

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Кафедра систем управления

**С. В. Снисаренко, Н. А. Стасевич**

***СОВРЕМЕННЫЕ МЕТОДЫ  
СИНТЕЗА СИСТЕМ УПРАВЛЕНИЯ***

Лабораторный практикум  
для студентов специальности I-53 01 07  
«Информационные технологии и управление в технических системах»  
всех форм обучения

Минск 2008

УДК 681.511.4 (075.8)  
ББК 32.965.5. я 73  
С 56

**Р е ц е н з е н т**  
канд. техн. наук,  
доцент кафедры ИТАС БГУИР А. М. Севернев

**Снисаренко, С. В.**

С 56      Современные методы синтеза систем управления : лаб. практикум для студ. спец. I-53 01 07 «Информационные технологии и управление в технических системах» всех форм обуч. / С. В. Снисаренко, Н. А. Стасевич. – Минск : БГУИР, 2008. – 59 с. : ил.

ISBN 978-985-488-253-6

Настоящий лабораторный практикум включает в себя 6 лабораторных работ. Все работы выполняются на ЭВМ с использованием пакета Matlab. Каждая работа содержит краткие теоретические сведения, пример расчета, варианты заданий и список вопросов для самопроверки.

Практикум предназначен для студентов 5-го курса специальности I-53 01 07 «Информационные технологии и управление в технических системах» при выполнении ими лабораторных работ по курсу «Современные методы синтеза систем управления».

Приведенные в практикуме работы могут служить основой для дальнейших исследований в области синтеза нелинейных систем управления.

**УДК 681.511.4(075.8)**  
**ББК 32.965.5. я 73**

**ISBN 978-985-488-253-6**

© Снисаренко С. В., Стасевич Н. А., 2008  
© УО «Белорусский государственный университет информатики и радиоэлектроники», 2008

## СОДЕРЖАНИЕ

<b>Лабораторная работа №1.</b> Исследование псевдолинейного корректирующего устройства с запоминанием экстремума сигнала ошибки....	5
<b>Лабораторная работа №2.</b> Синтез оптимальных регуляторов для нелинейных объектов управления на основе использования притягивающих инвариантных многообразий .....	12
<b>Лабораторная работа №3.</b> Синтез дифференциальных анализаторов с использованием метода структурного синтеза Л. М. Бойчука .....	20
<b>Лабораторная работа №4.</b> Синтез регуляторов с переменной структурой для управления нелинейным объектом.....	27
<b>Лабораторная работа №5.</b> Синтез управляющих систем на основе нечеткой логики .....	34
<b>Лабораторная работа №6.</b> Аппроксимация функций с помощью нейронных сетей .....	45
Литература.....	58

## Список сокращений

ПКУЗЭ – псевдолинейное корректирующее устройство с запоминанием экстремума

САУ – системы автоматического управления

ЛКУ – линейные корректирующие устройства

АЧХ – амплитудно-частотная характеристика

ФЧХ – фазочастотная характеристика

ДНКУ – двухканальные нелинейные корректирующие устройства

ПД – пиковый детектор

ОУ – объект управления

НУ – начальные условия

СПС – системы переменной структуры

СР – скользящий режим

ДУ – дифференциальное уравнение

ПИ-регулятор – пропорционально-интегральный регулятор

НС – нейронные сети

Библиотека БГУИР

## Лабораторная работа №1

### ИССЛЕДОВАНИЕ ПСЕВДОЛИНЕЙНОГО КОРРЕКТИРУЮЩЕГО УСТРОЙСТВА С ЗАПОМИНАНИЕМ ЭКСТРЕМУМА СИГНАЛА ОШИБКИ

**Цель** – изучить организацию разрывного управления в ПКУЗЭ (псевдолинейном корректирующем устройстве с запоминанием экстремума) и исследовать влияние настройки его параметров на качество работы САУ.

#### 1.1 Краткие теоретические сведения

Линейные минимально-фазовые корректирующие устройства (ЛКУ) часто не могут обеспечить синтезируемой САУ желаемые (оптимальные) качественные показатели, особенно в условиях неполной информации о состоянии объекта управления. Ограниченные возможности ЛКУ объясняются в первую очередь тем, что для них характерна жесткая однозначная связь между амплитудными (АЧХ) и фазовыми (ФЧХ) частотными характеристиками. Такая связь не позволяет, например, создать апериодическое звено с опережающей ФЧХ в некотором диапазоне частот.

В псевдолинейных корректирующих устройствах в отличие от ЛКУ используются разрывные (релейные) режимы работы, которые формируют управления в виде комбинаций определённых частей сигналов с положительными и отрицательными знаками. Псевдолинейные двухканальные нелинейные корректирующие устройства (ДНКУ) или фильтры Гизекинга позволяют синтезировать ФЧХ независимо от АЧХ, что открывает возможность получения устойчивых систем, имеющих заданную точность и хорошие фильтрующие свойства на высоких частотах.

Одним из недостатков ДНКУ является то, что в нём упреждающий сдвиг по фазе и связанный с этим сдвигом момент переключения с разгона на торможение зависят от частоты и параметров фильтра  $K_{\phi}(s)$ . Так как момент переключения не может быть выбран произвольно, то ДНКУ позволяют получить лишь ограниченные фазовые упреждения в сравнительно узком диапазоне частот. При неудачном выборе параметров фильтра в канале фазы введение ДНКУ может даже ухудшить качество переходных процессов корректируемой системы.

В связи с этим перспективной является идея регулирования момента переключения (реверса) не в функции частоты, а в зависимости от величины экстремума сигнала ошибки. Такая идея может быть реализована в виде ПКУЗЭ, в котором переключение с режима разгона на торможение осуществляется в момент времени, когда текущее значение ошибки  $e(t)$  становится равным заранее определённой доле экстремума  $e_m$ , т.е.  $e(t) = m e_m$ , где  $m \in [0; 1]$ . Так как величину  $m$  на отрезке  $[0; 1]$  можно назначать произвольно, то появляется сво-

бода выбора моментов реверса, обеспечивающих переход от разгона к торможению. Изменяя коэффициенты усиления  $k^+$ ,  $k^-$  соответственно для режимов разгона и торможения, можно дополнительно регулировать качество переходных процессов синтезируемой САУ.

Для системы, работающей в режиме однократного переключения (рисунок 1.1), задача синтеза ПКУЗЭ сводится к определению коэффициентов  $k^+ = tg g$ ,  $k^- = tg b$  и величины  $m$ , обеспечивающей переход на участок торможения. Ясно, что увеличение угла  $g$  приводит к ускорению переходного процесса и росту перерегулирования, а такое же изменение угла  $b$  (т.е. коэффициента  $k^-$ ) оказывает противоположный эффект. Алгоритм работы системы с ПКУЗЭ при одном переключении пояснён на фазовой плоскости (рисунок 1.2).

Изображающая точка  $N$  в режиме разгона движется на фазовой траектории по часовой стрелке. В момент прохождения оси абсцисс ошибка достигает максимального значения  $e_m$ , а затем начинает уменьшаться. Переключение происходит при условии  $e(t) = me_m$ , и система переходит в режим торможения, который длится до обнуления сигнала ошибки.

Функциональная схема системы с ПКУЗЭ приведена на рисунке 1.3. Для запоминания максимального значения ошибки  $e_m$  служит пиковый детектор (ПД). Выходной сигнал с ПД умножается на величину  $m \approx 0,65$ , а затем сравнивается с текущим значением модуля ошибки  $|e(t)|$ . Результат сравнения  $z(t) = |e(t)| - me_m$  подаётся на сигнум – реле, которое выдаёт  $k^+$  при  $|e(t)| > me_m$  или  $-k^-$  при  $|e(t)| < me_m$ , осуществляя тем самым переход от разгона к торможению. Блок умножения присваивает модулю сигнала ошибки  $|e(t)|$  соответствующий знак с коэффициентом  $k^+$  или  $k^-$ . При этом создаётся зависимость  $u = f(e)$ , имеющая вид петли с обходом по часовой стрелке (см. рисунок 1.1).

Изменение параметров САУ, влияние возмущений и другие факторы приводят к тому, что в режиме работы с одним переключением изображающая точка на фазовой траектории может пройти мимо начала координат. Это вызывает или появление перерегулирования, или установившейся ошибки, что в любом случае ухудшает качество работы системы. Поэтому в системах с разрывными законами управления приходится на практике применять многократное переключение с режимов разгона на торможение, как показано на фазовом портрете, изображённом на рисунке 1.4. После первого переключения в момент прохождения скорости изменения ошибки через нуль ( $\dot{e} = 0$ ) необходимо сбросить прежнее значение экстремума  $e_m$  и занести в ПД новую максимальную величину  $e'_m$ . Второе переключение на режим торможения произойдёт при выполнении условия  $|e(t)| = me'_m$ , а дальнейшие циклы переключения происходят аналогичным образом, до тех пор пока ошибка не обнулится.

Все переключения должны осуществляться в четвёртом квадранте фазовой плоскости, где  $e\dot{e} \leq 0$ , а сброс и запоминание – в моменты времени, когда

начинает выполняться неравенство  $e\dot{e} > 0$ . Выходной сигнал пикового детектора (см. рисунок 1.3)  $u_{ПД}$  формируется по закону

$$u_{ПД} = \begin{cases} \max(u_{ПД}, |e|), & \text{при } e\dot{e} \leq 0, \\ |e|, & \text{при } e\dot{e} > 0. \end{cases}$$

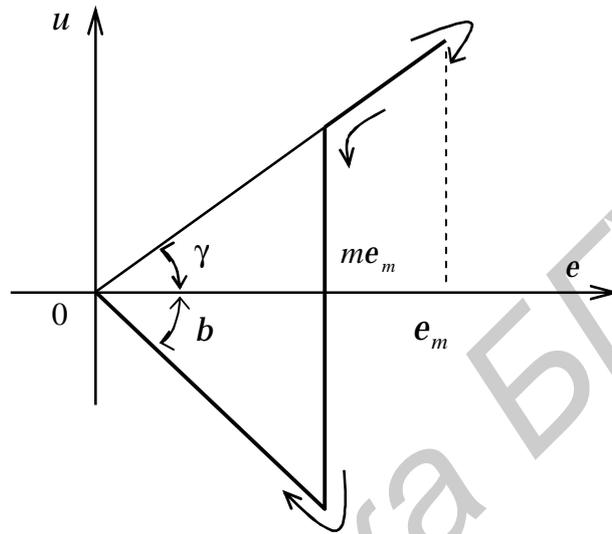


Рисунок 1.1 – График зависимости  $u = f(e)$

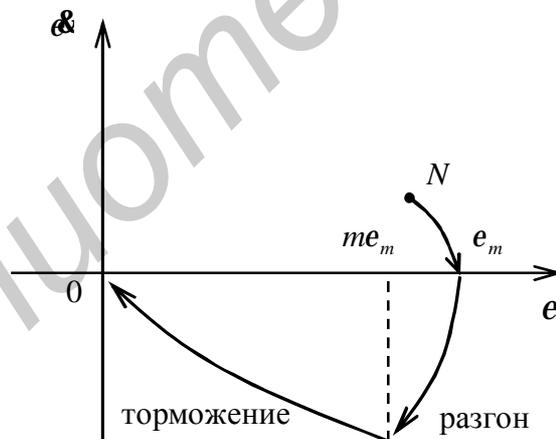


Рисунок 1.2 – Фазовый портрет работы системы в режиме однократных переключений

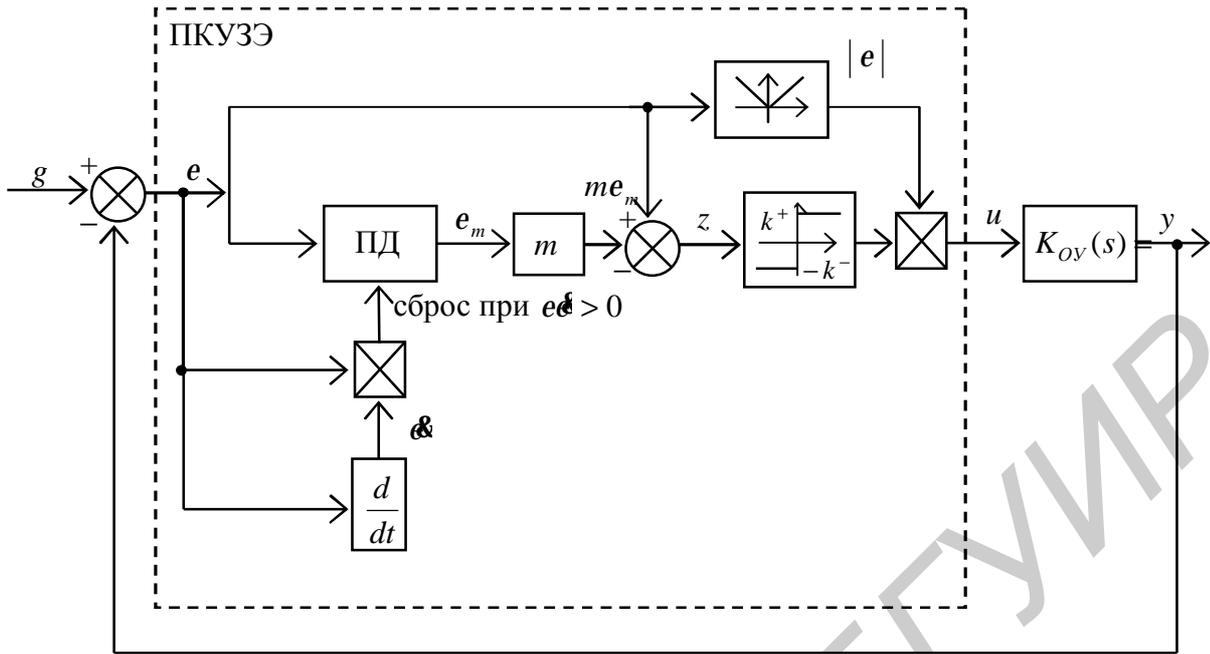


Рисунок 1.3 – Функциональная схема системы с ПКУЗЭ

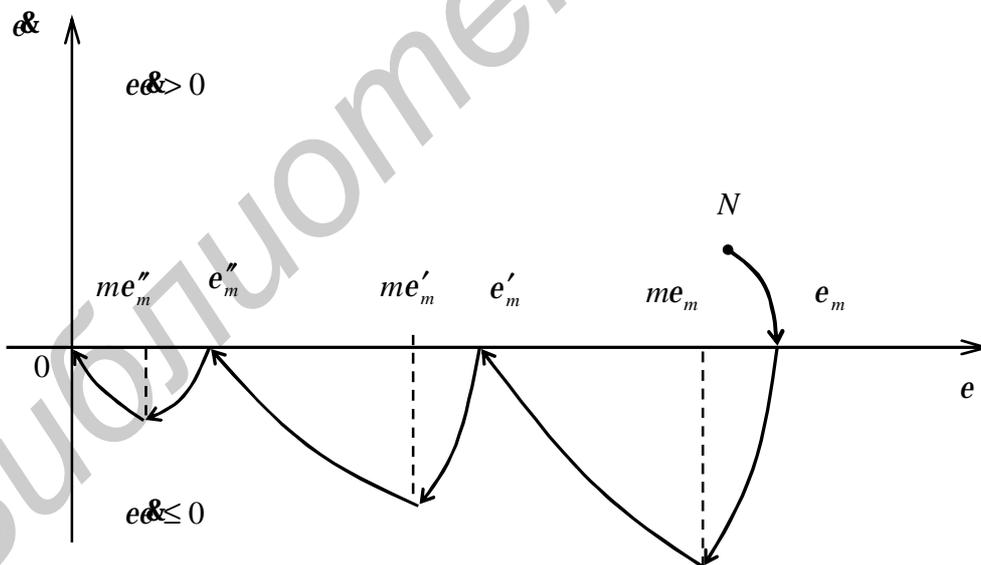


Рисунок 1.4 – Фазовый портрет работы системы в режиме многократных переключений

## 1.2 Порядок и цель моделирования

1 Вначале анализируется нескорректированная система, для чего на ОУ подаётся непосредственно сигнал ошибки  $e(t)$ , а не управление с выхода ПКУЗЭ. После регистрации переходного процесса и фазового портрета исследуется скорректированная система.

2 Целью моделирования является получение переходных процессов и фазовых портретов при различных значениях  $m \in [0,1 \text{ К } 0,9]$ ,  $k^+ \in [0,4 \text{ К } 1,2]$ ,  $k^- \in [0,5 \text{ К } 1,5]$ . Следует подобрать такие  $m, k^+, k^-$ , при которых перерегулирование в САУ не превышало  $15^\circ$ .

Схема моделирования с использованием пакета прикладных программ Matlab представлена на рисунке 1.5.

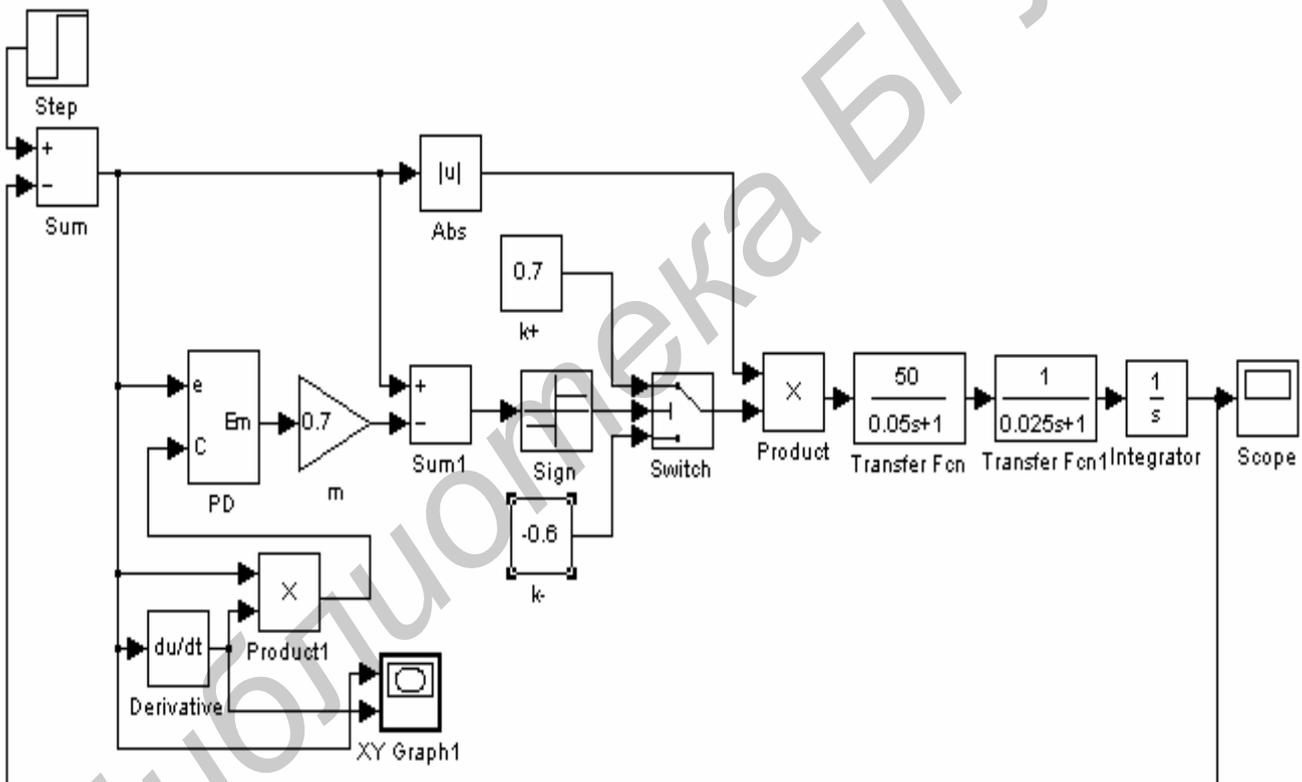


Рисунок 1.5 – Модель системы, набранная в пакете Matlab

Для определения максимального значения ошибки используется пиковый детектор. Он является основным функциональным элементом схемы. На рисунке 1.5 он выделен в подсистему и обозначен как PD. Развернутое описание блока представлено на рисунке 1.6.

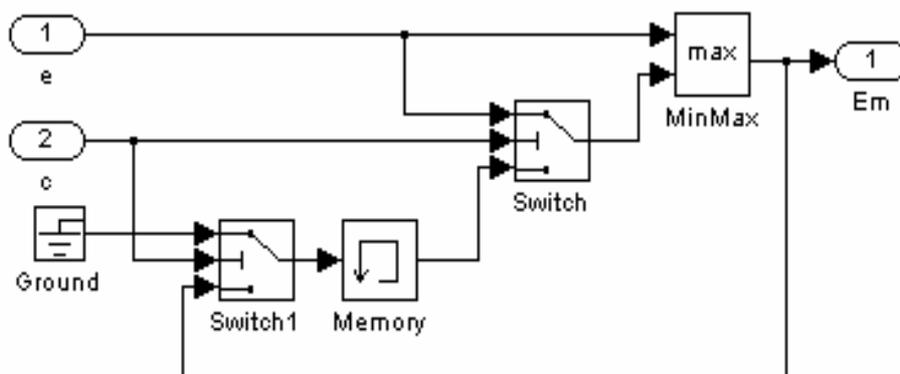


Рисунок 1.6 – Пиковый детектор

### 1.3 Варианты заданий

Параметры передаточной функции ОУ  $K_{OY}(s)$  необходимо взять для своего варианта из таблицы 1.1.

Исходные данные для ОУ могут быть изменены преподавателем.

Таблица 1.1 – Варианты заданий

Номер варианта	Передаточная функция объекта управления					
	$\frac{k_V}{s(T_1s + 1)(T_2s + 1)}$					
	$k_V$	$T_1$	$T_2$	$k_V$	$T_1$	$T_2$
1	50	0,025	0,05			
2	70	0,02	0,03			
3	80	0,015	0,025			
4	100	0,1	0,02			
5	110	0,08	0,015			
6				60	0,05	0,035
7				80	0,04	0,015
8				100	0,02	0,04
9				90	0,03	0,02
10				110	0,01	0,1
11	60	0,04	0,06			
12	80	0,04	0,025			
13	100	0,03	0,07			
14	90	0,035	0,055			
15	110	0,01	0,12			

## 1.4 Содержание отчёта

1 Цель работы.

2 Переходные процессы и фазовые портреты нескорректированной и скорректированной САУ для различных значений  $m$ ,  $k^+$ ,  $k^-$ .

3 Выводы о влиянии параметров  $m$ ,  $k^+$ ,  $k^-$  на качественные показатели системы.

## 1.5 Контрольные вопросы

1 Почему используют многократное чередование режимов разгона и торможения при работе системы с ПКУЗЭ?

2 Можно ли с помощью ПКУЗЭ реализовать САУ, оптимальную по быстрдействию?

3 Как следует выбирать значения величин  $m$ ,  $\gamma$ ,  $b$ ?

Библиотека БГУИР

## Лабораторная работа №2

### СИНТЕЗ ОПТИМАЛЬНЫХ РЕГУЛЯТОРОВ ДЛЯ НЕЛИНЕЙНЫХ ОБЪЕКТОВ УПРАВЛЕНИЯ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ПРИТЯГИВАЮЩИХ ИНВАРИАНТНЫХ МНОГООБРАЗИЙ

**Цель** – ознакомиться с методом применения притягивающих многообразий для аналитического конструирования оптимальных регуляторов и исследовать качественные показатели нелинейной системы с оптимальным регулятором.

#### 2.1 Краткие теоретические сведения

Поведение многих промышленных электромеханических объектов управления можно представить в виде системы нелинейных дифференциальных уравнений с линейным вхождением сигнала управления. Например, возмущенное движение ОУ часто записывают в следующем виде:

$$\begin{cases} \dot{x}_j = f_j(x_1, x_2, \mathbf{K}, x_n) + a_{j+1}x_{j+1}; & j = \overline{1, n-1}, \\ \dot{x}_n = f_n(x_1, x_2, \mathbf{K}, x_n) + U, \end{cases} \quad (2.1)$$

где  $x_j (j = \overline{1, n})$  – координаты состояния объекта;

$U$  – управляющее воздействие;

$a_{j+1} (j = \overline{1, n-1})$  – коэффициенты;

$f_j(x_1, x_2, \mathbf{K}, x_n)$  – нелинейные функции, дифференцируемые по своим аргументам, причём  $f_j(0, 0, \mathbf{K}, 0) = 0, j = \overline{1, n}$ .

Задача синтеза оптимального управления для ОУ (2.1) с использованием желаемого притягивающего многообразия заключается в определении закона управления с обратными связями

$$U = u(x_1, x_2, \mathbf{K}, x_n), \quad (2.2)$$

который сначала обеспечивает сближение изображающей точки, находящейся в произвольном начальном состоянии, с выбранным притягивающим многообразием, а затем желаемое устойчивое движение вдоль него к началу координат фазового пространства. При этом на траекториях движения должен минимизироваться обобщенный интегральный функционал

$$I = \int_0^{\infty} F(\mathbf{y}, \dot{\mathbf{y}}) dt = \int_0^{\infty} (m^2 \mathbf{y}^2 + c^2 \dot{\mathbf{y}}^2) dt, \quad (2.3)$$

где  $\psi(x_1, x_2, \mathbf{K}, x_n)$  – некоторая произвольная непрерывная и дифференцируемая по своим аргументам функция фазовых координат и, возможно, управления;

$m, c$  – положительные коэффициенты.

Необходимое уравнение оптимальности для функционала (2.3):

$$F(y, \dot{y}) = \dot{y} \frac{\partial F(y, \dot{y})}{\partial \dot{y}}$$

приводит к уравнению экстремалей

$$m^2 y^2 + c^2 \dot{y}^2 = 2c^2 \dot{y}^2,$$

которое можно переписать в виде

$$(T\dot{y} - y)(T\dot{y} + y) = 0, \quad (2.4)$$

где  $T = c/m$ .

Выражение (2.4) явно свидетельствует о наличии неустойчивого и устойчивого подсемейств экстремалей. Для устойчивых экстремалей функционала (2.3) справедливо уравнение

$$T\dot{y} + y = 0, \quad (2.5)$$

которое имеет решение

$$\psi(t) = y_0 e^{-\frac{t}{T}}. \quad (2.6)$$

Если начальные условия (НУ)  $x_{10}, x_{20}, \dots, x_{n0}$  таковы, что удовлетворяется тождество  $\psi_0 = 0$ , то согласно решению (2.6) для любого текущего времени  $t \neq 0$  будет выполняться условие  $\psi(x) = 0$ . Таким образом, множество  $\psi(x) = 0$  является инвариантным притягивающим многообразием в том смысле, что если изображающая точка попадет на него в начальный момент времени, то множество  $\psi(x) = 0$  будет представлять собой траектории движения системы для любого текущего времени. Поэтому многообразие

$$\psi(x) = \psi(x_1, x_2, \dots, x_n) \quad (2.7)$$

можно рассматривать как заданную программу движения САУ, у которой координаты  $x_1(t), x_2(t), \mathbf{K}, x_n(t)$  удовлетворяют уравнению (2.7). Это означает, что область притяжения программного (невозмущенного) движения  $\psi(x) = 0$  является вся область фазового пространства.

Для решения сформулированной задачи синтеза оптимального регулятора определим полную производную

$$\dot{y} = \frac{dy}{dt} = \sum_{i=1}^n \frac{\partial y(x_1, x_2, \mathbf{K}, x_n)}{\partial x_i} \dot{x}_i,$$

которая с учетом уравнения ОУ (2.1) примет вид

$$\dot{y} = \sum_{i=1}^n \frac{dy}{dx_i} f_i + \sum_{i=1}^{n-1} \frac{\partial y}{\partial x_i} a_{i+1} x_{i+1} + \frac{dy}{dx_n} U. \quad (2.8)$$

Подставляя значение производной  $\dot{y}$  (2.8) в уравнение для устойчивых экстремалей (2.5), получим основное функциональное уравнение

$$T \sum_{i=1}^n \frac{dy}{dx_i} f_i + T \sum_{i=1}^{n-1} \frac{\partial y}{\partial x_i} a_{i+1} x_{i+1} + T \frac{dy}{dx_n} U + y = 0, \quad (2.9)$$

из которого можно определить искомый оптимальный закон управления (2.2), доставляющий минимум функционалу (2.3) на траекториях замкнутой САУ.

Функциональное уравнение (2.9) является необходимым и достаточным условием для того, чтобы соотношение (2.7) было инвариантным притягивающим многообразием синтезированной системы.

Процедуру синтеза оптимальных законов управления (2.2) на основе притягивающих многообразий обычно выполняют в три этапа:

1) для известного ОУ (2.1) и выбранного функционала качества (2.3) задаются функцией  $\psi(x)$ , например, в виде

$$\psi(x_1, x_2, \dots, x_n) = \sum_{i=1}^n b_i x_i + j(x_1, x_2, \mathbf{K}, x_{n-1}), \quad (2.10)$$

где  $b_i$ , ( $i = \overline{1, n}$ ) – настроечные параметры;

$j(x_1, x_2, \mathbf{K}, x_{n-1})$  – произвольная, непрерывно дифференцируемая по своим аргументам нелинейная функция;

2) определяют частные производные  $\partial y / \partial x_i$  ( $i = \overline{1, n}$ ) и подставляют их вместе с функцией (2.10) в основное функциональное уравнение (2.9), откуда находят искомый оптимальный закон управления  $U(x)$ ;

3) из условия  $\psi(x)=0$  выражают координату  $x_n$  и, подставив ее в первые  $n - 1$  уравнений объекта (2.1), получают систему

$$\dot{x}_i = f_i(x_1, x_2, \mathbf{K}, x_{n-1}, b_i) \quad (i = \overline{1, n-1}),$$

которая описывает движение изображающей точки вдоль притягивающего многообразия  $\psi(x)=0$  к началу координат фазового пространства. Выбором зна-

чений настроечных параметров  $\epsilon_1, \epsilon_2, \dots, \epsilon_n$  и вида произвольной функции  $j(x_1, x_2, \mathbf{K}, x_{n-1})$  этому движению придают желаемый характер.

Пусть, например, ОУ типа (2.1) описывается нелинейной системой дифференциальных уравнений третьего порядка

$$\begin{cases} \dot{\mathbf{x}}_1 = x_2, \\ x_2 = \sin x_1 + x_3, \\ \dot{\mathbf{x}}_3 = U. \end{cases} \quad (2.11)$$

Для объекта (2.11) необходимо синтезировать оптимальный по критерию (2.3) закон управления  $U(x_1, x_2, x_3)$ , который обеспечивает сближение изображающей точки, находящейся в произвольном начальном состоянии, с притягивающим инвариантным многообразием

$$\psi(x_1, x_2, x_3) = b_1 x_1 + b_2 x_2 + b_3 x_3 + j(x_1, x_2) = 0, \quad (2.12)$$

где для настроечного параметра  $b_3$  примем для простоты условие  $b_3 = 1$ .

Подставив частные производные

$$\partial \psi / \partial x_1 = b_1 + \partial j(x_1, x_2) / \partial x_1; \quad \partial \psi / \partial x_2 = b_2 + \partial j(x_1, x_2) / \partial x_2; \quad \partial \psi / \partial x_3 = 1$$

в функциональное уравнение (2.9), с учетом (2.11) и (2.12) получим

$$\begin{aligned} T \left( b_1 + \frac{\partial j}{\partial x_1} \right) \cdot x_2 + T \left( b_2 + \frac{\partial j}{\partial x_2} \right) \cdot (\sin x_1 + x_3) + TU + \\ + b_1 x_1 + b_2 x_2 + x_3 + j(x_1, x_2) = 0, \end{aligned}$$

откуда найдем оптимальное управление  $U(x_1, x_2, x_3)$  в виде

$$\begin{aligned} U(x_1, x_2, x_3) = -\frac{b_1}{T} x_1 - \frac{b_2}{T} x_2 - \frac{1}{T} x_3 - \frac{j(x_1, x_2)}{T} - \left( b_1 + \frac{\partial j}{\partial x_1} \right) \cdot x_2 - \\ - \left( b_2 + \frac{\partial j}{\partial x_2} \right) \cdot (\sin x_1 + x_3). \end{aligned} \quad (2.13)$$

Подставив координату  $x_3 = -b_1 x_1 - b_2 x_2 - j(x_1, x_2)$  из равенства (2.12) во второе уравнение ОУ (2.11), получим систему уравнений

$$\begin{cases} \dot{\mathbf{x}}_1 = x_2, \\ \dot{\mathbf{x}}_2 = \sin x_1 - b_1 x_1 - b_2 x_2 - j(x_1, x_2), \end{cases} \quad (2.14)$$

которая описывает невозмущенное (программное) движение замкнутой системы вдоль многообразия (2.12) к началу координат фазового пространства.

Выбором значения настроечного параметра  $b_1$  и  $b_2$  вида произвольной функции  $j(x_1, x_2)$  можно придать системе желаемые свойства, например, гарантировать ее асимптотическую устойчивость в целом. Для этого, согласно (2.14), необходимо выполнить следующие условия:

$$b_1 > 0, b_2 > 0, j(x_1, x_2) = \sin x_1,$$

которые обеспечивают отрицательность производной  $\dot{L}$ .

Тогда закон управления (2.13) и уравнение движения (2.14) примут соответственно вид

$$U = -k_1 x_1 - k_2 x_2 - k_3 x_3 - k_4 \sin x_1 - x_2 \cdot \cos x_1, \quad (2.15)$$

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -b_1 x_1 - b_2 x_2, \end{cases} \quad (2.16)$$

где  $k_1 = b_1/T$ ;

$$k_2 = b_1 + b_2/T;$$

$$k_3 = b_2 + 1/T;$$

$$k_4 = b_2 + 1/T.$$

В зависимости от значений настроечных параметров можно получать различные качественные показатели замкнутой системы.

Структура системы представлена на рисунке 2.1. Функциональная схема исследуемой нелинейной САУ содержит последовательно соединенные блоки: ОУ, описываемый уравнениями (2.11), оптимальный регулятор, формирующий закон управления  $U(x_1, x_2, x_3)$  (2.15). Для исследования нескорректированной САУ, возбужденной начальными условиями на интеграторе, необходимо на вход системы подать сигнал  $x_1(t)$ . В случае анализа замкнутой скорректированной системы на вход объекта поступает сигнал  $U(x_1, x_2, x_3)$  с регулятора.

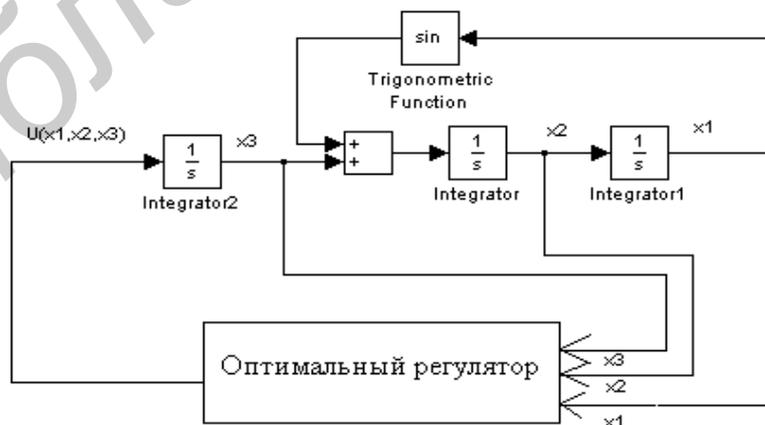


Рисунок 2.1 – Структура системы с оптимальным регулятором

Модель системы, набранная в пакете прикладных программ Matlab, представлена на рисунке 2.2.

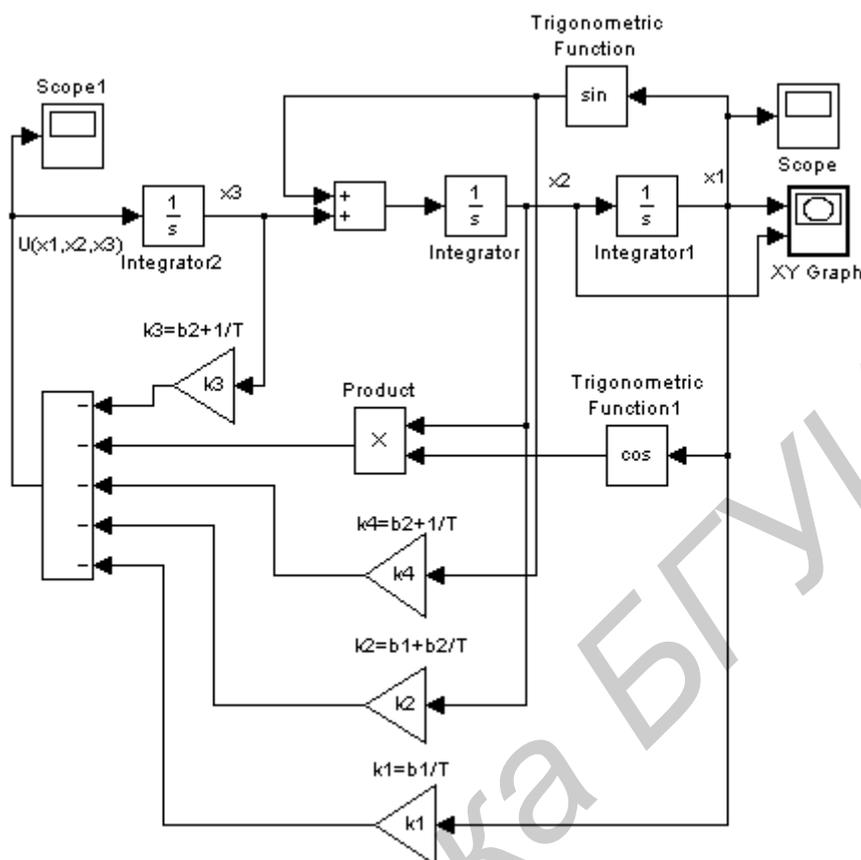


Рисунок 2.2 – Модель системы, набранная в пакете Matlab

## 2.2 Порядок и цель моделирования

В соответствии с вариантом задания получить выражение для оптимального закона управления.

Исходя из полученного закона управления, составить структуру скорректированной замкнутой системы (объект управления и регулятор) и набрать ее в пакете Matlab.

Сначала моделируется система без регулятора (возбужденная начальными условиями на интеграторе). После исследования и регистрации переходного процесса и фазового портрета ОУ моделируется система с оптимальным регулятором.

Цель моделирования заключается в сравнении выходных характеристик САУ (переходного процесса и фазового портрета) при различных значениях настроечных параметров регулятора и начальных условий на интеграторе (для каждого из параметров в варианте задания представлен диапазон их изменения).

## 2.3 Варианты заданий

ОУ описывается системой дифференциальных уравнений следующего вида:

$$b_1 \in [2, 10], b_2 \in [5, 15], T \in [0.1, 0.5]$$

$$1 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = \sin x_2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$2 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = 3x_1^3 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$3 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = (1 - \cos x_2) + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$4 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_1^2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$5 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_2^2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$6 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = \sin x_1 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$7 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_2^2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$8 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_1^2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$9 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_2^3 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$10 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_1^2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$11 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_1^3 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$12 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = (1 - \cos x_1) + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$13 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = (1 - \cos x_2) + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$14 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = \sin x_2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$15 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = 2x_2^3 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

## 2.4. Содержание отчета

- 1 Цель работы.
- 2 Процесс синтеза оптимального закона управления и его конечный вид.
- 3 Структура замкнутой скорректированной системы (набранная в пакете Matlab).
- 4 Графики переходных процессов и фазовые портреты системы при различных значениях настроечных параметров регулятора.
- 5 Анализ сравнения поведения системы при различных настройках регулятора и при различных начальных условиях.

## 2.5. Контрольные вопросы

- 1 Почему областью притяжения программного движения  $\psi(x) = 0$  является вся область фазового пространства?
- 2 Почему процедура синтеза оптимальных регуляторов на основе притягивающих многообразий оказывается эффективней классического метода АКОР?
- 3 Чем обусловлено влияние величин  $T$  и  $b_i$  на качество переходного процесса и фазовый портрет системы?
- 4 Почему нелинейная функция  $j(x_1, \dots, x_{n-1})$ , входящая в выражение (2.10), зависит только от  $n - 1$ , а не от  $n$  координат пространства состояний?

## Лабораторная работа №3

### СИНТЕЗ ДИФФЕРЕНЦИАЛЬНЫХ АНАЛИЗАТОРОВ С ИСПОЛЬЗОВАНИЕМ МЕТОДА СТРУКТУРНОГО СИНТЕЗА Л. М. БОЙЧУКА

**Цель** – ознакомиться с принципами построения и функционированием дифференциальных анализаторов.

#### 3.1 Краткие теоретические сведения

Одним из самых мощных методов синтеза нелинейных систем управления является метод структурного синтеза Л. М. Бойчука.

Сущность метода заключается в выборе структуры регулятора, обеспечивающего выполнение объектом некоторого задания. При этом появляется возможность обеспечить поставленные требования к качеству его отработки. Особенно актуальным является использование данного метода в задачах синтеза сложных нелинейных САУ, к которым, в частности, относятся системы функционального регулирования.

К таким системам можно отнести дифференциальные анализаторы – устройства для преобразования некоторых функциональных зависимостей, отображающих кривые на плоскости, в функции времени соответственно каждой геометрической координате. Эти функции являются управляющими для исполнительных механизмов, обеспечивающих перемещение по соответствующим координатам. Дифференциальные анализаторы являются одним из основных элементов систем программного управления станками, вырабатывая задания для следящих систем приводов подач.

Рассмотрим синтез системы управления двумя исполнительными устройствами для отработки плоской кривой произвольного вида.

Пусть некоторой линии на плоскости соответствует уравнение, заданное в неявной форме:

$$F(x, y) = 0, \quad (3.1)$$

где  $F(x, y) = 0$  – есть функция, имеющая частные производные первого порядка по всем своим аргументам.

Это уравнение полагаем функциональным заданием для двух исполнительных механизмов (представленных в виде интеграторов)

$$\begin{cases} \dot{x} = U_x; \\ \dot{y} = U_y, \end{cases} \quad (3.2)$$

где  $x, y$  – выходные величины;

$U_x, U_y$  – управляющие воздействия.

Необходимо найти структуру системы управления объектами (3.2), обеспечивающую отработку функциональной зависимости (3.1) устойчиво и с требуемыми показателями качества.

Ошибку отработки задания (отклонение выходных координат от заданной траектории) можно представить в следующем виде:

$$j = F(x, y). \quad (3.3)$$

Примем, что ошибка должна экспоненциально сходиться к нулю. Тогда дифференциальное уравнение для ошибки будет иметь вид

$$T \frac{dj}{dt} + j = 0. \quad (3.4)$$

Дифференцирование будем проводить относительно новой переменной  $s$ , связанной со временем соотношением

$$\frac{ds}{dt} = b_s, \quad (3.5)$$

где  $b_s$  – некоторая неотрицательная функция координат  $x, y$ , называемая коэффициентом преобразования.

Такая подстановка носит название преобразования Лиувилля. Его физический смысл заключается в следующем. Пусть задан переходный процесс в функции переменной  $s$ :  $Y = Y(s)$ , причем эта функция однозначна, непрерывна и существует при всех  $s \geq 0$  (рисунок 3.1).

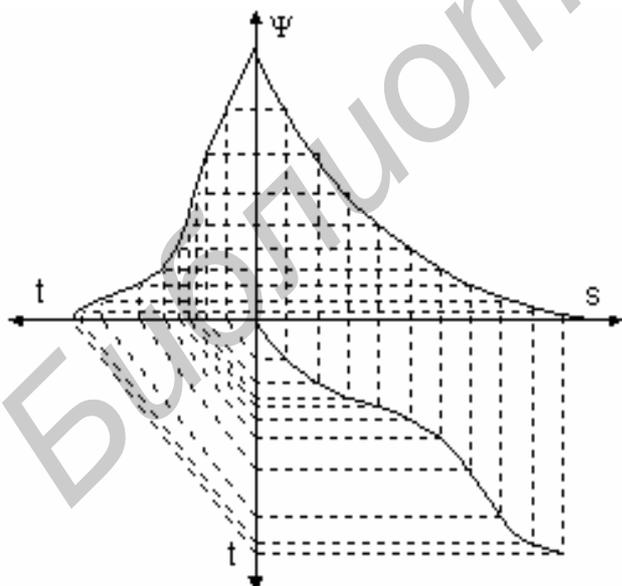


Рисунок 3.1 – Графическая интерпретация преобразования Лиувилля

Пусть задано также соотношение для перехода от переменной  $s$  (локальное время) к переменной  $t$  (реальное время), которое, учитывая выражение (3.5), следует записать в виде

$$t = \int_0^s \frac{1}{b_s} ds. \quad (3.6)$$

Для того чтобы получаемый переходный процесс во времени  $Y = Y(t)$  также представлял собой однозначную и непрерывную функцию, существующую при всех  $t \geq 0$ , преобразование (3.6) должно быть взаимно однозначным.

Подставив в уравнение (3.4) выражения (3.3), (3.5) и дифференцируя по  $s$ , получим

$$F_x U_x + F_y U_y = -\frac{b_s}{T} F,$$

где  $F_x = \frac{dF(x, y)}{dx};$

$$F_y = \frac{dF(x, y)}{dy},$$

или с учетом уравнений (3.2) имеем

$$F_x U_x + F_y U_y = -\frac{b_s}{T} F. \quad (3.7)$$

Очевидно, что выразить две переменные из одной формулы невозможно. Поэтому представим управляющие воздействия в виде линейных комбинаций частных производных  $F_x$  и  $F_y$ :

$$\begin{cases} U_x = c_{11}F_x + c_{12}F_y; \\ U_y = c_{21}F_x + c_{22}F_y. \end{cases} \quad (3.8)$$

Коэффициенты в уравнениях (3.8) найдем из условия обращения уравнения (3.7) в тождество. Подставляя выражения (3.8) в уравнение (3.7), получим

$$c_{11}F_x^2 + (c_{12} + c_{21})F_xF_y + c_{22}F_y^2 + \frac{b_s}{T}F = 0. \quad (3.9)$$

Это уравнение обратится в ноль, если принять  $b_s = F_x^2 + F_y^2 \geq 0$  и выбрать коэффициенты  $c_{12} = -c_{21} = c$  и  $c_{11} = c_{22} = -F/T$ . Тогда искомая структура управления будет иметь вид

$$\begin{cases} U_x = -\frac{1}{T}FF_x + cF_y; \\ U_y = -cF_x - \frac{1}{T}FF_y, \end{cases} \quad (3.10)$$

где  $c \neq 0$  – произвольная постоянная.

Коэффициент  $c$  определяет скорость перемещения по заданной кривой, а его знак определяет направление движения.

Значение этого коэффициента определяется из скорости движения изображающей точки вдоль кривой (см. рисунок 3.1). Следует учитывать, что в это время  $F = 0$ . Скорость перемещения изображающей точки определяется как

$$V_0^2 = \dot{x}^2 + \dot{y}^2. \quad (3.11)$$

Из уравнений (3.2), (3.10), (3.11) следует, что

$$c = \frac{V_0}{\pm \sqrt{F_x^2 + F_y^2}}. \quad (3.12)$$

С учетом выражения (3.12) перепишем закон управления (3.10):

$$\begin{cases} U_x = -\frac{1}{T} FF_x + \frac{V_0 F_y}{\sqrt{F_x^2 + F_y^2}}; \\ U_y = -\frac{V_0 F_x}{\sqrt{F_x^2 + F_y^2}} - \frac{1}{T} FF_y. \end{cases} \quad (3.13)$$

Рассмотрим пример синтеза дифференциального анализатора для обработки функциональной зависимости, отображающей овал Кассини с центром в начале координат.

Уравнение овала в декартовой системе координат имеет следующий вид:

$$F(x, y) = (x^2 + y^2)^2 - 2c^2(x^2 - y^2) - (a^4 - c^4) = 0. \quad (3.14)$$

Частные производные будут равны соответственно

$$\begin{cases} F_x = 4x(x^2 + y^2) - 4c^2x; \\ F_y = 4y(x^2 + y^2) + 4c^2y. \end{cases} \quad (3.15)$$

Подставив уравнения (3.14) и (3.15) в выражение для закона управления (3.13), получим

$$\begin{cases} U_x = -\frac{\left( (x^2 + y^2)^2 - 2c^2(x^2 - y^2) - (a^4 - c^4) \right) (4x \cdot (x^2 + y^2) - 4c^2x)}{T} + \\ \quad + \frac{V_0 (4y \cdot (x^2 + y^2) + 4c^2y)}{\sqrt{(4x \cdot (x^2 + y^2) - 4c^2x)^2 + (4y \cdot (x^2 + y^2) + 4c^2y)^2}}; \\ U_y = -\frac{V_0 (4x \cdot (x^2 + y^2) - 4c^2x)}{\sqrt{(4x \cdot (x^2 + y^2) - 4c^2x)^2 + (4y \cdot (x^2 + y^2) + 4c^2y)^2}} - \\ \quad - \frac{\left( (x^2 + y^2)^2 - 2c^2(x^2 - y^2) - (a^4 - c^4) \right) \cdot (4y \cdot (x^2 + y^2) + 4c^2y)}{T}. \end{cases} \quad (3.16)$$

Как можно заметить, синтезированная система с управлениями (3.16) является существенно нелинейной. В ее структуре будут присутствовать блоки деления, умножения и др.

Динамические показатели работы системы определяются значениями параметров  $T$  и  $V_0$ . При этом  $T$  влияет на качество обработки функциональной зависимости (это особенно заметно в случае воздействия помехи), а параметр  $V_0$  определяет скорость обработки заданной кривой.

Функциональная схема синтезированного дифференциального анализатора, набранная в пакете Matlab (рисунок 3.2), состоит из двух исполнительных механизмов, представленных интеграторами (3.2) и существенно нелинейной частью системы управления, формирующей управляющие воздействия для исполнительных механизмов (3.16). К выходу интеграторов через переключатель Switch прикладываются возмущающие воздействия, которые представляют собой белый шум с интенсивностью, равной 0,001. Переключатель управляется сигналом  $u = \pm 1$  (второй вход ключа).

Если  $u = -1$ , то переключатель не пропускает помеху на выход интеграторов, т.е. система работает в идеальном режиме. Если  $u = 1$ , то на выход исполнительных механизмов подается возмущающее воздействие.

В этом случае мы исследуем работу системы в реальном режиме (при наличии помех). При этом основные параметры были выбраны следующими:  $a = 2$ ,  $c = 1$ ,  $T = 0,02$  с,  $V_0 = 10$ .

### 3.2 Порядок и цель моделирования

Сначала моделируется система без учета влияния помех. После исследования и регистрации функциональной зависимости (снимается в виде фазового портрета) моделируется система в условиях воздействия помех.

Цель моделирования заключается в сравнении качества обработки системой заданной функциональной зависимости при различных значениях параметра  $T = 0,2$ ,  $T = 0,02$  и  $T = 0,002$ .

Предлагается синтезировать схему ДА для обработки одной из предложенных кривых, заданных в неявном виде:

– окружность:

$$F(x, y) = x^2 + y^2 - 1 = 0;$$

– овал Кассини:

$$F(x, y) = (x^2 + y^2)^2 - 2c^2(x^2 - y^2) - (a^4 - c^4) = 0, \\ a, c = \text{const}, a = 2, c = 1;$$

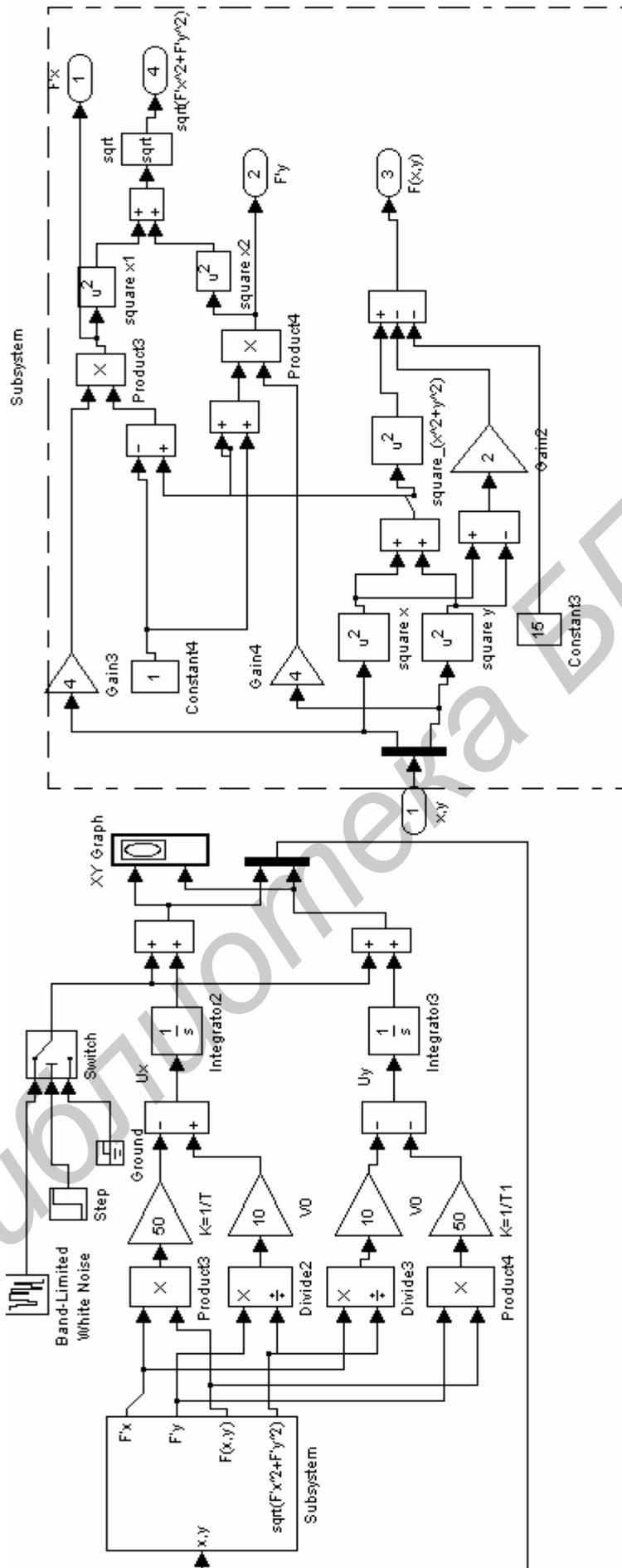


Рисунок 3.2 – Функциональная схема синтезированного дифференциального анализатора

– лемниската Бернулли:

$$F(x,y) = (x^2 + y^2)^2 - 2c^2(x^2 - y^2) = 0, \quad c = 1;$$

– кардиоида:

$$F(x,y) = (x^2 + y^2 + 2rx)^2 - 4r^2(x^2 + y^2) = 0, \quad r = 1;$$

– розы трехлепестковой:

$$F(x,y) = (x^2 + y^2)^2 - a(3x^2y - y^3) = 0, \quad a = 2;$$

– розы четырехлепестковой:

$$F(x,y) = (x^2 + y^2)^3 - 4a^2x^2y^2 = 0, \quad a = 2.$$

### 3.3 Содержание отчета

1 Цель работы.

2 Графики функциональной зависимости (фазовый портрет), отрабатываемой синтезированным дифференциальным анализатором при различных значениях параметра  $V_0$  и  $T$  системы управления.

3 Сравнения поведения системы при работе в идеальном и реальном режимах работы. Анализ влияния параметра  $T$  на качество работы системы в условиях помех.

### 3.4 Контрольные вопросы

1 В чем преимущество ДА, синтезированных по замкнутой схеме?

2 Какова физическая сущность переменной  $s$  в преобразовании Лиувилля?

3 Почему должно выполняться условие  $b_s \geq 0$ ?

4 Как влияет параметр  $T$  на качество работы системы?

5 Какова последовательность синтеза нелинейных систем управления по методу Бойчука?

## Лабораторная работа №4

### СИНТЕЗ РЕГУЛЯТОРОВ С ПЕРЕМЕННОЙ СТРУКТУРОЙ ДЛЯ УПРАВЛЕНИЯ НЕЛИНЕЙНЫМ ОБЪЕКТОМ

**Цель** – ознакомиться с принципами построения и функционированием регуляторов с переменной структурой для управления нелинейными объектами.

#### 4.1 Краткие теоретические сведения

В последнее время существенно возрос интерес к исследованию систем с переменной структурой (СПС). Это связано с тем, что подобные системы обладают свойством грубости (нечувствительности к изменению параметров системы и инвариантностью к внешним возмущениям).

Задача синтеза в СПС обычно сводится к выбору гиперповерхностей в фазовом пространстве, на которых функции управления будут претерпевать разрывы. При выполнении определенных соотношений, о которых будет сказано ниже, в таких системах с необходимостью возникает специфический вид движений – так называемый скользящий режим (СР).

Скользящий режим – это особое (вырожденное) движение САУ, при котором изображающая точка колеблется с бесконечно большой частотой в некоторой малой окрестности гиперповерхности разрыва (или на пересечении гиперповерхностей). Следует отметить, что движение системы в скользящем режиме описывается дифференциальными уравнениями (ДУ) меньшего порядка, чем собственные движения системы. При этом качественные показатели синтезированной САУ обуславливаются только положением гиперповерхности переключения и не зависят от собственных параметров системы.

Рассмотрим условия, при которых в СПС будет существовать скользящий режим. Пусть

$$S = S(x_1, x_2, \dots, x_n) = 0 \quad (4.1)$$

есть уравнение гиперповерхности переключения.

В этом случае можно рассматривать  $S(x_1, x_2, \dots, x_n)$  как отклонение изображающей точки от этой гиперповерхности.

Для того чтобы в системе существовал СР, необходимо сформировать управление таким образом, чтобы изображающая точка, один раз попав на гиперповерхность переключения, уже не могла сойти с нее и дальше двигалась только вдоль этой поверхности.

Пусть в некоторый момент времени  $S > 0$ . Тогда производная по времени  $dS/dt$  (скорость изменения  $S$ ) должна быть отрицательна. В этом случае  $S$  ( $S$  – отклонение от гиперповерхности разрыва) будет убывать до нуля, т.е. изображающая точка будет двигаться по направлению к  $S$ . Пусть далее изо-

бражающая точка перескочила поверхность  $S = 0$  и теперь  $S < 0$ . Соответственно должно измениться и значение производной, а именно: должно выполняться условие  $dS/dt > 0$ . В этом случае изображающая точка будет двигаться обратно по направлению к поверхности скольжения.

Исходя из всего вышесказанного, можно записать уравнение существования скользящего режима:

$$S \frac{dS}{dt} < 0. \quad (4.2)$$

Таким образом, можно выделить 2 стадии в процессе синтеза СПС, работающие в скользящем режиме:

- определение закона управления и выбор поверхности (поверхностей) переключения, на которых функция управления будет претерпевать разрывы;
- анализ существования в системе СР.

Описанный подход обладает одним существенным недостатком – до сих пор не существует общих методов, позволяющих выбрать поверхность переключения для нелинейного объекта таким образом, чтобы синтезированная САУ работала в скользящем режиме.

Для преодоления этого недостатка предложено определять выражение для закона управления, исходя из выполнения условия существования СР. Это условие представляется следующим образом:

$$S' \cdot \text{sign}S = -kW(x_1, x_2, \dots, x_n), \quad (4.3)$$

где  $W(\bar{x})$  – положительно-определенная функция фазовых переменных такая, что

$$\begin{cases} W(\bar{x}) = 0, & \text{при } \bar{x} = 0; \\ W(\bar{x}) \neq 0, & \text{при } \bar{x} \neq 0. \end{cases}$$

Уравнение гиперплоскости скольжения  $S$  выберем в виде линейной комбинации фазовых координат системы:

$$S(x_1, x_2, \dots, x_n) = \sum_{i=1}^n c_i x_i, \quad (4.4)$$

где  $c_i$  – некоторые коэффициенты, определяющие положение гиперповерхности переключения в пространстве.

При движении вдоль гиперповерхности переключения система будет описываться линейным дифференциальным уравнением, соответствующим выражению (4.4):

$$\sum_{i=1}^n c_i x^{(i-1)} = 0, \quad (4.5)$$

где  $x^{(i-1)}$  – производная  $(i-1)$ -го порядка.

Можно заметить, что при движении в скользящем режиме вдоль гиперповерхности (4.4) будет происходить линеаризация нелинейной системы.

Для того чтобы система устойчиво двигалась в скользящем режиме, необходимо выбирать коэффициенты  $c_i (i=1, n)$  таким образом, чтобы линейное дифференциальное уравнение (4.5) имело устойчивое решение. Для проверки устойчивости можно использовать любой алгебраический критерий (Гурвица, Рауса и др.).

В качестве функции  $W$  можно принять норму вектора фазовых координат

$$W(\bar{x}) = |x_1| + |x_2| + \dots + |x_n|. \quad (4.6)$$

Тогда выражение (4.3) с учетом (4.6) примет вид

$$S' \text{sign} S = -k(|x_1| + |x_2| + \dots + |x_n|). \quad (4.7)$$

Если домножить обе части (4.7) на  $\text{sign} S$ , то можно заметить, что производная

$$S' = -k(|x_1| + |x_2| + \dots + |x_n|) \text{sign} S \quad (4.8)$$

постоянно будет иметь знак, противоположный знаку  $S$ . Таким образом, будет постоянно выполняться условие существования скользящего режима, т.е. система будет входить в СР при любых значениях начальных условий.

Теперь рассмотрим синтез закона управления, обеспечивающего существование скользящего режима для существенно нелинейного объекта, описываемого системой ДУ вида

$$\begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = \sin x_1 + x_3; \\ \dot{x}_3 = U(x_1, x_2, x_3). \end{cases} \quad (4.9)$$

Гиперповерхность  $S(x_1, x_2, x_3)$  выберем следующего вида:

$$S(x_1, x_2, x_3) = c_1 x_1 + c_2 x_2 + c_3 x_3, \quad (4.10)$$

где коэффициент  $c_3$  для простоты примем равным единице, а коэффициенты  $c_1$  и  $c_2$  выберем из условия симметрирования в пространстве гиперплоскости скольжения:  $c_1 = c_2 = c$ ;  $c > 0$ . В этом случае будет удовлетворяться условие устойчивости решения соответствующего дифференциального уравнения.

Уравнение гиперповерхности скольжения в этом случае запишется следующим образом:

$$S(x_1, x_2, x_3) = cx_1 + cx_2 + x_3. \quad (4.11)$$

Полная производная по времени  $dS(x_1, x_2, x_3)/dt$ , взятая в силу системы (4.9), будет иметь вид

$$dS/dt = cx_2 + c \sin x_1 + cx_3 + U(x_1, x_2, x_3). \quad (4.12)$$

Подставим полученное выражение для производной в (4.8), откуда найдем выражение для закона управления

$$U(x_1, x_2, x_3) = -k(|x_1| + |x_2| + |x_3|) \operatorname{sign} S - cx_2 - c \sin x_1 - cx_3. \quad (4.13)$$

Полученный алгоритм управления содержит в себе операции взятия модуля и присвоения знака, которые легко реализуются с помощью цифровой системы управления. Кроме этого следует отметить, что нелинейности объекта управления входят в функцию управления непосредственно, т.е. не требуется обращать или дифференцировать нелинейности, входящие в структуру объекта. Это является существенным преимуществом предложенного метода перед другими методами синтеза нелинейных САУ.

Функциональная схема состоит из объекта управления, представляющего собой перевернутый маятник (4.9), и нелинейного регулятора, формирующего требуемый разрывной закон управления  $U(x_1, x_2, x_3)$  (4.13).

Регулятор состоит из линейной части (включающей в себя обратные связи по фазовым координатам  $x_2, x_3$ ) и нелинейной части (включающей нелинейные блоки  $\sin(\cdot)$  и модуль).

Модель системы, набранная в пакете прикладных программ Matlab, представлена на рисунке 4.1.

## 4.2 Порядок и цель моделирования

В соответствии с вариантом задания получить выражение для оптимального закона управления  $U(x_1, x_2, x_3)$ .

Исходя из полученного закона управления, составить структуру скорректированной замкнутой системы (объект управления и регулятор) и набрать ее в пакете Matlab.

Цель моделирования заключается в сравнении динамики работы синтезированной САУ при различных значениях ( $k = c; k < c; k > c$ ) параметров синтезированного регулятора. Значения коэффициентов рекомендуется принимать следующие:  $k \in [3, 10]; c \in [3, 7]$ .

Для получения более полного представления о динамике процессов, происходящих в синтезированной системе, необходимо снять следующие зависи-

мости: переходной процесс по координате  $x_1$ , фазовый портрет в плоскости координат  $x_1$  и  $x_2$ , функцию управления  $U$ .

В качестве дополнительного задания можно предложить промоделировать систему с несколько модифицированным законом управления. А именно: можно выбрать функцию  $W(\bar{x})$  иного вида, например  $W(\bar{x}) = \sqrt{x_1^2 + x_2^2 + x_3^2}$  либо любую другую.

При этом необходимо учитывать условие, согласно которому функция  $W(\bar{x})$  должна быть положительно определена. Процедура синтеза в этом случае полностью соответствует описанной в лабораторной работе.

### 4.3 Варианты заданий

ОУ описывается системой дифференциальных уравнений следующего вида:

$$1 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = \sin x_2 + x_3; \\ \dot{x}_3 = u. \end{cases} \quad 2 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = 3x_1^3 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$3 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = (1 - \cos x_2) + x_3; \\ \dot{x}_3 = u. \end{cases} \quad 4 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_1^2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$5 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_2^2 + x_3; \\ \dot{x}_3 = u. \end{cases} \quad 6 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = \sin x_1 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$7 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_2^2 + x_3; \\ \dot{x}_3 = u. \end{cases} \quad 8 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_1^2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$9 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_2^3 + x_3; \\ \dot{x}_3 = u. \end{cases} \quad 10 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_1^2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$11 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = x_1^3 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$12 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = (1 - \cos x_1) + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$13 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = (1 - \cos x_2) + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$14 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = \sin x_2 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

$$15 \begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = 2x_2^3 + x_3; \\ \dot{x}_3 = u. \end{cases}$$

#### 4.4 Содержание отчета

1 Цель работы.

2 Графики переходных процессов, фазовые портреты системы и график изменения функции управления при различных параметрах синтезированного регулятора ( $k = c$ ;  $k < c$ ;  $k > c$ ).

3 Анализ сравнения поведения системы в различных режимах работы при различных значениях параметров  $k$  и  $c$ .

#### 4.5 Контрольные вопросы

1 Что представляет собой движение изображающей точки в скользящем режиме?

2 Каковы достоинства и недостатки скользящих режимов?

3 Каким требованиям должно удовлетворять уравнение гиперповерхности скольжения?

4 Почему функция  $W(x)$  должна быть положительно определенной?

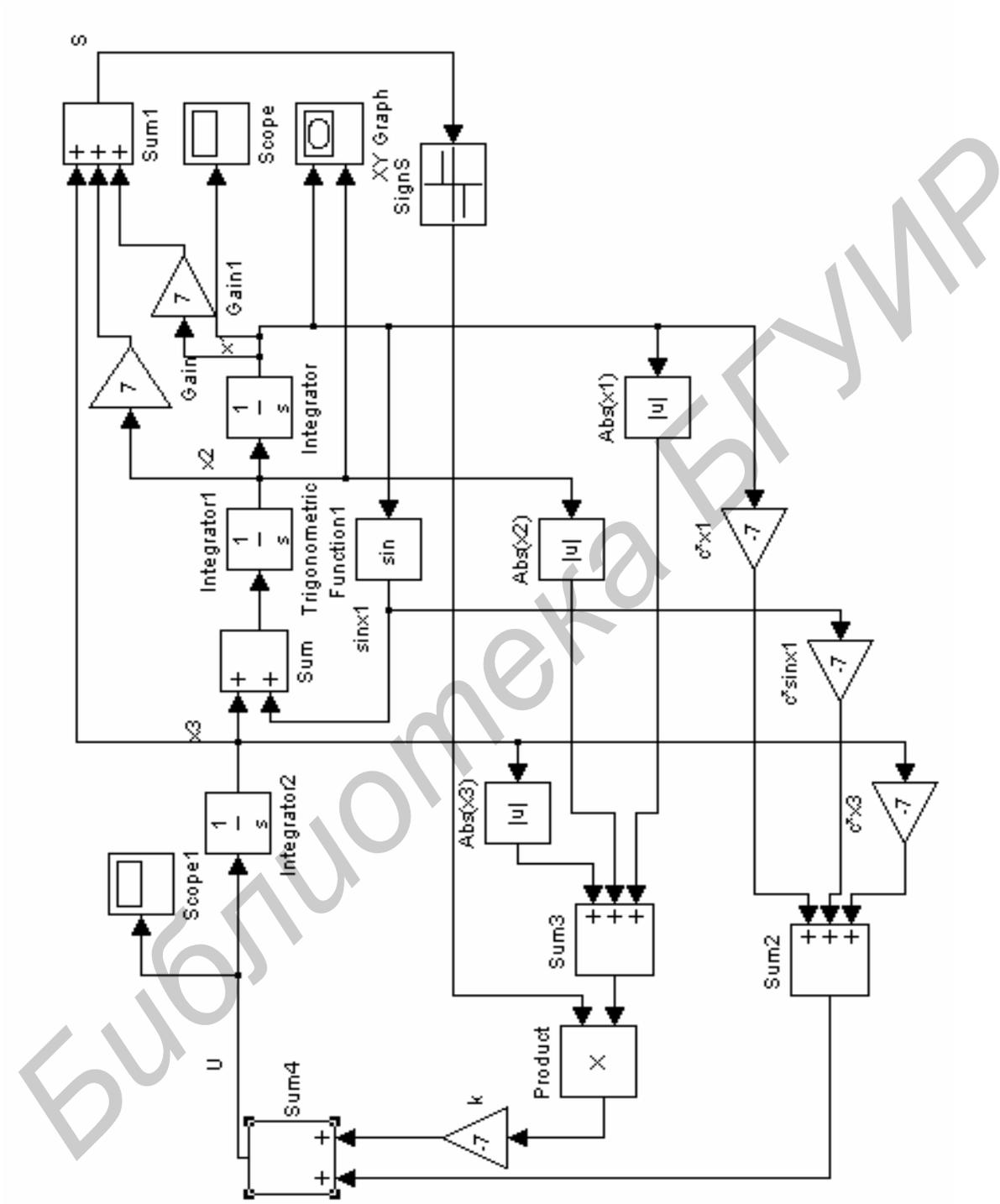


Рисунок 4.1 – Модель системы с переменной структурой

## Лабораторная работа №5

### СИНТЕЗ УПРАВЛЯЮЩИХ СИСТЕМ НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ

**Цель** – изучить основные понятия теории нечетких множеств и их приложений. Ознакомиться с пакетом математических приложений Matlab.

#### 5.1 Краткие теоретические сведения

Пусть  $E$  – универсальное множество,  $x$  – элемент  $E$ ,  $P$  – некоторое свойство. Обычное (четкое) множество  $A$  универсального множества  $E$ , элементы которого удовлетворяют свойству  $P$ , определяется как множество упорядоченных пар

$$A = \{m_A(x) / x\}, \quad (5.1)$$

где  $m_A(x)$  – характеристическая функция, принимающая значение 1, если  $x$  удовлетворяет свойству  $P$ , и 0 в противном случае.

В теории нечетких множеств для элементов  $x$  из  $E$  нет однозначного ответа «да/нет» относительно свойства  $P$ . В связи с этим нечеткое множество  $A$  универсального множества  $E$  определяется как множество упорядоченных пар с функцией принадлежности  $m_A(x)$ , принимающей значение в некотором упорядоченном множестве  $M$  (например  $M = [0, 1]$ ).

Функция принадлежности указывает степень (или уровень) принадлежности элемента  $x$  подмножеству  $A$ . Множество  $M$  называют множеством принадлежностей. Если  $M = \{0, 1\}$ , то нечеткое подмножество  $A$  может рассматриваться как обычное или четкое множество.

*Примеры записи нечеткого множества*

Пусть  $E = \{x_1, x_2, x_3, x_4, x_5\}$ ,  $A$  – нечеткое множество, для которого  $m_A(x_1) = 0,3$ ;  $m_A(x_2) = 0$ ;  $m_A(x_3) = 1$ ;  $m_A(x_4) = 0,6$ ;  $m_A(x_5) = 0,9$ .

Тогда  $A$  можно представить в виде  $A = \{0,3/x_1; 0/x_2; 1/x_3; 0,6/x_4; 0,9/x_5\}$ .

Такой способ используется для измеряемых понятий (скорость, температура и т.д.) или при выделении полярных значений. Разновидностью прямого способа задания функции принадлежности является групповой метод, когда группе экспертов предлагают сделать оценку того или иного явления, например, оценить: «этот человек лысый» или нет, – тогда количество утвердительных ответов, деленное на общее число экспертов, дает значение  $\mu_{\text{лысый}}$  для данного лица.

Кроме указанных способов задания функций принадлежности используют также типовые формы функций принадлежности (рисунок 5.1).

Аналитическая запись некоторых типовых функций принадлежности:

– треугольная:

$$m(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right), \quad (5.2)$$

где  $a, b, c$  – определяют параметры функции;

– гауссова:

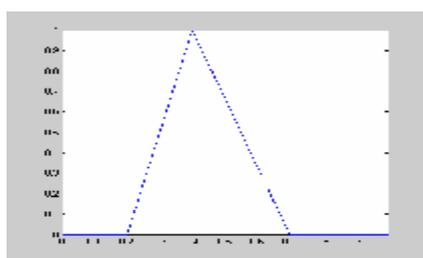
$$m(x) = e^{-\left(\frac{x-a}{b}\right)^2}, \quad (5.3)$$

где  $a, b$  – параметры функции принадлежности;

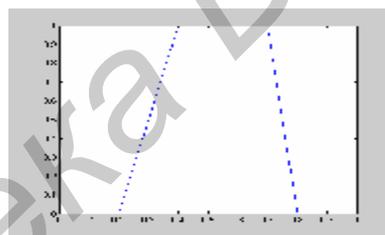
– сигмоидная:

$$m(x) = \frac{1}{1 + \exp(-a(x-b))}, \quad (5.4)$$

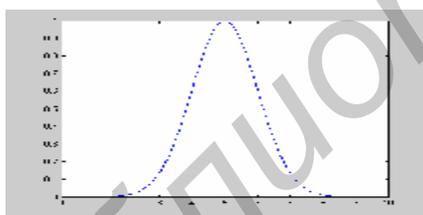
где  $a, b$  – параметры функции принадлежности.



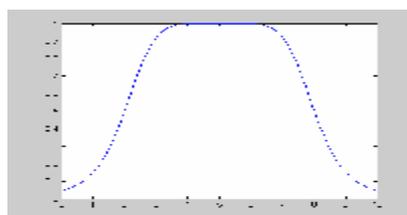
а



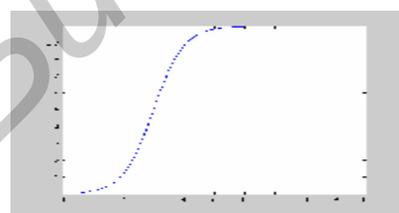
б



в



г



д

а – треугольная (trimf), б – трапецеидальные (trapmf), в – гауссова (gaussmf), г – обобщенная колоколообразная (gbellmf), д – сигмоидная (sigmf)

Рисунок 5.1 – Примеры типовых форм функций принадлежности

Выделяют основные операции над нечеткими множествами:

- Включение. Обозначение:  $A \subset B$ .
- Равенство. Обозначение:  $A = B$ .  $A$  и  $B$  равны, если

$$\forall x \in E \quad m_A(x) = m_B(x). \quad (5.5)$$

- Дополнение. Обозначение:  $B = \bar{A}$  или  $A = \bar{B}$ .
- Пусть  $M = [0,1]$ ,  $A$  и  $B$  дополняют друг друга, если

$$\forall x \in E \quad m_A(x) = 1 - m_B(x). \quad (5.6)$$

- Пересечение.  $A \cap B$  – наименьшее нечеткое подмножество, содержащееся одновременно в  $A$  и  $B$ :

$$m_{A \cap B}(x) = \min(m_A(x), m_B(x)) = m_A(x) \% m_B(x). \quad (5.7)$$

В экспертных и управляющих системах механизм нечетких выводов в своей основе имеет базу знаний, формируемую специалистами предметной области в виде совокупности нечетких предикатных правил вида:

$$\begin{aligned} \Pi_1: & \text{если } x \text{ есть } A_1, \text{ то } y \text{ есть } B_1, \\ \Pi_2: & \text{если } x \text{ есть } A_2, \text{ то } y \text{ есть } B_2, \\ & \dots \\ \Pi_n: & \text{если } x \text{ есть } A_n, \text{ то } y \text{ есть } B_n, \end{aligned}$$

где  $x$  – входная переменная,  
 $y$  – переменная вывода,

$A$  и  $B$  – функции принадлежности, определенные на  $x$  и  $y$  соответственно.

Знания эксперта отражает нечеткое причинное отношение предпосылки и заключения, поэтому его называют нечетким отношением:

$$R = A \rightarrow B, \quad (5.8)$$

где « $\rightarrow$ » – нечеткая импликация.

Отношение  $R$  можно рассматривать как нечеткое подмножество прямого произведения  $X \times Y$  полного множества предпосылок  $X$  и заключений  $Y$ .

Таким образом, процесс получения (нечеткого) результата вывода  $B'$  с использованием данного наблюдения  $A'$  и значения  $A \rightarrow B$  можно представить в виде

$$B' = A' \bullet R = A' \bullet (A \rightarrow B).$$

### Алгоритм нечеткого вывода

1 Нечеткость (фазификация, fuzzification). Функции принадлежности, определенные для входных переменных, применяются к их фактическим значениям для определения степени истинности каждой предпосылки каждого правила.

2 Логический вывод. Вычисленное значение истинности для предпосылок каждого правила применяется к заключениям каждого правила. Это приводит к одному нечеткому подмножеству, которое будет назначено переменной вывода для каждого правила. В качестве правил логического вывода используются только операции *min* (минимума) или *prod* (умножение).

3 Композиция. Нечеткие подмножества, назначенные для каждой переменной вывода (во всех правилах), объединяются вместе, чтобы сформировать одно нечеткое подмножество для каждой переменной вывода. При подобном объединении обычно используются операции *max* (максимум) или *sum* (сумма).

4 Дефазификация – приведение к четкости (defuzzification). Преобразование нечеткого набора выводов в число.

### Алгоритмы нечеткого вывода Мамдани (Mamdani)

Пусть заданы два нечетких правила:

$P_1$ : если  $x$  есть  $A_1$  и  $y$  есть  $B_1$ , тогда  $z$  есть  $C_1$ ,

$P_2$ : если  $x$  есть  $A_2$  и  $y$  есть  $B_2$ , тогда  $z$  есть  $C_2$ .

1 Нечеткость. Находят степени принадлежности для предпосылок каждого правила:  $A_1(x_0)$ ,  $A_2(x_0)$ ,  $B_1(y_0)$ ,  $B_2(y_0)$ .

2 Нечеткий вывод. Определяют уровни «отсечения» для предпосылок каждого правила (операция *min*):

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0);$$

определяют усеченные функции принадлежности

$$C'_1 = (\alpha_1 \wedge C_1(z)),$$

$$C'_2 = (\alpha_2 \wedge C_2(z)).$$

3 Композиция. Производится объединение найденных усеченных функций (операция *max*), получают нечеткое подмножество для переменной выхода с функцией принадлежности:

$$m_z(z) = C(z) = C'_1(z) \vee C'_2(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z)). \quad (5.9)$$

4 Дефазификация. Приведение к четкости (определение  $z_0$ ), например, центроидным методом (как центр тяжести для кривой  $m_z(z)$ ):

$$z_0 = \frac{\int_{\Omega} z m_z(z) dz}{\int_{\Omega} m_z(z) dz}. \quad (5.10)$$

Алгоритм Мамдани графически выглядит следующим образом (рисунок 5.2).

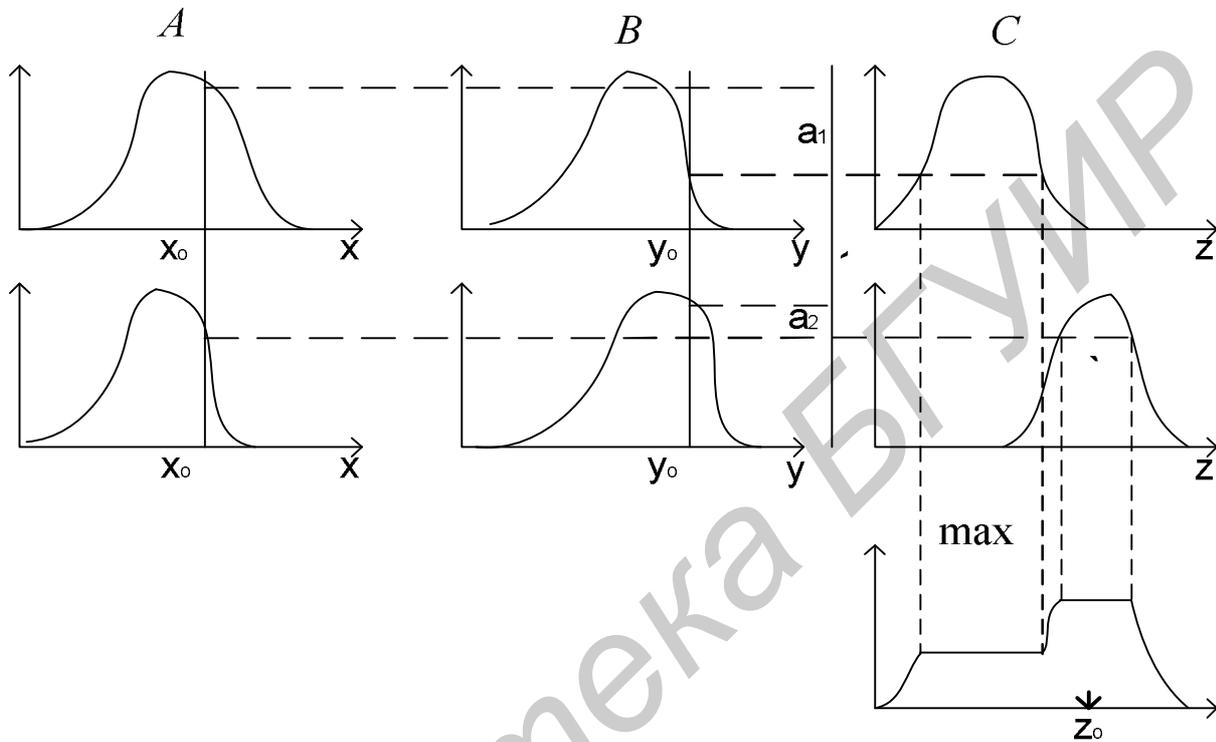


Рисунок 5.2 – Графическая интерпретация алгоритма Мамдани

Перечислим наиболее известные методы дефазификации.

1 Метод максимума – выбирается тот элемент нечеткого множества, который имеет наивысшую степень принадлежности этому множеству. Если этот элемент не является единственным, т.е. функция принадлежности  $m_A(y_1)$  имеет несколько локальных максимумов, или если имеется максимальное «плато», то выбор среди элементов, имеющих наивысшую степень принадлежности множеству, осуществляется на основе некоторого критерия.

2 Метод левого (правого) максимума – выбирается наименьшее (наибольшее) из чисел  $y_1, y_2, \dots, y_n$ , имеющих наивысшую степень принадлежности нечеткому множеству.

3 Метод среднего из максимумов – в качестве искомого четкого значения  $y_0$  принимается среднее арифметическое координат локальных максимумов

$$y_0 = \frac{1}{n} \sum_{i=1}^n y_i. \quad (5.11)$$

Возможные методы дефазификации и их обозначения в Matlab приведены на рисунке 5.3:

- 1) наименьший максимум (*smallest of max, som*);
- 2) наибольший максимум (*largest of max, lom*);
- 3) средний максимум (*mean of max, mom*);
- 4) бисекторный (*bisector of area*);
- 5) центроидный (*centroid*).

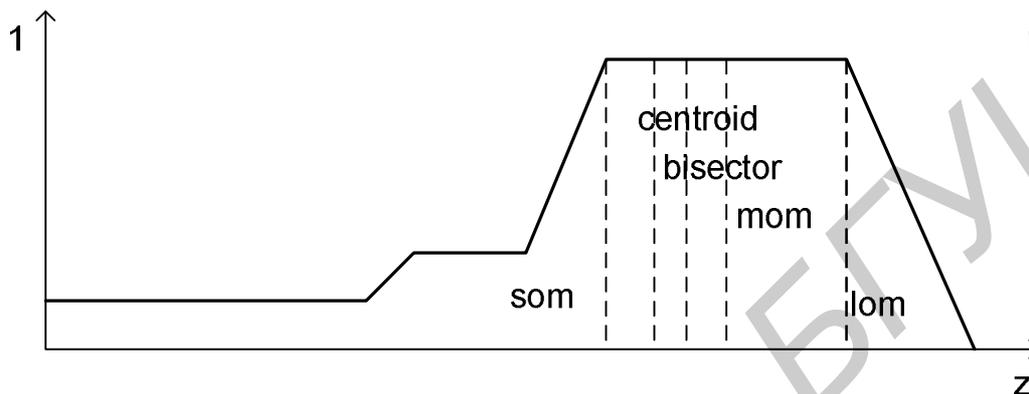


Рисунок 5.3 – Иллюстрация примеров фазификации  $z$

Приведем пример исследования автоматической системы управления с fuzzy-регулятором в среде Matlab.

Создадим в пакете Simulink файл модели, представленный на рисунке 5.4. Передаточная функция объекта управления описывается аperiodическим звеном с запаздыванием  $W_{oy} = \frac{k_{oy}}{T_{oy}s + 1} e^{-ts}$ , где  $\tau = 0,1$  с.

где  $k_{oy}$  – коэффициент усиления,  $T_{oy}$  – постоянная времени,  $t$  – время запаздывания.

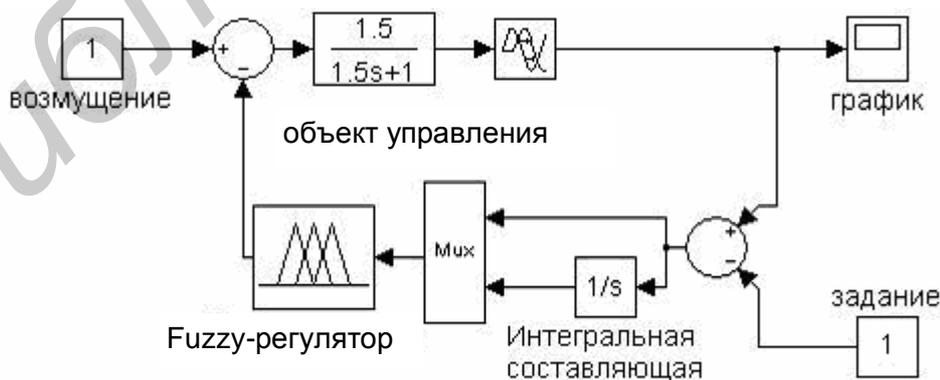


Рисунок 5.4 – Модель одноконтурной системы автоматического регулирования с ПИ-подобным fuzzy-регулятором

Теперь при помощи инструментов графического интерфейса пользователя (GUI) пакета «*Fuzzy Logic Toolbox*» создадим нечёткую систему типа Мамдани, реализующую типовой аналоговый ПИ-регулятор.

Командой *fuzzy* в окне Matlab вызываем окно Редактора фазиинференционной системы (*Fuzzy Inference System Editor*), выбираем тип системы – Мамдани, задаём два входа – для пропорциональной и интегральной составляющих и называем входные переменные, например,  $x_1$  и  $x_2$ , а выходную –  $y$ . Из данного окна вызываем окно Редактора функций принадлежности (*Membership Function Editor*) двойным щелчком мыши по изображению переменной  $x_1$  или при помощи меню *Edit*.

Здесь для лингвистического описания каждой переменной выберем семь треугольных термов (*NB, NM, NS, ZE, PS, PM, PB*). Для входных переменных регулятора рекомендуются симметричные диапазоны изменения, при этом

$$x_1 \in \left[ -\frac{1}{K_1}; \frac{1}{K_1} \right] \quad \text{и} \quad x_2 \in \left[ -\frac{1}{K_0}; \frac{1}{K_0} \right], \quad (5.12)$$

где  $K_1, K_0$  – оптимальные настройки пропорциональной и интегрирующей частей ПИ-регулятора в смысле какого-либо критерия.

Например, для переменной  $x_1$  лингвистическое описание приведено на рисунке 5.5.

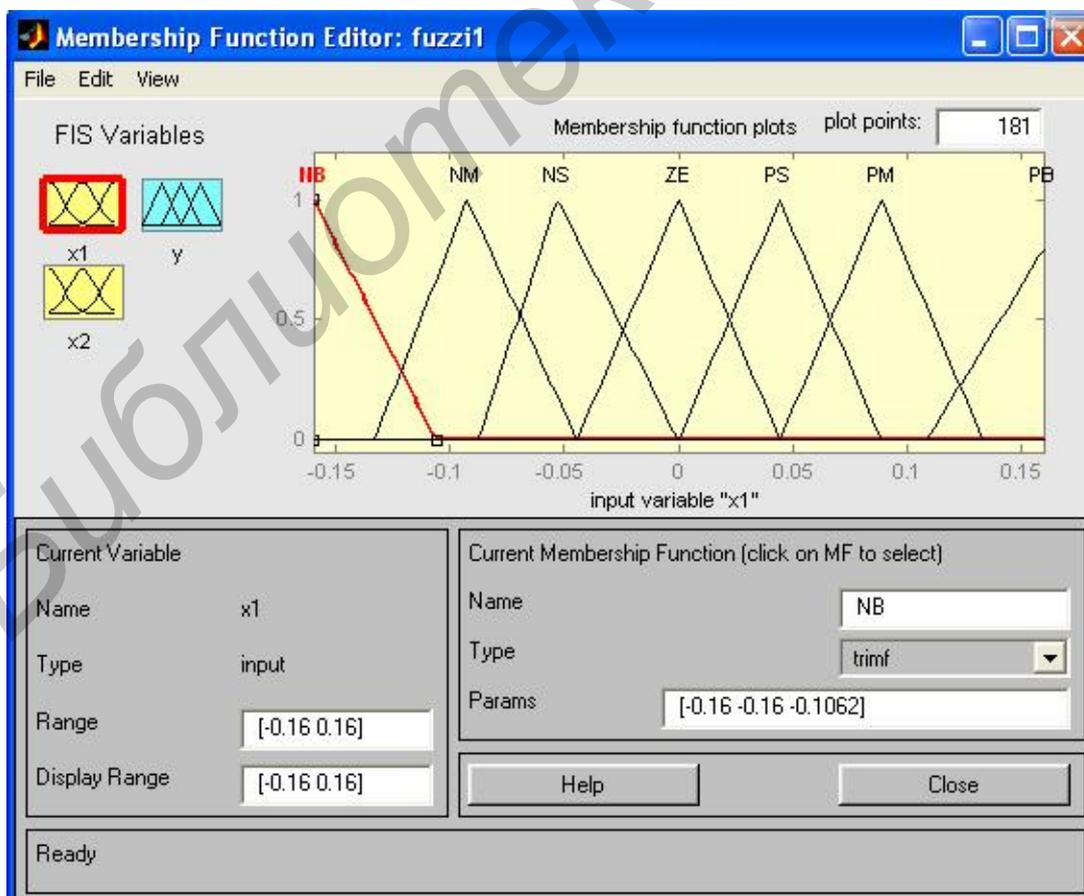


Рисунок 5.5 – Лингвистическое описание переменной  $x_1$

Термы выходной переменной лучше выбирать непересекающимися. Это повысит чёткость регулирования. Теперь необходимо сформировать базу правил fuzzy-регулятора. В основу положен способ, предложенный в литературе. Линейный непрерывный ПИ-регулятор с дифференциальным уравнением

$$y(t) = k_D \cdot e(t) + \frac{1}{T_I} \int_0^t e(t) \cdot dt \quad (5.13)$$

можно заменить близким по стратегии и логике управления fuzzy-регулятором, если в качестве его выходной переменной рассматривать приращение управляющего воздействия  $\Delta y$ . Тогда ПИ-закон регулирования можно представить в следующей дифференциальной форме (5.14):

$$\frac{dy(t)}{dt} = k_D \cdot \frac{de(t)}{dt} + \frac{1}{T_I} \cdot e(t) \quad (5.14)$$

или в разностной форме:

$$\Delta y(k) = y(k) - y(k-1) = k_D \cdot \Delta e(k) + \frac{\Delta t}{T_I} \cdot e(k). \quad (5.15)$$

Таким образом, для входных переменных  $\varepsilon(k)$  и  $\Delta \varepsilon(k)$  и выходной  $\Delta y(k)$  может быть синтезирован fuzzy-регулятор, реализующий нелинейный закон

$$\Delta y(k) = F[\Delta e(k), e(k)] \quad (5.16)$$

и эквивалентный в определённом смысле ПИ-регулятору.

Для нашего случая  $x_1$  соответствует сигналу рассогласования  $\varepsilon(k)$ ,  $x_2$  соответствует приращению сигнала рассогласования  $\Delta \varepsilon(k)$ , а  $y$  соответствует  $\Delta y(k)$ . Лингвистические правила для такого ПИ-подобного fuzzy-регулятора приведены в таблице 5.1.

Таблица 5.1 – Лингвистические правила для ПИ fuzzy-регулятора

$x_1 / x_2$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

Вызываем окно Редактора правил (*Rule Editor*) в меню Edit или нажатием Ctrl+3 и заполняем список правилами из таблицы 5.1. Правила формируются по типу: ЕСЛИ ... И ..., ТО.... (рисунок 5.6) Полученный файл сохраним под именем fuzzy1.fis.

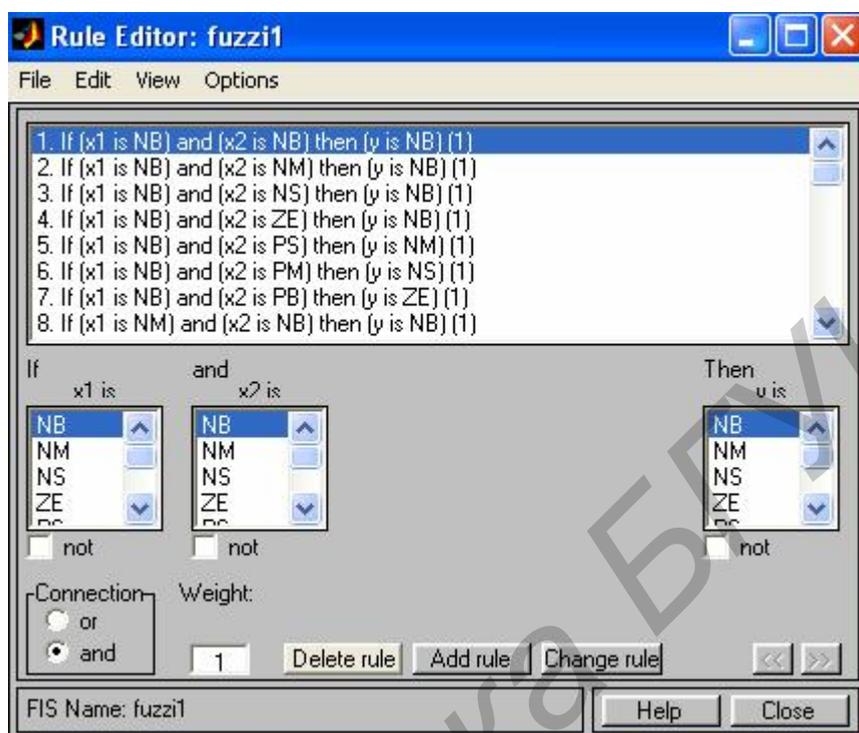


Рисунок 5.6 – Правила для нечеткого ПИ-регулятора

В окне параметров блока *Fuzzy Logic Controller* укажем имя файла fuzzy1. В окне модели в меню File выберем пункт *Model Properties*. В открывшемся окне выберем вкладку *Callbacks* и в поле *Model pre-load function* напишем: *fuzzy1=readfis('fuzzy1')*. Данная команда будет каждый раз при открытии файла модели помещать файл fuzzy1.fis в *Workspace* (рабочее пространство системы Matlab). Это необходимо для нормального функционирования модели.

Стоит заметить, что при внесении изменений в fis-файл нужно помещать его исправленную версию в *Workspace* либо при помощи пункта *Export/To Workspace* меню *File*, либо комбинацией клавиш Ctrl+T, либо каждый раз закрытием и открытием файла модели. В диалоговом окне *Simulation Parameters* меню *Simulation* во вкладке *Advanced* для опции *Boolean logic signals* необходимо установить значение *off*. При этом блоки логики будут допускать переменные в форме с плавающей точкой.

Работу fuzzy-регулятора можно сравнить с аналоговой системой регулирования (рисунок 5.7). Дифференциальное уравнение ПИД-регулятора имеет вид

$$y(t) = k_D \cdot e(t) + \frac{1}{T_I} \int_0^t e(t) \cdot dt + T_D \cdot \frac{de(t)}{dt}. \quad (5.17)$$

Для исследования ПИ-регулятора в блоке настроек регулятора необходимо в поле параметра  $T_D$  установить значение 0 (настройка дифференцирующей части ПИД-регулятора).

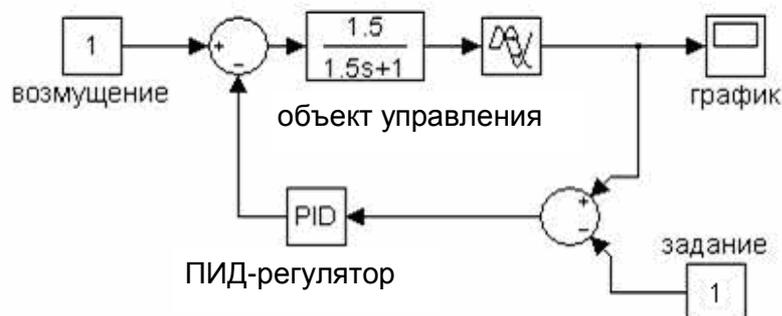


Рисунок 5.7 – Модель одноконтурной системы автоматического регулирования

## 5.2 Порядок и цель моделирования

Исследовать автоматическую систему регулирования с аналоговым ПИ-регулятором и fuzzy-регулятором. В качестве модели объекта принять апериодическое звено, параметры объекта принять из таблицы 5.3.

Для входных переменных регулятора рекомендуется задать симметричные диапазоны изменения  $x_1 \in \left[-\frac{1}{K_1}; \frac{1}{K_1}\right]$  и  $x_2 \in \left[-\frac{1}{K_0}; \frac{1}{K_0}\right]$ ,

где  $K_1, K_0$  – оптимальные настройки пропорциональной и интегрирующей частей ПИ-регулятора в смысле какого-либо критерия.

Расчет параметров приведен в таблице 5.2.

Таблица 5.2 – Настройки ПИ-регулятора

Регулятор	Апериодический процесс	Процесс с перерегулированием 20 %	Процесс с минимальным временем регулирования
ПИ	$K_1 = \frac{0,6 \cdot T}{K \cdot t}$ $K_0 = \frac{1}{K \cdot t}$	$K_1 = \frac{0,7 \cdot T}{K \cdot t}$ $K_0 = \frac{1}{K \cdot t}$	$K_1 = \frac{T}{K \cdot t}$ $K_0 = \frac{1}{K \cdot t}$

Для выходной переменной регулятора диапазон изменения рекомендуется брать в виде  $y \in [-0,5, C]$ , где верхняя граница  $C$  при единичном ступенчатом воздействии варьируется от 1,1 до 1,5, чтобы выходной сигнал регулятора мог компенсировать это возмущение.

По мере увеличения значения  $C$  уменьшается динамическая ошибка, но возрастают время регулирования и число колебаний переходного процесса. По-

этому рекомендуется  $C$  принимать равным 1,5, когда наблюдается оптимальное соотношение между величиной динамической ошибки, времени регулирования и количеством колебаний.

Цель моделирования заключается в сравнении динамики работы синтезированной САУ с аналоговым ПИ-регулятором и fuzzy-регулятором при различных значениях параметров синтезированного регулятора  $K_1, K_0$ .

### 5.3 Варианты заданий

Передаточная функция объекта управления описывается апериодическим звеном с запаздыванием  $W_{oy} = \frac{K}{Ts + 1} e^{-ts}$ , где  $\tau = 0,1$  с. Варианты заданий приведены в таблице 5.3

Таблица 5.3 – Параметры объекта управления

Вариант	1	2	3	4	5	6	7	8
$K$	1,5	1,8	1,3	1,9	2	2,2	1,4	2,5
$T$	1,8	1,8	1,5	1,5	1,3	1,5	1,5	1,6

Продолжение таблицы 5.3

9	10	11	12	13	14	15	16	17	18
2,3	2,4	3,5	3,8	3,3	3,9	3	3,2	3,4	4
1,4	1,7	1,8	1,5	1,5	1,5	1,8	1,2	1,4	1,9

### 5.4 Содержание отчета

- 1 Цель работы.
- 2 Исходные данные. Блок-схема моделируемой системы, графики переходных процессов для аналогового и нечеткого регуляторов.
- 3 Параметры элементов системы нечеткого вывода проектируемой системы (тип функций принадлежности, способ реализации логических выражений, способ дефазификации).
- 4 Анализ качества переходных процессов.
- 5 Выводы.

### 5.5 Контрольные вопросы

- 1 Что такое нечеткая логика? В каких приложениях используется?
- 2 Какие существуют операции над нечеткими множествами?
- 3 Способы задания нечеткого множества.
- 4 Функция принадлежности. Типы функций принадлежности.
- 5 Алгоритмы нечеткого вывода. Алгоритм Мамдани.

## Лабораторная работа №6

### АППРОКСИМАЦИЯ ФУНКЦИЙ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

**Цель** – изучить принципы функционирования нейронных сетей (НС) в рабочем режиме и режиме обучения.

#### 6.1 Краткие теоретические сведения

Нейронная сеть – совокупность нейронов, связанных между собой соответствующим образом. Нейрон – преобразовательный элемент, имеющий некоторое количество входов (синапсов), на которые поступают входные сигналы  $x_i$  и один выход (аксон), с которого снимается выходной сигнал  $y$ . Каждый синапс имеет вес  $w_i$ , на который умножается входной сигнал  $x_i$ .

Структура нейрона представлена на рисунке 6.1.

Внутри нейрона можно выделить блок суммирования, определяющий взвешенную сумму всех входных сигналов

$$U = \sum_{i=1}^n w_i \cdot x_i \quad (6.1)$$

и блок функции активации  $Y = F(U)$ .

Таким образом, нейрон функционирует за два такта:

- 1) суммирование входных сигналов;
- 2) вычисление  $Y$  по функции активации.

Функция активации должна удовлетворять двум условиям:

- 1)  $|F(U)| < 1$  при любом  $U$ ,
- 2) функция должна быть монотонной неубывающей.

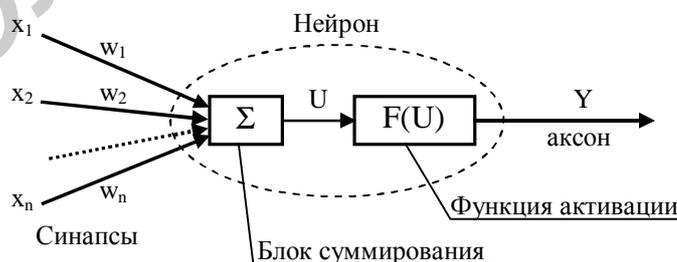


Рисунок 6.1 – Структура нейрона

Наиболее часто в качестве функций активации используются следующие функции:

1) ступенчатая функция

$$F(U) = \begin{cases} 1, & \text{если } U \geq a; \\ 0, & \text{если } U < a. \end{cases} \quad (6.2)$$

2) сигмоидная функция (рисунок 6.2, а)

$$F(U) = \frac{1}{1 + e^{-aU}}. \quad (6.3)$$

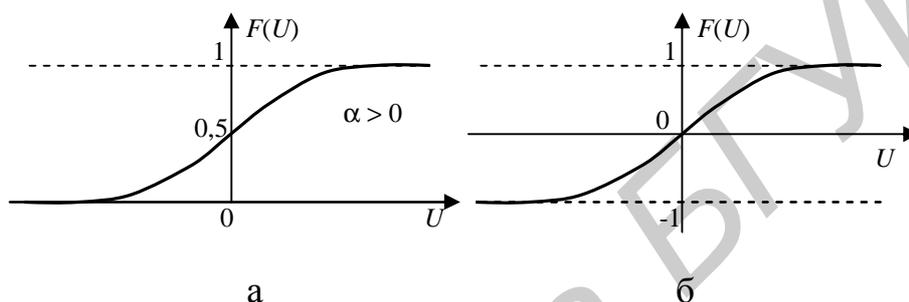


Рисунок 6.2 – Функции активации

3) гиперболический тангенс (рисунок 6.2, б)

$$F(U) = th(U) = \frac{e^U - e^{-U}}{e^U + e^{-U}}. \quad (6.4)$$

4) гладкие сжимающие функции

$$F(U) = \frac{U + Q}{|U + Q| + a}, \quad (6.5)$$

где  $Q$  – порог (смещение),

$a$  – параметр, определяющий крутизну статической характеристики нейрона.

Нейроны образуют нейронные сети путем соединения синапсов с аксонами. Наиболее распространенными и хорошо изученными являются трехслойные НС, состоящие из трех слоев нейронов: входного, скрытого и выходного (рисунок 6.3).

Нейроны входного слоя имеют только по одному синапсу. Количество нейронов входного слоя соответствует количеству входных переменных сети  $X$ .

Задачей нейронов этого слоя является только распределение входных сигналов по нейронам скрытого слоя, суммирования и вычисления функции активации в них не происходит. Количество нейронов в скрытом слое может быть различным и часто подбирается экспериментально. Недостаточное или избыточное количество нейронов в скрытом слое приводит к ухудшению

точности аппроксимации. Кроме того, избыточное количество усложняет сеть и уменьшает быстродействие.

Нейроны выходного слоя формируют выходные сигналы, их количество соответствует количеству выходов  $Y$ . Пример НС с 3 входными, 4 скрытыми и 2 выходными нейронами приведен на рисунке 6.3. Такая НС для краткости обозначается как (3-4-2).  $H_{ij}$  – нейроны.

Данные сети относятся к сетям прямого распространения, поскольку в них входные сигналы последовательно проходят через все нейроны и после преобразований напрямую подаются на выходы. Выходной сигнал  $y_{ij}$  каждого  $j$ -го нейрона в  $i$ -м слое определяется как

$$y_{ij} = F \left( \sum_{k=1}^{n(i-1)} w_{ij}^k \cdot y_{i-1,k} \right), \quad (6.6)$$

где  $n(i)$  – число нейронов в  $i$ -м слое.

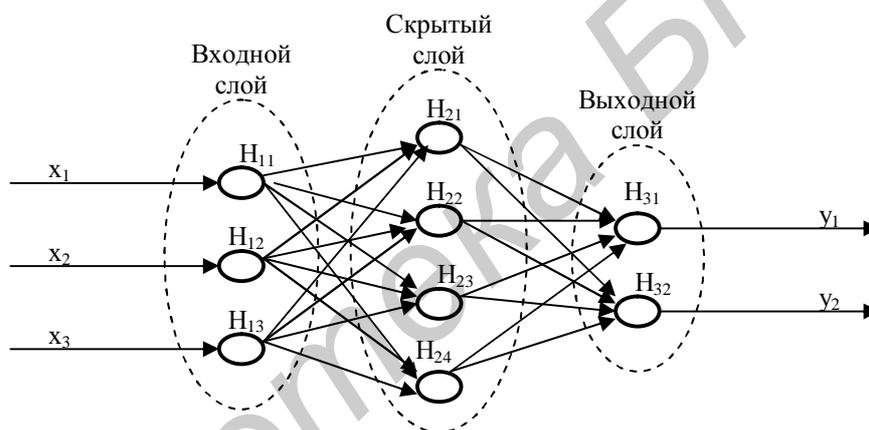


Рисунок 6.3 – Нейронная сеть вида (3-4-2)

Наиболее популярный класс многослойных сетей прямого распространения образуют многослойные перцептроны (MLP), в которых каждый вычислительный элемент использует пороговую или сигмоидальную функцию активации. Многослойный перцептрон может формировать сколь угодно сложные границы принятия решения и реализовывать произвольные булевы функции.

Сети, использующие радиальные базисные функции (RBF-сети), являются частным случаем двухслойной сети прямого распространения. Каждый элемент скрытого слоя использует в качестве активационной функции радиальную базисную функцию типа гауссовой. Радиальная базисная функция (функция ядра) центрируется в точке, которая определяется весовым вектором, связанным с нейроном. Каждый выходной элемент вычисляет линейную комбинацию этих радиальных базисных функций. С точки зрения задачи аппроксимации скрытые элементы формируют совокупность функций, которые

образуют базисную систему для представления входных примеров в построенном на ней пространстве.

Нейронные сети относятся к классу аппроксиматоров и «черных ящиков», аппроксимирующих некоторые функции вида

$$Y = F(X), \quad (6.7)$$

где  $Y$  – вектор выходных переменных,

$X$  – вектор входных.

Процесс аппроксимации заключается в подборе весовых коэффициентов  $w_{ij}$  и называется обучением НС. То есть НС может функционировать в двух режимах:

1) эксплуатации, когда на вход подаются сигналы, а на выходе снимаются результаты вычислений;

2) обучения, когда происходит корректировка весов таким образом, чтобы выходные сигналы наиболее точно соответствовали желаемым.

От качества обучения НС зависит точность ее работы в режиме эксплуатации.

Структура процесса обучения представлена на рисунке 6.4, где обозначены:  $Y_{жел}$  – желаемые значения выходных сигналов,  $E$  – ошибка обучения ( $E = Y_{жел} - Y$ ),  $K$  – корректирующие воздействия (обычно изменения весов  $Dw_{ij}$ ).



Рисунок 6.4 – Процесс обучения нейронной сети

Для обучения НС составляется обучающая выборка входных сигналов и соответствующих им выходных. Выборка может быть разделена на две части: рабочую выборку (на основе которой производится собственно обучение) и тестирующую выборку (для проверки качества обучения).

Далее определяется структура НС. Для трехслойной НС количество входных и выходных нейронов определяется по количеству входных и выходных переменных. Количество нейронов в скрытом слое  $N_c$  может быть взято из условия

$$N_c \leq \frac{(N_p - 1) \cdot N_{out}}{N_{in} + N_{out} + 1}, \quad (6.8)$$

где  $N_{in}$  и  $N_{out}$  – количество нейронов во входном и выходном слоях,

$N_p$  – количество обучающих примеров (объем выборки).

Весам синапсов необученной НС изначально присваиваются произвольные значения. Далее на вход НС подается первый вектор  $X$  из рабочей выборки, определяется вектор  $Y$  и ошибка обучения  $E$ . Исходя из значений вектора  $E$  корректируются веса синапсов. Затем подается следующий вектор  $X$  из выборки и т.д. Циклы обучения повторяются многократно, пока качество обучения не станет удовлетворительным (это можно проверить по тестирующей выборке).

Существует несколько методов обучения, которые можно классифицировать по способам использования учителя:

- обучение с учителем (коррекция весов производится исходя из сравнения текущего и желаемого выходных векторов);
- обучение с последовательным подкреплением знаний (сети не даются желаемые значения выходов, а ставится оценка «хорошо» или «плохо»);
- обучение без учителя (сеть сама вырабатывает правила обучения путем выделения особенностей из набора входных данных).

Для обучения RBF-сетей разработаны различные алгоритмы. Основной алгоритм использует двушаговую стратегию обучения, или смешанное обучение. Он оценивает позицию и ширину ядра с использованием алгоритма кластеризации «без учителя», а затем алгоритм минимизации среднеквадратической ошибки «с учителем» для определения весов связей между скрытым и выходным слоями. Поскольку выходные элементы линейны, применяется неитерационный алгоритм. После получения этого начального приближения используется градиентный спуск для уточнения параметров сети.

Этот смешанный алгоритм обучения RBF-сети сходится гораздо быстрее, чем алгоритм обратного распространения для обучения многослойных персептронов. Однако RBF-сеть часто содержит слишком большое число скрытых элементов. Это влечет более медленное функционирование RBF-сети, чем многослойного персептрона. Эффективность (ошибка в зависимости от размера сети) RBF-сети и многослойного персептрона зависят от решаемой задачи. Опишем правило обратного распространения

Для обучения обычно используется НС с функциями активации сигмоидного типа. Целью обучения по правилу обратного распространения является минимизация ошибки обучения, которая определяется как

$$E = \frac{1}{2} \sum_{i=1}^{N_{out}} (Y_i - Y_{жел i})^2. \quad (6.9)$$

Для уменьшения ошибки веса изменяются по правилу

$$w_{ij} = w_{ij} - \eta \frac{\partial E}{\partial w_{ij}}, \quad (6.10)$$

где  $\eta$  – константа, характеризующая скорость обучения.

Данная формула описывает процесс градиентного спуска в пространстве весов. Алгоритм обратного распространением состоит из следующих шагов.

Шаг 1. На вход НС подается вектор  $X$  из обучающей выборки и вычисляются выходы всех нейронов  $Y_{ij}$ .

Шаг 2. Определяется величина градиента ошибки  $EI$  для каждого нейрона выходного слоя:

$$EI_{Nj} = (Y_j - Y_{\text{жел } j}) \cdot Y_j \cdot (1 - Y_j), \quad (6.11)$$

где  $Y_j$  – выход  $j$ -го нейрона выходного слоя.

Шаг 3. Двигаясь от последнего слоя к первому, определяются градиенты  $EI_{ij}$  для каждого  $j$ -го нейрона каждого  $i$ -го слоя:

$$EI_{ij} = Y_{ij} \cdot (1 - Y_{ij}) \cdot \sum_{k=1}^{n(i+1)} EI_{i+1,k} \cdot w_{i+1,k}^j, \quad (6.12)$$

где  $k$  – номер синапса, соединяющего нейрон  $H_{ij}$  с нейроном  $H_{i+1,k}$  следующего слоя.

Шаг 4. Коррекция весов синапсов:

$$w_{ij}^k = w_{ij}^k - n \cdot EI_{ij} \cdot Y_{i-1,k}. \quad (6.13)$$

Коррекция весов для входного слоя не производится.

Шаг 5. Если обучающая выборка не закончилась, то шаги 1–5 повторяются.

Шаг 6. Определяется величина ошибки  $E$ . Если она не удовлетворительна, то шаги 1–6 повторяются.

Из описанного алгоритма видно, что процесс обучения НС включает два вложенных цикла обучения: внутренний цикл (шаги 1–5) повторяется соответственно количеству примеров из обучающей выборки, внешний (шаги 1–6) – до тех пор, пока не будет достигнуто удовлетворительное (с точки зрения ошибки) качество обучения.

После успешного обучения НС может быть протестирована на тестовой выборке. Если ошибка обучения на каждом примере из тестовой выборки удовлетворительна, то НС можно считать обученной и приступить к ее эксплуатации.

Приведем пример аппроксимации функции  $s=x_1+x_2$  с использованием программного пакета Matlab, в котором представлен графический интерфейс пользователя *NNTool*.

Чтобы запустить *NNTool*, необходимо выполнить одноимённую команду в командном окне Matlab, после этого появится главное окно *NNTool*, именуемое «Окном управления сетями и данными» (*Network/Data Manager*) (рисунок 6.5).

Панель «Сети и данные» (*Networks and Data*) имеет функциональные клавиши со следующими назначениями:

– помощь (*Help*);

- новые данные (*New Data*), позволяющего создавать новые наборы данных;
- новая сеть (*New Network*);
- импорт (*Import*), позволяющая импорт данных из рабочего пространства Matlab в пространство переменных *NNTool*;
- экспорт (*Export*), осуществляющая экспорт данных из пространства переменных *NNTool* в рабочее пространство Matlab;
- вид (*View*);
- удалить (*Delete*).

На панели «Только сети» (*Networks only*) расположены клавиши для работы исключительно с сетями. При выборе указателем мыши объекта любого другого типа эти кнопки становятся неактивными.

При работе с *NNTool* важно помнить, что клавиши *View*, *Delete*, *Initialize*, *Simulate*, *Train* и *Adapt* действуют применительно к тому объекту, который отмечен в данный момент выделением. Если такого объекта нет либо над выделенным объектом невозможно произвести указанное действие, соответствующая клавиша неактивна.

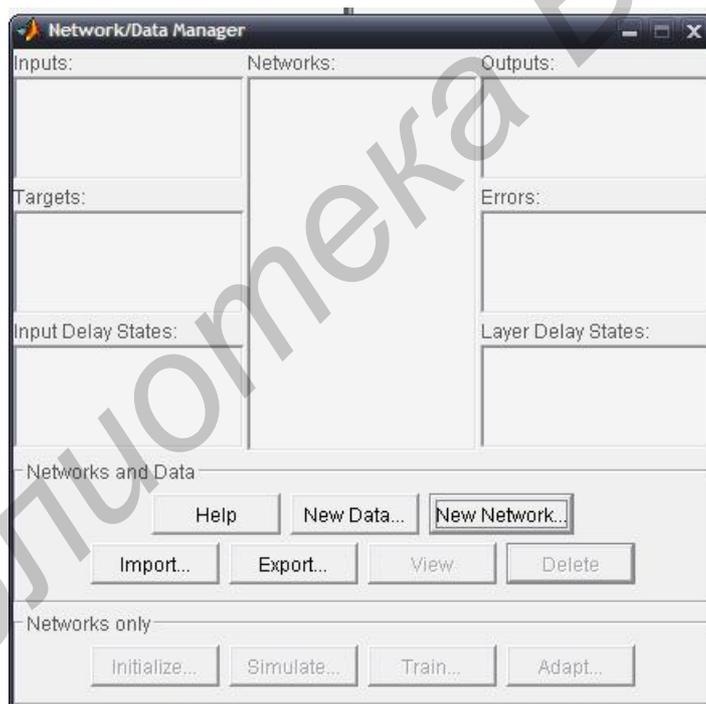


Рисунок 6.5 – Окно управления сетями и данными

Перед созданием сети необходимо заготовить набор обучающих и целевых данных,  $x_1$  и  $x_2$  – входы, а  $s$  – желаемый выход (таблица 6.1).

Таблица 6.1 – Целевые данные

$x_1$	0	1	2	3	4	5
$x_2$	0	1	2	3	4	5
$s$	0	2	4	6	8	10

Чтобы задать матрицу, состоящую из двух векторов – строк  $x=[x_1; x_2]$ , как входную, воспользуемся кнопкой *New Data*. В появившемся окне следует произвести изменения, и нажать клавишу *Создать (Create)*. После нажатия на *Create* в разделе *Targets* появится вектор  $s$  (рисунок 6.6).

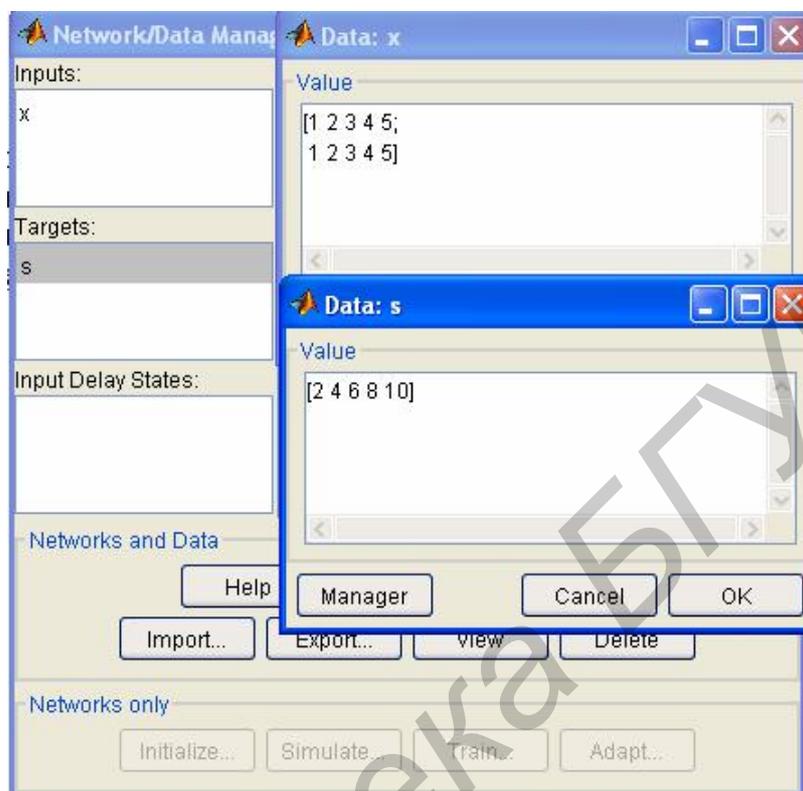


Рисунок 6.6 – Задание входных и целевого векторов

Данные в поле *Значение (Value)* могут быть представлены любым понятным для пакета выражением.

Теперь следует приступить к созданию нейронной сети. Нажимаем кнопку *New Network* и заполняем форму, как показано на рисунке 6.7. В данной работе была выбрана трехслойная нейронная сеть вида FFB (*Feed-forward backprop*) с числом нейронов в скрытом слое, равном 5. Функция активации сигмоидного вида.

При этом поля несут следующие смысловые нагрузки:

- имя сети (*Network Name*) – это имя объекта создаваемой сети,
- тип сети (*Network Type*) – определяет тип сети и в контексте выбранного типа представляет для ввода различные параметры в части окна,
  - входные диапазоны (*Input ranges*) – матрица с числом строк, равным числу входов сети, где каждая строка представляет собой вектор с двумя элементами: первый – минимальное значение сигнала, которое будет подано на соответствующий вход сети при обучении, второй – максимальное. Для упрощения ввода этих значений предусмотрен выпадающий список *Получить из входа (Get from input)*, позволяющий автоматически сформировать необходимые данные, указав имя входной переменной.

- количество нейронов (*Number of neurons*) в слое,
- передаточная функция (*Transfer function*) активации нейронов,
- функция обучения (*Learning function*), отвечающая за обновление весов и смещений сети в процессе обучения.

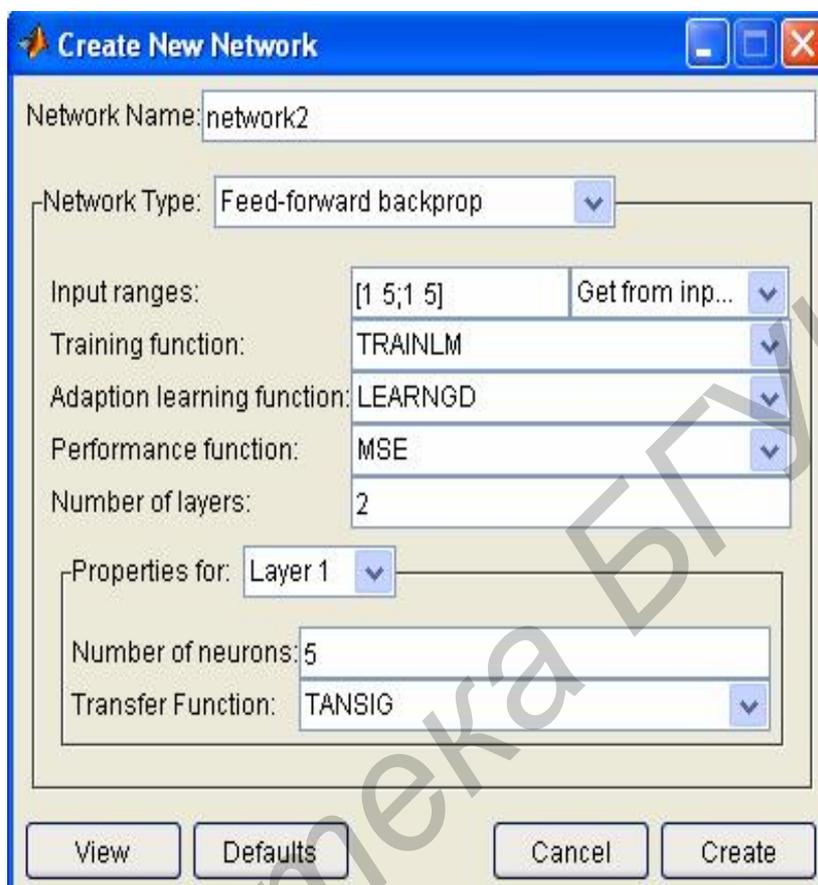


Рисунок 6.7 – Окно для создания сети

С помощью клавиши *View* (*View*) можно посмотреть архитектуру создаваемой сети (рисунок 6.8).

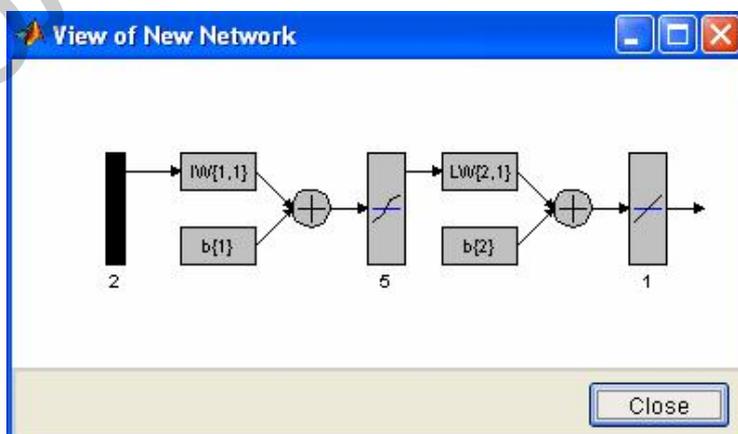


Рисунок 6.8 – Предварительный просмотр создаваемой сети

Для достижения цели сеть необходимо должным образом обучить, то есть подобрать подходящие значения параметров.

Вернёмся в главное окно *NNTool*. На данном этапе интерес представляет нижняя панель *Только сети (Networks only)*. Нажатие любой из клавиш на этой панели вызовет окно, на множестве вкладок которого представлены параметры сети, необходимые для её обучения и прогона, а также отражающие текущее состояние сети. Отметив указателем объект сети *network1*, вызовем окно управления сетью нажатием кнопки *Train*.

Перед нами возникнет вкладка *Train* окна свойств сети, содержащая, в свою очередь, ещё одну панель вкладок (рисунок 6.9).

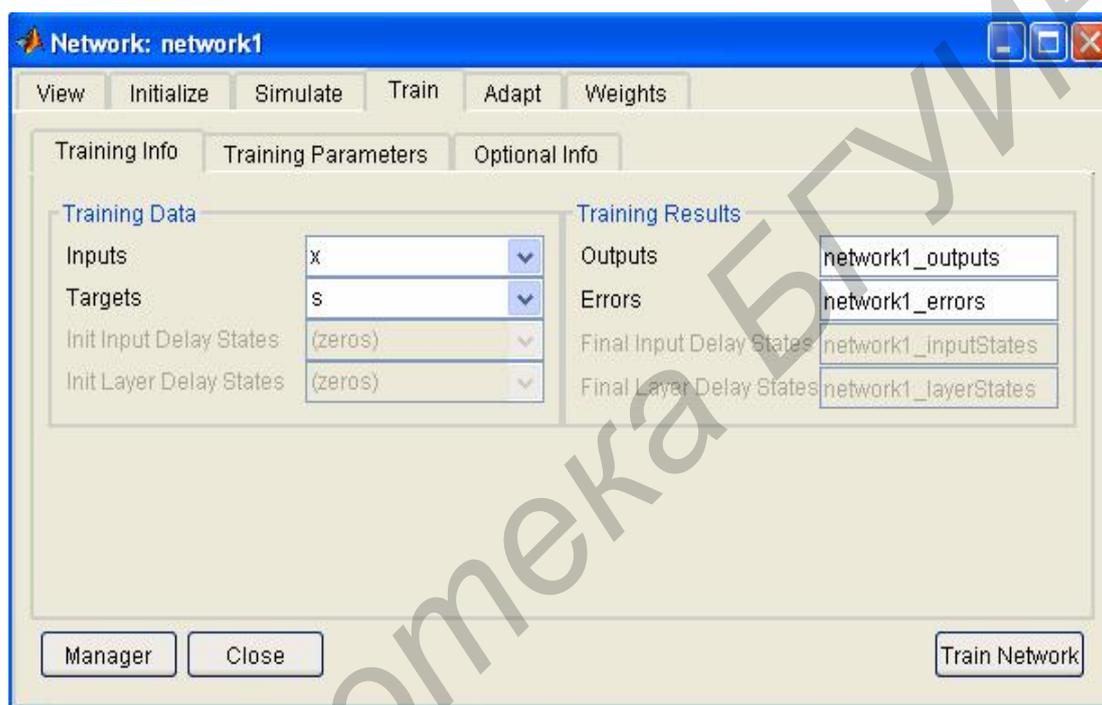


Рисунок 6.9 – Процесс обучения сети

Их главное назначение – управление процессом обучения. На вкладке *Информация обучения (Training info)* требуется указать набор обучающих данных в поле *Входы (Inputs)* и набор целевых данных в поле *Цели (Targets)*. Поля *Выходы (Outputs)* и *Ошибки (Errors)* *NNTool* заполняет автоматически. При этом результаты обучения, к которым относятся выходы и ошибки, будут сохраняться в переменных с указанными именами.

Завершить процесс обучения можно руководствуясь разными критериями. Возможны ситуации, когда предпочтительно остановить обучение, полагая достаточным некоторый интервал времени. С другой стороны, объективным критерием является уровень ошибки.

На вкладке *Параметры обучения (Training parameters)* для нашей сети можно установить следующие поля:

- количество эпох (*epochs*) – определяет число эпох (интервал времени), по прошествии которых обучение будет прекращено. Эпохой называют однократное представление всех обучающих входных данных на входы сети;

- достижение цели или попадание (*goal*) – здесь задаётся абсолютная величина функции ошибки, при которой цель будет считаться достигнутой;
- период обновления (*show*) графика кривой обучения, выраженный числом эпох;
- время обучения (*time*), по истечении которого обучение прекращается.

В методических целях убедимся в правильности решения задачи путём прогона обученной сети. Для этого необходимо открыть вкладку *Прогон (Simulate)* и выбрать в выпадающем списке *Входы (Inputs)* заготовленные данные.

В данной задаче естественно использовать тот же набор данных, что и при обучении  $x$ . При желании можно установить флажок *Задать цели (Supply Targets)*. Тогда в результате прогона дополнительно будут рассчитаны значения ошибки. Нажатие кнопки *Прогон сети (Simulate Network)* запишет результаты прогона в переменную, имя которой указано в поле *Выходы (Outputs)*.

Теперь можно вернуться в основное окно *NNTool* и, выделив мышью выходную переменную *network1*, нажать кнопку *Просмотр (View)*. Содержимое окна просмотра совпадает со значением вектора целей – сеть работает правильно.

При выборе нейронной сети для решения конкретной задачи трудно предсказать её порядок. Если выбрать неоправданно большой порядок, сеть может оказаться слишком гибкой и может представить простую зависимость сложным образом.

Это явление называется переобучением. В случае сети с недостаточным количеством нейронов, напротив, необходимый уровень ошибки никогда не будет достигнут. Здесь налицо чрезмерное обобщение.

## 6.2 Порядок и цель моделирования

Лабораторная работа выполняется с использованием программного пакета Matlab, в котором представлен графический интерфейс пользователя *NNTool*.

1 Необходимо с помощью нейронной сети провести аппроксимацию функции вида  $Y = f(X)$  на заданном интервале  $X$ , где  $X$  – вектор входных переменных  $X = \{x_1, x_2, \dots\}$ ,  $Y$  – вектор выходных переменных. Диапазоны изменения входных переменных указаны в таблице 6.1.

2 Провести обучение нейронной сети с помощью графического интерфейса. Рассмотреть варианты трехслойной сети вида FFB (*Feed-forward backprop*) с количеством нейронов в скрытом слое, равном 3, 5, 7, 10 и различными функциями активации (*logsig*, *purelin*, *tansig*). Число циклов обучения принять равным 100.

3 Осуществить проверку работоспособности сети с минимальной ошибкой обучения на тестовой выборке.

4 Построить графики аппроксимируемой функции при различных режимах обучения нейронной сети.

### 6.3 Варианты заданий

Варианты заданий для аппроксимации функций приведены в таблице 6.2

Таблица 6.2 – Варианты заданий

Вариант	Задание
1	$y = 2 \cdot \sin^2 x, x \in [0, 1]$
2	$y = 4 \frac{\cos x}{x}, x \in [-p, p]$
3	$y = 0.5 \cdot x^2 - 4.8 \cdot x + 3.5, x \in [1, 10]$
4	$y = \frac{x_1 - x_2}{x_1 + x_2}, x_1, x_2 \in [1, 10]$
5	$y =  x_1 - x_2 , x_1, x_2 \in [1, 10]$
6	$y = x_1 \cdot \sin(x_2), x_1 \in [1, 10], x_2 \in [-90^\circ, +90^\circ]$
7	$y = x_1 \cdot x_2 + x_3, x_1, x_2, x_3 \in [1, 10]$
8	$y = 1.5 \cdot x_1 +  x_2 - 2 \cdot x_3 , x_1, x_2, x_3 \in [1, 10]$
9	$y = \sqrt{x_1^2 + x_2^2}, x_1, x_2 \in [-10, 10]$
10	$y = 2.3 \cdot x_1 \cdot x_2 - 0.5 \cdot x_1^2 + 1.8 \cdot x_2, x_1, x_2 \in [1, 10]$
11	$y = x_1 \cos x_2, x_1 \in [0, 1], x_2 \in [0, \pi]$
12	$y = 2 \cdot x_1 + 5 \cdot x_1 \cdot x_2 + x_2; x_1, x_2 \in [-5, 5]$
13	$y = \sin x_1 \cdot \sin x_2 \cdot \sin x_3; x_1, x_2, x_3 \in [0, \pi]$
14	$y = 2 \cdot x_1 \cdot \cos x_2, x_1 \in [0, 1], x_2 \in [0, \pi]$
15	$y = x_1 + x_2 + x_3, x_i \in [0, 10]$

### 6.4 Содержание отчета

1 Цель работы.

2 Задание. Обучающая выборка.

3 Вид нейронной сети, параметры процесса обучения. График, иллюстрирующий изменение ошибки в процессе обучения.

4 Выводы о работоспособности сети с минимальной ошибкой обучения на основе проверки ее на тестовой выборке.

## 6.5 Контрольные вопросы

- 1 Что такое нейронная сеть и каковы ее основные свойства?
- 2 Какова структура нейрона?
- 3 Какие функции активации могут быть использованы в нейронных сетях?
- 4 Какие требования предъявляются к функциям активации?
- 5 Какие функции выполняет входной слой в многослойной сети?
- 6 В чем заключается обучение нейронных сетей?
- 7 Почему один из алгоритмов обучения получил название «алгоритм обратного распространения»?
- 8 Чем отличается обучение «с учителем» от обучения «без учителя»?

Библиотека БГУИР

## Литература

- 1 Горбачев, А. Д. Тексты лекций по курсу «Современные методы синтеза систем управления» / А. Д. Горбачев. – Минск : БГУИР, 1994.
- 2 Хлыпало, Е. И. Расчет и проектирование нелинейных корректирующих устройств в автоматических системах / Е. И. Хлыпало. – Л. : Энергоиздат, 1982.
- 3 Колесников, А. А. Последовательная оптимизация нелинейных агрегированных систем управления / А. А. Колесников. – М. : Энергоиздат, 1987.
- 4 Бойчук, Л. М. Метод структурного синтеза нелинейных систем автоматического управления / Л. М. Бойчук. – М. : Энергия, 1970.
- 5 Емельянов, С. В. Системы автоматического управления с переменной структурой / С. В. Емельянов. – М. : Наука, 1967.
- 6 Барбашин, Е. А. Функции Ляпунова / Е. А. Барбашин. – М. : Наука, 1970.
- 7 Заде, Л. А. Понятие лингвистической переменной и ее применение к принятию приближенных решений / Л. А. Заде. – М. : Мир, 1976.
- 8 Дьяконов, В. П. Математические пакеты расширения MATLAB : специальный справочник / В. П. Дьяконов, В. Н. Круглов. – СПб. : Питер, 2001.
- 9 Кофман, А. Введение в теорию нечетких множеств / А. Кофман. – М. : Радио и связь, 1982.
- 10 Розенблатт, Ф. Принципы нейродинамики. Перцептроны и теория механизмов мозга / Ф. Розенблатт. – М. : Мир, 1965.
- 11 Васильев, В. И. Интеллектуальные системы управления с использованием нейронных сетей : учеб. пособие / В. И. Васильев, Б. Г. Ильясов, С. С. Валеев. – Уфа : УГАТУ, 1997.
- 12 Зозуля, Ю. И. Интеллектуальные системы обработки информации на основе нейросетевых технологий : учеб. пособие / Ю. И. Зозуля. – Уфа : УГАТУ, 2000.

Учебное издание

**Снисаренко Светлана Валерьевна**  
**Стасевич Наталья Александровна**

## ***СОВРЕМЕННЫЕ МЕТОДЫ СИНТЕЗА СИСТЕМ УПРАВЛЕНИЯ***

**Лабораторный практикум**  
для студентов специальности I-53 01 07  
«Информационные технологии и управление в технических системах»  
всех форм обучения

Редактор Т. П. Андрейченко  
Корректор М. В. Тезина  
Компьютерная верстка Е. Г. Бабичева

---

Подписано в печать 21.04.2008.  
Гарнитура «Таймс».  
Уч.-изд. л. 3,0.

Формат 60x84 1/16.  
Печать ризографическая.  
Тираж 120 экз.

Бумага офсетная.  
Усл. печ. л. 3,6.  
Заказ 619.

---

Издатель и полиграфическое исполнение: Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники»  
ЛИ №02330/0056964 от 01.04.2004. ЛП №02330/0131666 от 30.04.2004.  
220013, Минск, П. Бровки, 6