия между процессом обучения и технологическим процессом, описываемым с помощью теории автоматического управления, примем следующее обозначение: термины, находящиеся в скобках, относятся к теории автоматического управления, а ставящиеся им в соответствие высказывания находятся рядом вне скобок. Величина  $f$  – внешние возмущающие факторы, влияющие на обучаемого (внешнее возмущающее воздействие, действующее на

объект управления);  $y$  – результат, полученный от обучаемого (выходная управляемая величина);  $z$  – контроль обучения (ошибка управления), достигаемый с помощью отрицательной обратной связи, т.е. установление соответствия достигнутых результатов в процессе обучения заданным критериям и принятие необходимых решений;  $u$  – поощрение или рекомендации (управляющее воздействие);  $g$  – информация, воспринимаемая обучаемым с учетом возмущающих факторов (величина, характеризующая сумму управляющего воздействия и внешнего возмущения).

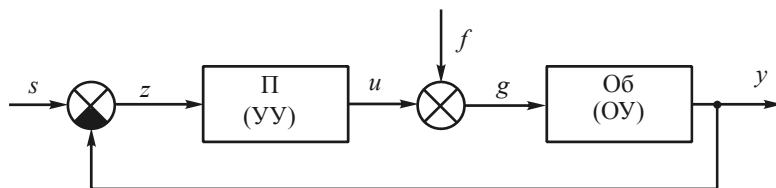


Рис. 8.4. Система автоматического управления технологическим процессом:

П (УУ) – преподаватель (устройство управления);

Об (ОУ) – обучаемый (объект управления)

Процесс обучения можно рассматривать как процесс приближения динамической модели, формирующейся в сознании обучаемого, к модели содержания изучаемого предмета (классическое выделение ошибки между эталонной моделью и объектом управления). Структурная схема этого процесса приведена на рис. 8.5.

Активность обучаемых является основным фактором, влияющим на эффект процесса обучения. Она очень чувствительна к изменениям условий обучения как в положительном смысле (повышение активности), так и в отрицательном (снижение и торможение). Взаимосвязь между активностью и достигаемым результатом выражается перевернутой параболой [1]. Эта кривая показывает, что результаты достигают максимума при определенном среднем уровне активности и удаляются от максимума тем дальше, чем больше уровень активности отстает от своего оптимального значения.

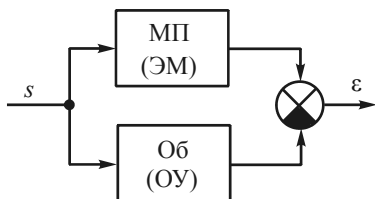


Рис. 8.5. Структура выделения ошибки с помощью эталонной модели: МП (ЭМ) – модель предмета (эталонная модель); Об (ОУ) – обучаемый (объект управления);  $s$  – цель обучения (сигнал задания);  $\varepsilon$  – ошибка в явной форме (сигнал рассогласования)

Таким образом, преподаватель (устройство управления) в процессе обучения должен проводить непрерывную оценку обучаемого (объекта управления) на основе своих знаний о предметной области и на основе этой информации выдавать рекомендации, направляющие обучаемого к цели. Исходя из этого, схему, представленную на рис. 8.4, можно преобразовать (рис. 8.6).

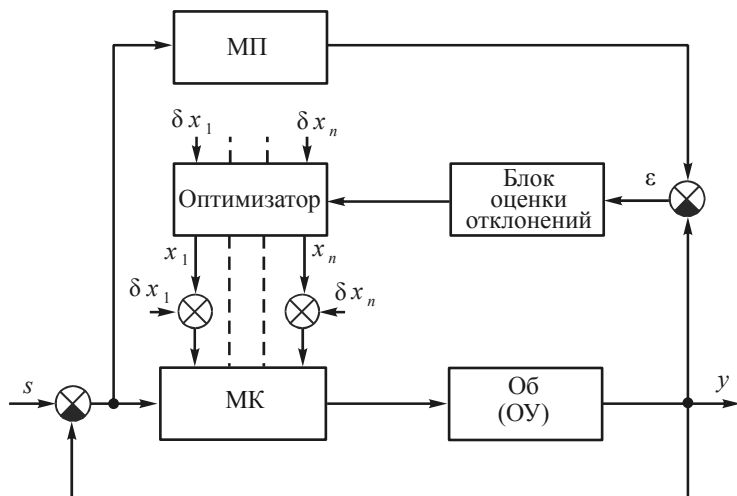


Рис. 8.6. Обобщенная схема системы обучения: МК – модель контроля учебно-познавательной деятельности

Необходимо определить цель одновременно для обучаемого и модели предмета. Следует заметить, что применение в этой модели прин-

ципов перестраиваемых структур позволяет использовать информацию из различных предметных областей. Результаты обучаемого и модели содержания учебного материала сопоставляются, и из их различий формируется оценка отклонений.

На модель контроля учебно-познавательной деятельности (корректирующее устройство), помимо основных рекомендаций по достижению требуемой цели  $x_i$  ( $i = 1, 2, \dots, n$ ), подаются наводящие вопросы (поисковые составляющие)  $\delta x_i$ .

Для поддержания активности обучаемого применяется оптимизатор, который, используя информацию о наводящих вопросах (поисковых составляющих)  $\delta x_i$ , выявляет отклонения от экстремума оценки и вырабатывает требуемые рекомендации, тем самым сводя ошибку между знаниями обучаемого и моделью содержания учебного материала к минимуму, а значит, повышает эффект процесса обучения.

Таким образом, использование данного подхода позволяет создавать гипермодели, имитирующие процесс обучения, что наделяет разрабатываемый корпоративный портал широкими возможностями при функционировании в условиях неполной информации.

#### 8.4. ОБОБЩЕННАЯ СТРУКТУРА ПОРТАЛА

Исходя из рассмотренных в предыдущем разделе подходов к синтезу систем управления, выделим основные модули, на которые опирается разработанный корпоративный портал:

- базы знаний;
- модуль контроля;
- модуль лабораторного практикума;
- модуль оперативного обмена данными.

При построении подобных систем обычно возникает вопрос, что такое знания и чем они отличаются от обычных данных. Под данными будем понимать информацию, полученную в результате наблюдений или измерений отдельных свойств, характеризующих объекты, процессы и явления предметной области. Под знаниями же будем понимать связи и закономерности предметной области (принципы, модели, законы), полученные в результате практической деятельности и профессионального опыта, позволяющего специалистам ставить и решать задачи в данной области.

При работе со знаниями ключевым этапом является формирование поля знаний (выявление и определение объектов и понятий предметной области, их свойств и связей между ними, а также представление их в наглядной и интуитивно понятной форме).

Обобщенную синтаксическую структуру поля знаний ( $Pz$ ) можно представить как

$$Pz = (I, O, M),$$

где  $I$  – структура исходных данных, подлежащих обработке и интерпретации в системе;  $O$  – структура выходных данных, т.е. результат работы системы;  $M$  – операционная модель предметной области, на основании которой происходит модификация  $I$  в  $O$ .

Без тщательной проработки поля знаний не может быть речи о создании базы знаний. В разработанном портале база знаний включает термины, определения, утверждения, иллюстративные примеры, типовые задачи и методы их решения, а также интерактивные видеоматериалы и гипердокументы предметной области.

Поскольку система знаний ценна не сама по себе, а именно возможностями ее использования, а использовать эту систему можно лишь выполняя над ней те или иные операции, алгоритмика которых определяется особенностями языка представления знаний, любой современный метод представления знаний является совокупностью взаимосвязанных средств формального описания знаний и оперирования этими описаниями. Под системой знаний будем понимать совокупность знаний, хранящихся в вычислительной среде и необходимых для решения комплекса прикладных задач конечным пользователем.

Для такого манипулирования представленными знаниями созданы модули контроля и лабораторного практикума. Используя эти модули, конечный пользователь испытывает потребность в получении недостающей информации из той или иной предметной области базы знаний.

В модуле контроля собран ряд методов тестирования и оценки знаний. Для возможности оценки знаний конечного пользователя вычислительной средой предложен подход, базирующийся на оценке основных показателей качества переходного процесса в системах автоматического управления.

Данный подход рассматривается по аналогии с процессами, протекающими в обучении и системах автоматического управления. Под основными показателями качества управления будем понимать величины, характеризующие систему в переходном процессе, вызванном определенными внешними воздействиями [9]:

- 1) ошибка регулирования;
- 2) время регулирования;
- 3) перерегулирование;
- 4) показатель колебательности.

График переходного процесса системы автоматического регулирования по каналу задания представлен на рис. 8.7.

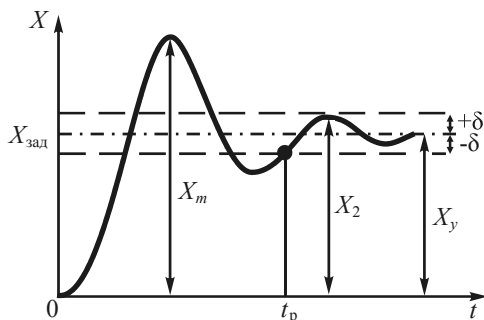


Рис. 8.7. Пример переходного процесса, иллюстрирующий показатели его качества

Величину перерегулирования можно вычислить по формуле

$$\sigma = \frac{x_m - x_y}{x_y} 100\% .$$

Величина перерегулирования — максимальная ошибка, которую допускает конечный пользователь относительно достигнутой цели (задания).

Время регулирования  $t_p$  — это время, за которое результаты конечного пользователя начинают отличаться от достигнутой цели менее, чем на заданное  $\delta$ , где  $\delta$  — точность. Иногда может возникать ошибка, которая не исчезает даже по истечении длительного интервала времени,

это статическая ошибка регулирования  $\epsilon_{ст}$ . Данная ошибка не должна превышать некоторой наперед заданной величины.

Степень затухания  $\Psi$  показывает, насколько быстро конечный пользователь достиг поставленной цели и вычисляется по формуле

$$\Psi = \frac{x_m - x_2}{x_m}.$$

После того как конечный пользователь ответит на все вопросы, выданные модулем контроля, происходит передача результатов в формате XML на сервер. В этих результатах заложены идентификационный номер конечного пользователя, идентификационный номер вопроса, идентификационный номер выбранного конечным пользователем ответа и количество времени, потребовавшегося ему для этой операции. На сервере происходит обработка полученной информации, преобразование ее в нужную форму и вычисление количественных показателей качества знаний конечного пользователя.

Модуль лабораторного практикума построен на базе Java-апплетов по принципу, базирующемуся на имитационном моделировании. Приоритет этому принцип был отдан не только из-за возможности анализа систем в условиях большой размерности и неполной информации о структуре системы, но и по причине доступности методологии для широкого круга специалистов.

Наряду с универсальным характером и высоким уровнем детализации механизмов функционирования систем дополнительное достоинство имитационного моделирования заключается в возможности синтеза на его основе комплексных моделей, сочетающих преимущества различных подходов.

Для преодоления априорной неопределенности структура имитационных моделей предполагает использование информации эксперта, общих сведений о реальной системе, включая ее аналитическую модель, содержит блоки имитации и обработки их результатов. Выбор той или иной структуры для конкретных условий определяется уровнем исходной информации, что может служить основой классификации методов имитационного моделирования и возникающих при этом задач [3].

Приведем общую структуру имитационной модели, заложенную в программно-аппаратную лабораторную площадку.



Принято под структурой математической модели  $y = F(X, < s >)$  понимать алгоритм  $< s >$ , определяющий порядок расчета выходных переменных  $Y$  системы по значениям ее входов  $X$ .

Структура, как способ организации целого из составных частей, формируется в процессе декомпозиции системы из множества  $A$  моделей ее подсистем, элементов и схем  $R$  сопряжения между ними:

$$< s >: (M_a, a \in A, R(A)). \quad (8.1)$$

Степень конкретизации структуры модели, а следовательно и содержание множества  $A$  и характер  $R$ , зависят от уровня исходной информации об исследуемой системе.

В имитационных моделях для преодоления априорной неопределенности структура (8.1) претерпевает следующие изменения [49]:

– вводится пара  $(A_n, R_n)$ , отражающая структуру реальной системы, либо последовательность действий субъекта, принимающего решения. При этом не исключается использование  $(A, R)$  и их комбинации с  $(A_n, R_n)$ ;

– на основе  $(A_n, R_n)$  организуется алгоритм имитации  $I(A_n, R_n)$  с целью получения данных  $V$  для восполнения недостающих сведений о системе;

– дополнительно строится алгоритм  $\psi_n(V)$  обработки результатов имитации  $V$ , вид которого зависит от вида показателей эффективности системы. Если показатели являются функцией параметров системы, то при ее восстановлении также возможно использование принципов имитации.

Тогда, по аналогии с (8.1), структура имитационной модели представима в виде

$$< s_n >: \{I(M_a, a \in A_n, R_n(A_n)), \psi_n(V)\}. \quad (8.2)$$

Используя эти принципы, конечный пользователь получает доступ к программно-аппаратной площадке, на которой возведены все необходимые ему модели для имитации и исследования того или иного технологического процесса. Немаловажно, что эти исследования для удаленного пользователя осуществляются в режиме реального времени.

Модуль оперативного обмена данными представляет собой набор приложений, по средствам которых конечные пользователи могут:

- обсуждать те или иные тематические вопросы предметной области (форум);
- осуществлять передачу коротких сообщений другим пользователям, которые в этот момент находятся на портале (on-line);
- оставлять сообщения для отсутствующих (off-line) пользователей портала;
- производить обмен данными (отчеты, программы и т.д.).

Используя корпоративный портал, преподаватель, располагая значительной электронной образовательной средой, не тратит времени на воспроизведение необходимой информации и использует сэкономленное время для более глубокого объяснения материала [8]. Совместное же использование единого гиперпространства между преподавателем и обучаемым создает творческое информационное поле для получения углубленных практических навыков.

## 8.5. ИНТЕГРАЦИЯ РАСПРЕДЕЛЕННЫХ WEB-СЕРВЕРОВ

Конечному пользователю, как упоминалось выше, для проведения комфортной работы необходим канал связи с большой пропускной способностью. К сожалению, на данный момент это может быть реализовано только в отдельно взятых регионах сети (не учитывая коммерческие линии связи) [2].

Эту задачу можно решить путем установления в требуемых частях сети дополнительных Web-серверов, на которых локально хранится информация с центрального сервера, причем при желании ее можно обновлять непосредственно с последнего. Конечный пользователь портала имеет возможность выбирать удаленный сервер для работы (свой регион сети) и выполнять все необходимые операции так же, как если бы он соединялся с центральным сервером (рис. 8.8).

Клиенты 1 и 2 имеют равные возможности, хотя могут располагаться друг от друга на большом расстоянии. Это обеспечивается с помощью вызова удаленных процедур на расширяемом языке разметки XML-RPC (eXtensible Mark-up Language Remote Procedure Call), который представляет собой сравнительно новую технологию вызова методов на распределенных компьютерах и извлечения информации из этих методов. С помощью языка XML передаются структурированные сообщения, инкапсулирующие вызовы функций, которые выполняются в

удаленных системах [11]; таким образом можно бесшовно интегрировать локальные и удаленные системы.

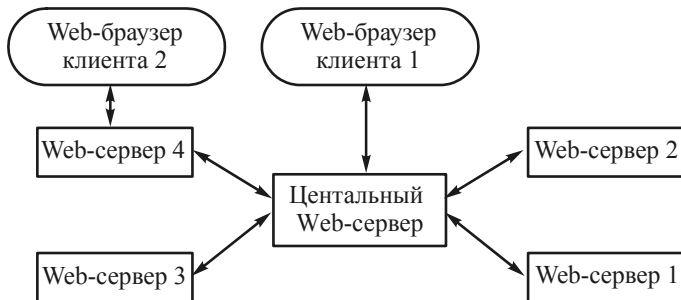


Рис. 8.8. Структура обмена данными между серверами и клиентами

Проиллюстрируем, как это достоинство можно использовать, например, в обучении.

На занятии в классе, оборудованном техническими средствами, – персональными компьютерами (ПК), объединенными в вычислительную сеть с выходом в Internet, большим монитором (БМ), подключенным к главному компьютеру (ГК) класса, – преподаватель с главного компьютера подключается к базе знаний одного из региональных Web-серверов (СР) портала и загружает необходимую ему информацию, причем вся информация, представленная в мультимедийной форме, отображается на большом дисплее (плазменной панели) и дублируется на всех стандартных мониторах персональных компьютеров, расположенных в аудитории. У обучаемых появляется возможность воспринимать информацию в удобной форме (цвет, видео, звук) и воспроизвести ее самостоятельно в любой момент времени из любой части глобальной информационной сети.

Приведем иерархию такого процесса на рис. 8.9. Одновременно с портала можно получать информацию как по одной дисциплине, так и по разным. Например, в момент времени  $t_1$  на ГК<sub>1</sub> и ГК<sub>3</sub> воспроизводится информация по условной дисциплине 1, причем компьютеры могут быть расположены на большом расстоянии друг от друга (разные вузы), а на ГК<sub>2</sub> воспроизводится информация по условной дисциплине 2 и т.д. Достигается это тем, что вся информация, расположенная на сер-

вере, представлена в XML-форме и имеет однородную структуру, поэтому без труда может перестраиваться в нужную форму.

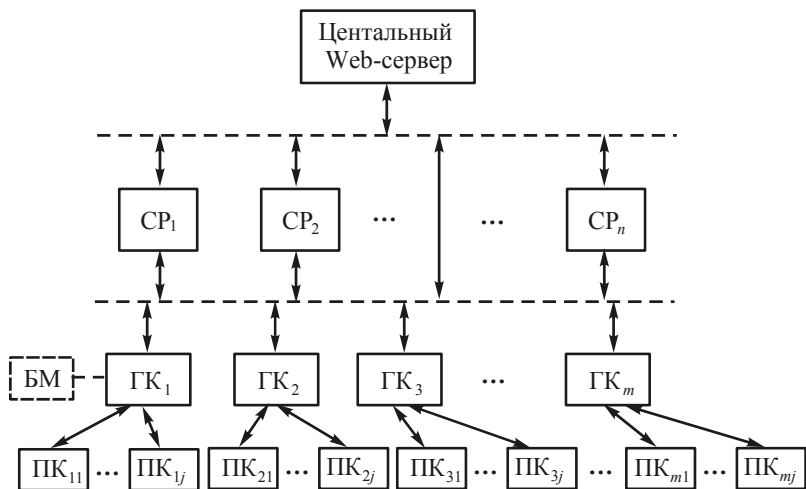


Рис. 8.9. Пример иерархии процесса, основанного на использовании корпоративного портала

Таким образом, использование перестраиваемых структур дает широкие возможности при передаче, обработке информации, задании алгоритмов функционирования того или иного устройства и автоматизации технологических процессов.

## 8.6. ВЫВОДЫ

Рассмотренные принципы применяются при построении корпоративных порталов различного назначения. Основой для функционирования портала может быть следующее программное обеспечение:

- Web-сервер: Apache ver.1.3.20;
- язык сценариев: PHP ver. 4.2.3;
- СУБД: MySQL ver. 3.23.51.

Применение языка разметки XML дает широкие функциональные возможности при разработке основных приложений и наделяет портал базовыми свойствами однородности. Этот аспект приводит к тому, что взаимодействие между приложениями сервера и клиента существенно

упрощается, поскольку программа может сама определять структуру документа XML. Следовательно, можно написать приложения, создающие документы с новой структурой, и эти документы могут быть интерпретированы без необходимости создания специального программного обеспечения.

В этом случае все элементы обработки данных нашей сети серверов, клиентов и приложений будут использовать один и тот же механизм обмена данными. Такой механизм является расширяемым и позволяет определять структуру документа в процессе работы приложения, т.е. достигается перестраиваемость. Кроме того, он поддерживается практически на каждой платформе, прост в использовании и способен обрабатывать данные разметки из различных источников. Для выполнения запросов клиентов разработчики приложений смогут использовать данные из других серверов и нетрадиционных источников.

Таким образом, Web-приложения смогут перейти от архитектуры клиент/сервер на истинно многоуровневую модель.

## ЛИТЕРАТУРА

1. *Автоматизация* производства и промышленная электроника / Под ред. А.И. Берг, В.А. Трапезникова и др.: В 4 т. – М.: Советская энциклопедия, 1965.
2. *Грейвс М.* Проектирование баз данных на основе XML. – М.: Вильямс, 2002. – 640 с.
3. *Лапко А.В., Ченцов С.В.* Непараметрические системы обработки информации. – М.: Наука, 2000. – 350 с.
4. *Либерти Д., Крейли М.* Создание документов XML для Web: Учебное пособие. – М.: Вильямс, 2000. – 256 с.
5. *Мартин Д., Бирбек М., Кэй М. и др.* XML для профессионалов. – М.: ЛОРИ, 2001. – 866 с.
6. *Цыпкин Я.З.* Адаптация и обучение в автоматических системах. – М.: Наука, 1968. – 400 с.
7. *Шальто А.А.* Логическое управление. Методы аппаратной и программной реализации алгоритмов. – СПб.: Наука, 2000. – 780 с.
8. *Шидловский С.В.* Принципы перестраиваемых структур в Internet-технологии // Научная сессия ТУСУР-2003: Материалы регион. науч.-техн. конф. – Томск: ТУСУР, 2003. – Ч. 3. – С. 101–103.
9. *Шидловский С.В.* Теория автоматического управления: Учебное пособие. – Томск: Изд-во НТЛ, 2003. – 40 с.
10. *Шидловский С.В., Светлаков А.А.* Исследование функциональных возможностей многофункционального логического модуля, реализующего опера-

ции удаления аргументов из булевых функций // Вестник Сибирского отделения АН ВШ. – 2002. – № 1(8). – С. 72–78.

11. Эдду С.Э. XML: Справочник. – СПб.: Питер, 2000. – 480 с.

12. *Shidlovskiy S.V.* Multifunctional Automaton for Computation of Disordered Boolean Functions // The VIII International Scientific and Practical Conference of Students, Post-graduates and Young Scientist «Modern Techniques and Technology» (МТТ'2002). – Tomsk: Tomsk Polytechnic University, 2002.

---

---

## ЗАКЛЮЧЕНИЕ

---

---

Работа посвящена актуальному, весьма перспективному и быстро развивающемуся на основе использования автоматного принципа обработки информации направлению в области построения цифровых управляющих устройств и разработки их внутренней структуры для систем автоматического управления технологическими процессами.

Основной целью работы являлось повышение качества функционирования систем автоматического регулирования путем построения и исследования математических моделей адаптивных автоматизированных структурно перестраиваемых систем управления обработки информации и проектирования.

Отметим наиболее важные результаты, полученные в работе.

1. Установлено соответствие между булевой моделью логики перестраиваемых структур, изотропными средами и устройствами управления, заключающееся в том, что их взаимосвязь приводит к построению систем автоматического регулирования, способных функционировать при неполной информации об объекте и автоматически выбирать наилучший закон управления из заданного класса.

2. Разработана булева модель логики перестраиваемых структур для определенных классов булевых функций.

3. Разработаны изотропные среды, обеспечивающие реализацию систем булевых формул из классов неповторных упорядоченных и неупорядоченных булевых функций, а также из класса повторных упорядоченных произвольных нормальных булевых формул из букв и систем булевых формул как с пропусками аргументов, так и без них.

4. Разработаны функционально необходимые модели регуляторов, в том числе с перестраиваемой структурой, с широкими возможностями и большей приспособленностью для практической реализации.

5. Обеспечено повышение качества регулирования и расширение функциональных возможностей систем управления.

6. Разработано специализированное программное обеспечение структурно перестраиваемых автоматизированных систем управления технологическим процессом, в том числе система автоматизированного проектирования приборов и оборудования.

**Рекомендации.** Дальнейшие исследования могут развиваться в следующих направлениях:

1) исследование предельных возможностей разработанных ячеек в матричных и иерархических структурах;

2) разработка новых многофункциональных логических модулей и изотропных сред, охватывающих полный класс булевых функций;

3) создание высокоинтеллектуальных управляющих устройств с перестраиваемой структурой;

4) создание так называемых «кинематических ячеистых автоматов» – наноустройств, состоящих из идентичных блоков. Такие автоматы могут наращивать функциональность путем подключения дополнительных блоков, т.е., по сути, собирать себя сами.

В итоге системы получают способность менять свою структуру и подключать к решению задачи необходимые узлы в зависимости от возникающей ситуации. Возможно, именно это и есть те пути, по которым техника будет продвигаться к созданию сплошных эволюционирующих и адаптирующихся вычислительных сред, управляющих технологическими процессами.



# Приложение 1

## ТАБЛИЦЫ ДЛЯ ФОРМУЛ S-СТРУКТУРЫ И Ф-ИМПЛИКАНТ

Таблица П.1.1

Ячейка 1		Ячейка 2		$f$
$\alpha_1$	$S_1$	$\alpha_2$	$S_2$	
0	0	0	0	0
0	0	0	1	0
0	0	1	0	$A_1A_2$
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	$A_2$
0	1	1	1	0
1	0	0	0	$A_1$
1	0	0	1	$A_1$
1	0	1	0	$A_1 \vee A_2$
1	0	1	1	$A_1$
1	1	0	0	0
1	1	0	1	0
1	1	1	0	$A_2$
1	1	1	1	0

Таблица П.1.2

$S_1$	$S_2$	$S_3$	$\alpha_1$	$\alpha_2$	$\alpha_3$	$f$	$S_1$	$S_2$	$S_3$	$\alpha_1$	$\alpha_2$	$\alpha_3$	$f$
1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	$A_1A_2A_3$	1	0	0	0	0	1	$A_2A_3$
0	0	0	0	1	0	$A_1A_2$	1	0	0	0	1	0	$A_2$
0	0	0	0	1	1	$A_1A_2 \vee A_3$	1	0	0	0	1	1	$A_2 \vee A_3$
0	0	0	1	0	0	$A_1$	1	0	0	1	0	0	0

Продолжение табл. П.1.2

1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	1	0	1	$A_1 \vee A_2 A_3$	1	0	0	1	0	1	$A_2 A_3$
0	0	0	1	1	0	$A_1 \vee A_2$	1	0	0	1	1	0	$A_2$
0	0	0	1	1	1	$A_1 \vee A_2 \vee A_3$	1	0	0	1	1	1	$A_2 \vee A_3$
0	0	1	0	0	0	0	1	0	1	0	0	0	0
0	0	1	0	0	1	0	1	0	1	0	0	1	0
0	0	1	0	1	0	$A_1 A_2$	1	0	1	0	1	0	$A_2$
0	0	1	0	1	1	$A_1 A_2$	1	0	1	0	1	1	$A_2$
0	0	1	1	0	0	$A_1$	1	0	1	1	0	0	0
0	0	1	1	0	1	$A_1$	1	0	1	1	0	1	0
0	0	1	1	1	0	$A_1 \vee A_2$	1	0	1	1	1	0	$A_2$
0	0	1	1	1	1	$A_1 \vee A_2$	1	0	1	1	1	1	$A_2$
0	1	0	0	0	0	0	1	1	0	0	0	0	0
0	1	0	0	0	1	$A_1 A_3$	1	1	0	0	0	1	$A_3$
0	1	0	0	1	0	0	1	1	0	0	1	0	0
0	1	0	0	1	1	$A_1 \vee A_3$	1	1	0	0	1	1	$A_3$
0	1	0	1	0	0	$A_1$	1	1	0	1	0	0	0
0	1	0	1	0	1	$A_1 \vee A_3$	1	1	0	1	0	1	$A_3$
0	1	0	1	1	0	$A_1$	1	1	0	1	1	0	0
0	1	0	1	1	1	$A_1 \vee A_3$	1	1	0	1	1	1	$A_3$
0	1	1	0	0	0	0	1	1	1	0	0	0	0
0	1	1	0	0	1	0	1	1	1	0	0	1	0
0	1	1	0	1	0	0	1	1	1	0	1	0	0
0	1	1	0	1	1	0	1	1	1	0	1	1	0
0	1	1	1	0	0	$A_1$	1	1	1	1	0	0	0
0	1	1	1	0	1	$A_1$	1	1	1	1	0	1	0
0	1	1	1	1	0	$A_1$	1	1	1	1	1	0	0
0	1	1	1	1	1	$A_1$	1	1	1	1	1	1	0

Таблица П.1.3

Ф-ИМПЛИ- КАНТЫ	$A_1 A_2 A_3$	$A_1 A_4 A_6$	$A_4 A_7 A_8 A_9$	$A_5 A_7 A_8$	$A_9 A_{10} A_{12} A_{14}$	$A_9 A_{13}$	$A_{10} A_{11} A_{12}$	$A_{13} A_{14}$
1	2	3	4	5	6	7	8	9
$\varphi_1$	1	0	1	0	0	0	1	1
$\varphi_2$	1	0	0	1	1	0	0	0
$\varphi_3$	1	0	0	1	0	1	0	0

Продолжение табл. П.1.3

1	2	3	4	5	6	7	8	9
$\varphi_4$	1	0	0	1	0	0	1	1
$\varphi_5$	0	1	0	0	1	0	0	0
$\varphi_6$	0	1	0	0	0	1	0	0
$\varphi_7$	0	1	0	0	0	0	1	1

Таблица П.1.4

Ф-импли- каны	$A_1A_4A_6$	$A_5A_7A_8$	$A_9A_{10}A_{12}A_{14}$	$A_9A_{13}$
$\varphi_2$	0	1	1	0
$\varphi_3$	0	1	0	1
$\varphi_4$	0	1	0	0
$\varphi_5$	1	0	1	0
$\varphi_6$	1	0	0	1
$\varphi_7$	1	0	0	0

## Приложение 2

### СТРУКТУРНАЯ СХЕМА АЛГОРИТМА, ТЕКСТ ПРОГРАММЫ И ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ ДЛЯ РАСЧЕТА ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ПИ-РЕГУЛЯТОРА

Алгоритм расчета (рис. П.2.1) реализован на языке MatLab, так как данная среда программирования является наиболее удобной. Код программы представлен в виде m-файла. Программа выводит на дисплей график зависимости оптимальных параметров настройки (ОПН) регулятора согласно заданным требованиям, предъявляемым к системе, на котором необходимо выбрать точку, соответствующую желаемому интегральному критерию качества регулирования (максимум – первый, 95% максимума вправо – второй). Затем рассчитывается замкнутая система автоматического регулирования с выбранными параметрами настройки регулятора. В завершении работы выводится график переходного процесса замкнутой АСР.

#### Текст программы

```
%=====
%Задание параметров объекта регулирования
%=====
K=6; %Коэффициент усиления
T=97.7; %Постоянная времени
tau=23.4;%Величина запаздывания
%=====
%Заданные требования к системе регулирования
%=====
m=0.305; %Степень колебательности
%=====
%Расчет ОПН ПИ-регулятора
%=====
w=[0:0.001:0.03];%Частота для расчета
[TDddd ST]=size(w);%Определения количества ...
...рассчитываемых точек
```

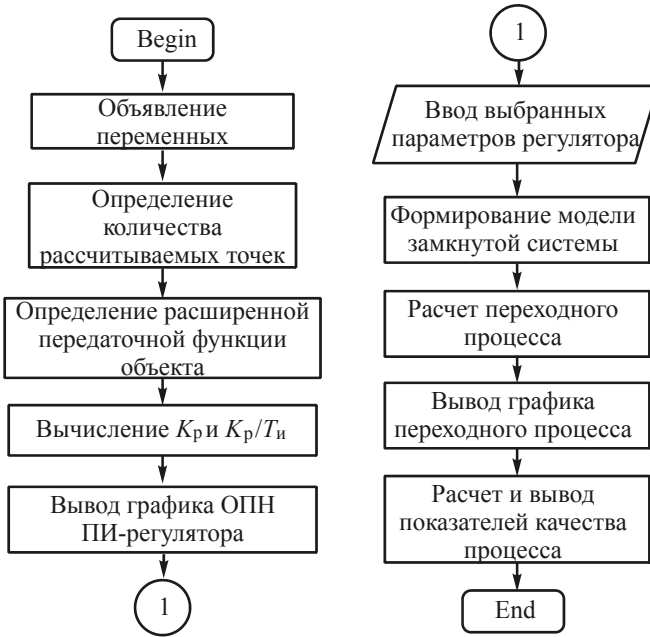


Рис. П.2.1. Структурная схема алгоритма

```

WmXobPI=(K*exp(-tau*(1i-m).*w))./(T*...
... ((1i-m).*w)+1);% Расширенная передаточная...
... функция объекта
C1=0;
C0=0;
C1=(-m.*imag(WmXobPI)-real(WmXobPI))./ ...
... (abs(WmXobPI).*abs(WmXobPI))); ...
...%Коэффициент усиления регулятора
%=====
%Цикл для вычисления Kr/Ti
%=====
for jj=1:ST
    if (imag(WmXobPI(jj))<=0)
        C0(jj)=((-m*m+1).*imag(WmXobPI(jj)).* ...
...w(jj))./(abs(WmXobPI(jj)).*abs(WmXobPI(jj))));
    else
    
```

```

        C0(jj)=((- (m*m+1) .* ...
...imag(WmXobPI(jj)).*w(jj))./ ...
... (abs(WmXobPI(jj)).*abs(WmXobPI(jj))));
    end;
end;

plot(C1,C0,'g');grid on; %Вывод результата...
... расчета ОПН ПИ-рег
%-----
%Интерактивный ввод результатов вычисления
%с графика
%-----
C0=input('Kp/Tu= ?');
C1=input('Kp= ?');
Kрпи=C1;
Тipi=Kрпи/C0; %Расчет постоянной...
... времени изодрома

%-----
%Расчет замкнутой АСР с найденными близкими
% к оптимальным параметрами ПИ-регулятора
%-----
disp('----- Start System Computation -----');
%передаточная функция ПИ-регулятора в ЛТИ форме
sys1=tf(Kрпи*[Тipi 1],[Тipi 0], ...
...'inputname','1in','outputname','1out');
%передаточная функция объекта в ЛТИ форме
sys2=tf(K,[Т 1],'td',tau,'inputname', ...
...'2in','outputname','2out');
sys=append(sys1,sys2);%агрегатированная ...
... модель без перекрестных связей
Q=[2 -1 0;1 2 0];%матрица связей
inputs=[2];%матрица внешних входов
outputs=[2];%матрица внешних выходов
sysc=connect(sys,Q,inputs,outputs);
%модель в пространстве состояний с...
...учетом перекрестных связей между блоками
[y x]=step(sysc); %расчет переходной...

```

```
... характеристики замкнутой системы
plot(x,y); %вывод переходной...
... характеристики замкнутой системы

%-----
%Определение показателей качества процесса
%регулирования
%-----

[si si2]=size(y);
[A1 c1]=max(y);
[A2 c2]=min(y(c1:1:si));
[A3 c3]=max(y(c1+c2:1:si));
sigma=(abs(A2-y(si))/(A1-y(si)))*100
%перерегулирование
psi=((A1-y(si))-(A3-y(si)))/(A1-y(si))
%степень затухания
FIdin=(A1-y(si))
%максимальная динамическая ошибка
YY=y-y(si);
delta=0.05*y(si);
zz=abs(YY)-delta;
[zn zx]=min(abs(zz));
Treg=x(zx)
%Время регулирования
```

## Приложение 3

### СТРУКТУРНАЯ СХЕМА АЛГОРИТМА, ТЕКСТ ПРОГРАММЫ И ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ ИНТЕГРАЛЬНОГО РЕГУЛЯТОРА С ПЕРЕСТРАИВАЕМОЙ СТРУКТУРОЙ

Алгоритм функционирования интегрального регулятора с перестраиваемой структурой (рис. П.3.1) реализован на языке MatLab, так как данная среда программирования является наиболее удобной и более прикладной для решаемой задачи. Код программы представлен в виде двух m-файлов, реализующих соответственно две функции.

Первая представляет из себя так называемую *S*-функцию среды MatLab, которые предназначены для построения пользовательских блоков системы Simulink и позволяют описывать части системы или всю систему на алгоритмическом языке. Реализованная функция непосредственно выполняет алгоритм представленные в разделе 5.6, формируя управляющее воздействие из множества значений  $\{1, 0, -1\}$ .

Вторая функция реализует МЛМ *L*-структуры и вызывается первой для вычисления ряда систем булевых формул.

*S*-функция носит название `isps_mlm` и в качестве аргументов принимает массив значений состоящий из коэффициента наклона прямой в фазовом пространстве  $c$ , постоянных величин служащих для формирования внутренних переменных  $V1$ ,  $V2$  и постоянных времени  $T0$ ,  $T1$  двух линейных структур участвующих при решении дифференциального уравнения.

#### Текст программы

```
function [sys,x0,str,ts] = isps_mlm(t, x, u, ...
... flag, c, V1, V2, T0, T1)
switch flag,
    % Инициализация%
    case 0,
        [sys,x0,str,ts]=mdlInitializeSizes;
    % Вычисление вектора выхода %
    case 3,
        sys=mdlOutputs(t,x,u,c,V1,V2,T0,T1);
    % Пустые флаги%
    case {1, 2, 4, 9}
```



```

sys=[];
% Ошибочный вход %
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
    
```

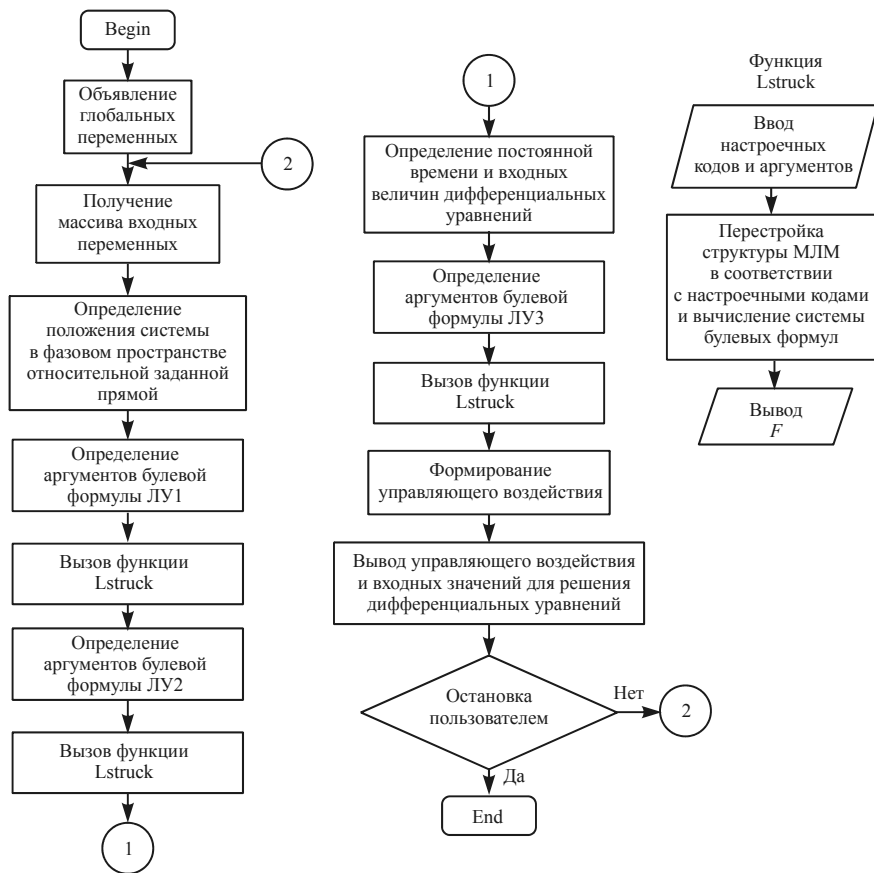


Рис. ПЗ.1. Структурная схема алгоритма

```

% Функция инициализации
function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
    
```

```

sizes.NumContStates = 0; %число непрерывных ...
... состояний
sizes.NumDiscStates = 0; %число дискретных ...
... состояний
sizes.NumOutputs     = 4; %число выходов
sizes.NumInputs      = 4; %число входов
sizes.DirFeedthrough = 1; %число прямых проходов
sizes.NumSampleTimes = 1;

sys = simsizes(sizes); %определение размера ...
... векторов переменных модели
x0 = []; %установка начальных значений для ...
... всех переменных состояний
str = []; %пустая матрица
ts = [-1 0]; %Инициализация матрицы периодов ...
... квантования
%Объявление переменных
global s, b, ResetTemp, sigma1, sigma2, ...
... DeltaT, ro, U1, U2, U3, U4, U5, gamma, ...
... vv, tetta, fi, nyu, ksi, SIGMA, ALPHA, ...
... BETTA, lyamda, Tsig, eps, ss, dzita;
%Начальные условия
eps = 0.00528;
Tsig = 5;
[s ss b ResetTemp sigma1 sigma2 DeltaT ro U1 ...
... U2 U3 U4 U5 gamma vv tetta fi nyu ksi ...
... SIGMA ALPHA BETTA lyamda dzita] = 0
% Окончание функции инициализации
%-----
% Функция вывода
%-----
function sys=mdlOutputs(t,x,u,c,V1,V2,T0,T1)
%с - коэффициент прямой в фазовой плоскости
%V1, V2 - константы
%T0,T1 - постоянные времени
tetta=u(3); %решение дифференциального ...
... уравнения 1

```

```
fi=u(4); %решение дифференциального уравнения 2
%Уравнение прямой в фазовой плоскости
ss=u(2)+c*u(1);
s=sign(ss)*u(1);
ss=s*sign(u(1));
%Вычисление вспомогательных переменных
U1=~(sign(abs(sign(s)-1)));
nyu=U1*u(1);
U2=~SIGMA;
ksi=U2*nyu;
if tetta>eps
    dzita=1;
else
    dzita=0;
end
%Решение системы булевых формул с помощью ...
... МЛМ L-структуры
ResetTemp=Lstruck(~abs(sign(s)),dzita, ...
... 0,0,1,0,0);
U4=ResetTemp(1);
ALPHA=U4*fi;
lyamda=ALPHA+ВЕТТА;
if lyamda>eps
    SIGMA=1;
else
    SIGMA=0;
end
%Решение системы булевых формул с помощью МЛМ...
... L-структуры
ResetTemp=Lstruck(abs(sign(nyu)), ...
... ~SIGMA,0,0,1,0,0);
U3=ResetTemp(1);
%Входная величина для решения ...
... дифференциального уравнения 1
gamma=U3*V1;
U5=SIGMA;
ВЕТТА=U5*fi;
%Постоянная времени в дифференциальном ...
```

```

... уравнении 2
Tsig=T0*(~SIGMA)+T1*SIGMA;
%Входная величина для решения ...
... дифференциального уравнения 2
vv=U2*V2;
%Аргументы логического устройства
ResetTemp=0;
ResetTemp(1)=sign(sign(ksi+eps/2)+1);
ResetTemp(2)=sign(sign((eps/2)-ksi)+1);
%Решение системы булевых формул с помощью МЛМ...
... L-структуры
ResetTemp=Lstruck(~ResetTemp(2),ResetTemp(2), ...
... ~ResetTemp(1),ResetTemp(1),0,0,1);
%Формирование выходной величины принимающее ...
... значение {1,0,-1}
ro=(ResetTemp(1)+(ResetTemp(3)*(-1)));
%Выход регулятора
sys(1)=ro;
sys(2)=gamma;
sys(3)=vv;
sys(4)=Tsig;
% Окончание функции вывода

%Модель L-структуры
%% [F1,F2,F3]=Lstruck(x1,x2,x3,x4,z1,z2,z3)
function rez=Lstruck(y1,y2,y3,x,z1,z2,z3)
% Система булевых формул МЛМ L-структуры
F1=y1&(~z1&z2|z3)|(y1|z1|z2)&x&(~z3);
F2=y1&z1&z2&(~z3)|y2&z1&(~z2)&z3|(y1|(~z1)|z2)&y
2&(~z3)|(y2|z1|z2)...
...&x&z3;
F3=y3&(~z3)|(y2|(~z1)|(~z2))&y3&z3|y2&(~z1)&z2&z
3;
%Вывод результата
rez=[F1; F2; F3]; return;

```

## СПИСОК ОСНОВНЫХ СОКРАЩЕНИЙ

- АИПС – автоматизированная информационно-поисковая система  
АСР – автоматическая система регулирования  
БФ – булева формула  
БФу – булева функция  
ДКВ – динамический критерий выдачи  
ДНФ – дизъюнктивная нормальная форма  
ИП-автомат – информационно-поисковый автомат  
ИПС – информационно-поисковая система  
ИПЯ – информационно-поисковый язык  
КНФ – конъюнктивная нормальная форма  
М-автомат – многофункциональный автомат  
МЛМ – многофункциональный логический модуль  
НЛР – нечеткий логический регулятор  
ОС – однородная среда  
ОУ – объект управления  
ПИ-регулятор – пропорционально-интегральный регулятор  
ПИД-регулятор – пропорционально-интегро-дифференциальный регулятор  
ПОД – поисковый образ документа  
ПОЗ – поисковый образ запроса  
РАФЧХ – расширенная амплитудно-фазочастотная характеристика  
СДНФ – совершенная дизъюнктивная нормальная форма  
СПС – система с перестраиваемой структурой  
ТИ – таблица истинности  
УЛМ – универсальный логический модуль  
УУ – устройство управления  
XML (eXtensible Markup Language) – расширяемый язык разметки

**Шидловский  
Станислав**

**Викторович**

Родился в 1981 г. в Томске. В 1997 г. поступил, а в 2002 г. окончил Томский политехнический университет по специальности «Автоматизация технологических процессов и производств».

В 2004 г. досрочно закончил аспирантуру с защитой кандидатской диссертации по специальности «Автоматизация и управление технологическими процессами и производствами».

С 2005 г. – доцент кафедры информационно-измерительной техники Томского государственного университета систем управления и радиоэлектроники. Обеспечивает учебный процесс кафедры по двум дисциплинам: «Теория автоматического управления» и «Автоматизация технологических процессов и производств».

Автор более 40 научных трудов, в том числе ряда патентов РФ. Лауреат Всероссийского конкурса «Ползуновские гранты» (2005 г.), победитель федеральной целевой научнотехнической программы Роснауки (2006 г.).



**Shidlovskiy  
Stanislav**

**Viktorovich**

Was born in 1981 in Tomsk. In 1997 passed the entering exams successfully and in 2002 graduated from Tomsk polytechnical university on speciality «Automation of technological process and productions».

In 2004 he finished a postgraduate studying ahead of schedule with the defense of the candidate's thesis on speciality «Automation and control of technological process and productions».

Since 2005 – the associate professor of the faculty of information-measuring technical devices of Tomsk state university of control systems and radio electronics. He provides the educational process at the chair on two disciplines: «the Theory of automated control» and «Automation of technological and production process».

The author of more than 40 scientific works, including some patents of the Russian Federation. The laureate of the All-Russian competition «Polzunov's grants» (2005), the winner of the federal purpose-oriented scientific and technical program of Russian science (2006).

***Готовится к изданию  
во втором квартале 2007 г.***

**Шидловский С.В.** Автоматическое управление. Перестраиваемые структуры в системах с распределенными параметрами. – Томск: Томский государственный университет, 2007.

Монография посвящена комплексному исследованию перестраиваемых структур в системах управления с распределенными параметрами и является продолжением монографии автора «Автоматическое управление. Перестраиваемые структуры».

В книге освещены вопросы логического управления, изложены методы и алгоритмы автоматизированного проектирования, моделирования и обработки информации систем с распределенными параметрами.

Наряду с классическими методами анализа и синтеза систем управления с распределенными параметрами предложен автоматико-структурный метод, который реализуется как совокупность аппаратно-программных фрагментов вычисления процедур, составляющих алгоритмы управления из заданных классов. Метод позволяет строить высокоэффективные модульно наращиваемые структуры с массовым параллелизмом, обеспечивающие требуемую производительность, технически и экономически оправданные, для конкретного ряда приложений.

Книга предназначена для специалистов в области управления системами с распределенными параметрами.

*Научное издание*

*Станислав Викторович Шидловский*

**АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ.  
ПЕРЕСТРАИВАЕМЫЕ СТРУКТУРЫ**

Редактор *К.Г. Шилько*  
Корректор *Н.А. Афанасьева*  
Оригинал-макет *С.В. Шидловского*  
Дизайн обложки *В.Г. Караваева*

Подписано в печать 17.08.2006 г.  
Формат 60×84<sup>1</sup>/<sub>16</sub>. Бумага мелованная.  
Печать офсетная. Гарнитура «Таймс».  
Усл. печ. л. 16,74. Уч.-изд. л. 18,75.  
Тираж 500 экз. Заказ

Томский государственный университет  
634050, г. Томск, пр. Ленина, 36  
Участок оперативной ризографии и офсетной печати  
Редакционно-издательского отдела ТГУ

ISBN 5-94621-186-2



9 785946 211864