

Министерство образования и науки Российской Федерации
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Энергетический факультет

А.Н. Рыбалев, Д. А. Теличенко, В.Ю. Косицын

СОВРЕМЕННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

*ПОСОБИЕ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ,
КУРСОВОМУ И ДИПЛОМНОМУ ПРОЕКТИРОВАНИЮ*

Благовещенск

2010

ББК 32.965я73

Т 31

*Печатается по решению
редакционно-издательского совета
энергетического факультета
Амурского государственного
университета*

Рыбалев А.Н., Теличенко Д. А., Косицын В.Ю.

«Современные системы управления». Пособие к выполнению лабораторных работ, курсовому и дипломному проектированию» – Благовещенск: Амурский гос. ун-т, 2011. – 101 с.

Предназначено для студентов специальностей 220301 «Автоматизация технологических процессов и производств» и др. изучающих дисциплины «Современные системы управления», «Системы автоматического управления» и выполняющих по данным дисциплинам лабораторные работы. Пособие может быть использовано также при выполнении курсовых и дипломных работ, связанных с расчетами и моделированием систем автоматического управления.

Рецензенты: докт. техн. наук, проф. Е.Л. Еремин,
докт. техн. наук, проф. В.В. Воронин

© Амурский государственный университет, 2011

ВВЕДЕНИЕ

Учебное пособие предназначено для студентов специальностей «Автоматизация технологических процессов и производств тепловых электрических станций». Основной упор сделан на рассмотрение систем регулирования, применяющихся в теплоэнергетике.

Методическое пособие состоит из восьми лабораторных работ. В *первой* работе производится процедура получения математического описания объекта или системы управления, выполняется их структуризация. Во второй работе студенты знакомятся с классическими законами управления тепловыми объектами. Третья и четвертая работы представляют собой пример применения адаптивных систем в теплоэнергетике. Пятая работа посвящена системам, работающим на основе нечеткой логики. Шестая работа предлагает реализовать заданную систему регулирования на языке микроконтроллера, предварительно выполнив дискретизацию системы и синтезировав принципиальную электрическую схему. Седьмая работа направлена на получение исследуемой модели системы, максимально приближенной к реальным условиям. В восьмой работе студентам предлагается оснастить одну из полученных систем управления пользовательским графическим интерфейсом.

Результаты, полученные в первой лабораторной работе, служат исходными данными для выполнения последующих работ.

ЛАБОРАТОРНАЯ РАБОТА №1

МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ ОБЪЕКТОВ УПРАВЛЕНИЯ

Цель работы:

1. Получение математического описания объекта управления по кривой разгона.
2. Изучение формы математического представления объектов управления.

Краткие теоретические сведения

Экспериментальный метод нахождения уравнений динамики объекта, как правило, заключается в подборе дифференциальных уравнений, решение которых наилучшим образом совпадают с экспериментально полученными реакциями объекта на известные изменения входных координат при заданных начальных условиях. Известны и другие аналитические зависимости для описания динамики объекта [13].

Все экспериментальные методы нахождения уравнений динамики путем аппроксимации экспериментальных данных базируются на предположениях о сосредоточенности параметров объекта, стационарности во времени его динамических свойств и линейности их при малых изменениях входных координат.

Практическое применение экспериментального метода для определения динамических характеристик объектов связано, во-первых, с получением кривых разгона, во-вторых, в нахождении какой-либо математической зависимости для их описания.

Кривая разгона (рис. 1.1) – это реакция объекта на ступенчатое (в классической теории – единичное) воздействие при нулевых начальных условиях.

Как правило, основная задача после снятия разгонной кривой, состоит в выборе *метода* получения математической модели. Математическая модель должна быть адекватной, достаточно точно отображать динамические процессы в системе при относительно простой структуре, а также пригодной для исполь-

зования ее в последующих математических расчетах, например, при синтезе регулятора.

Существует большое количество методов получения математического описания по кривой разгона. Для сравнения далее приводятся наиболее распространенных на практике методы, а именно метод последовательного логарифмирования и метод оценки параметров модели непосредственно по кривой разгона [13].

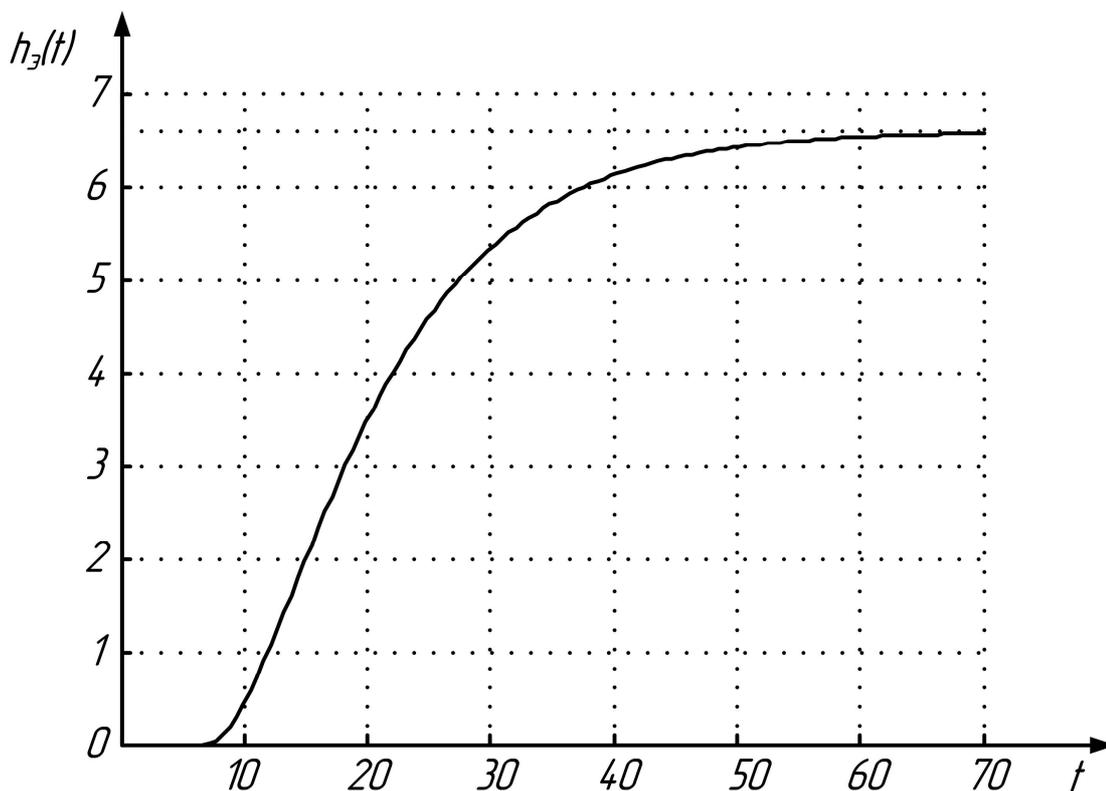


Рис. 1.1 Экспериментальная кривая разгона

1. Метод последовательного логарифмирования

Применяется для линейных и близким к линейным (линеаризованным) объектам и базируется на том, что решение обыкновенного линейного дифференциального уравнения с постоянными коэффициентами может быть представлено в виде суммы слагаемых, которые зависят от того, какие корни имеет характеристическое уравнение.

Перед выполнением аппроксимации кривую разгона необходимо представить в табличной форме, то есть по каждому значению входной величины определить значение выходной. Данное действие рекомендуется выполнять с

постоянным шагом. Если в системе присутствует транспортное запаздывание τ , то на этом этапе оно отбрасывается.

Решение задачи состоит в определении по кривой разгона порядка n и коэффициентов дифференциального уравнения

$$a_n \cdot y^{(n)}(t) + a_{n-1} \cdot y^{(n-1)}(t) + \dots + a_1 \cdot \dot{y}(t) + y(t) = k \cdot x(t) \quad (1.1)$$

На первом этапе осуществляется нормирование переходной характеристики

$$h(t) = \frac{h_s(t)}{h_s(\infty)}, \quad (1.2)$$

где $h_s(t)$ - исходная кривая, $h(t)$ - нормированная кривая.

Далее, в случае с простыми вещественными некрратными корнями, решением дифференциального уравнения (1.1) будет

$$h(t) = 1 - \sum_{k=1}^n H_k e^{-p_k t}, \quad (1.3)$$

где H_k - постоянные коэффициенты, p_k - корни характеристического уравнения, удовлетворяющие требованию

$$\frac{|p_{k+1}|}{|p_k|} \geq 2 - 3. \quad (1.4)$$

На постоянные коэффициенты накладываются следующие ограничения

$$\begin{cases} 1 - H_1 - \dots - H_n = 0 \\ p_1 \cdot H_1 + p_2 \cdot H_2 + \dots + p_n H_n = 0 \\ p_1^{n-1} \cdot H_1 + p_2^{n-1} \cdot H_2 + \dots + p_n^{n-1} H_n = 0 \end{cases} \quad (1.5)$$

С увеличением времени t слагаемые $H_k e^{-p_k t}$ асимптотически стремятся к нулю. Наиболее медленно это происходит у первого слагаемого, корень которого является наименьшим по модулю. Благодаря этому, начиная с некоторого момента времени t_1 , становится справедливым равенство:

$$h_1(t) \approx 1 - h(t) = H_1 e^{-p_1 t} \quad (1.6)$$

Логарифмируя выражение (1.6), получаем

$$\ln|h_1(t)| = \ln H_1 - p_1 \cdot t \quad (1.7)$$

Для построения данной функции удобно воспользоваться логарифмической шкалой. На оси абсцисс в линейном масштабе откладывается время, а на оси ординат в логарифмическом масштабе $\ln|h_1(t)| = \ln|1-h(t)|$. В силу описанных свойств с некоторого момента времени $t = t_1$ кривая $\ln|h_1(t)|$ будет мало отличаться от собственной асимптоты – прямой $\ln H_1 - p_1 \cdot t$. Отрезок, отсекаемый данной асимптотой на оси ординат, равен $\ln H_1$, из чего легко определить значение H_1 . Тангенс угла наклона асимптоты к оси абсцисс равен p_1 . (рис. 1.2). Знак коэффициента H_1 определяется по знаку функции $h_1(t)$ при $t \rightarrow \infty$. Для крив

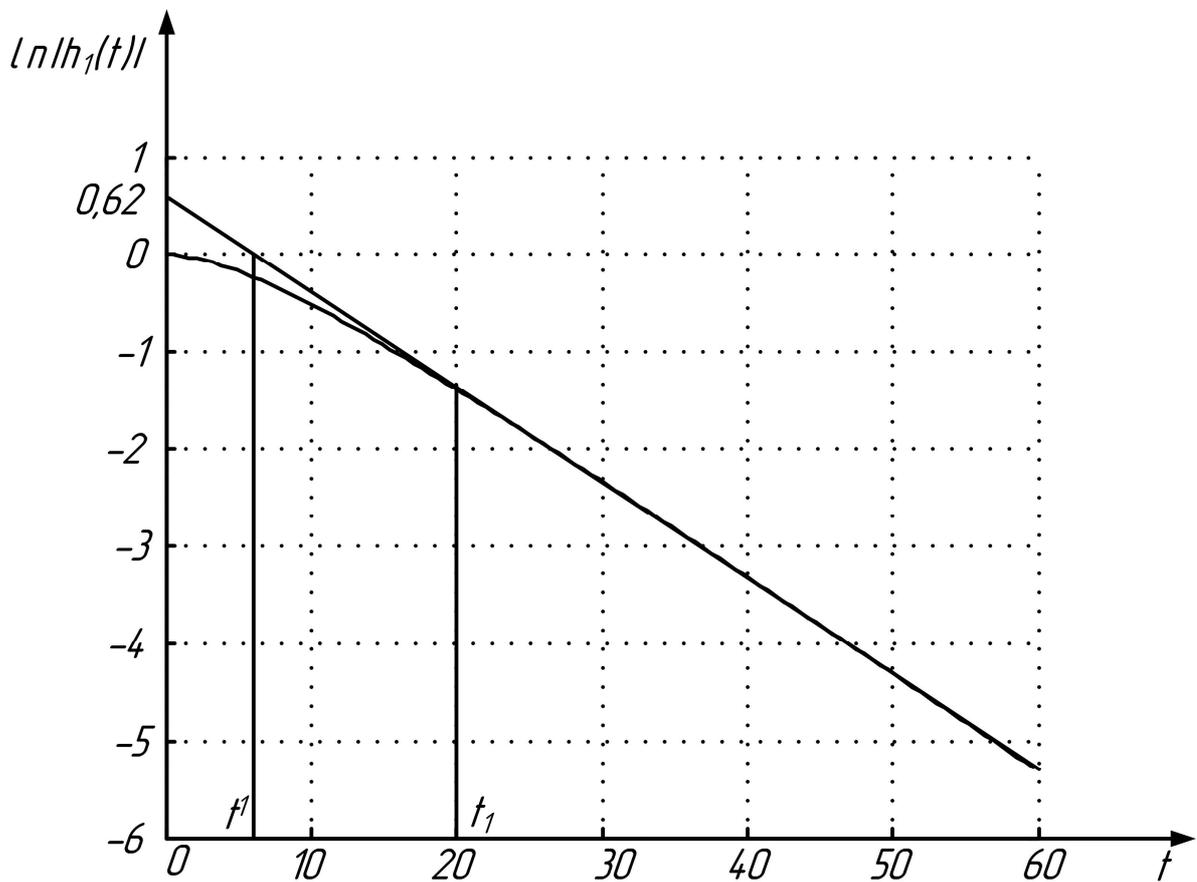


Рис. 1.2 – К определению первого слагаемого (1.3)

$$t = 20 \dots 60$$

$$\ln H_1 = 0,62$$

$$H_1 = 1,86, t' = 6,3$$

(8)

$$p_1 = \frac{\ln H_1}{t'} = 0,0986$$

Для определения следующего по величине вещественного корня вычисляют функцию

$$h_2(t) = h_1(t) - H_2 e^{-p_2 \cdot t} \quad (1.9)$$

Логарифмируя данное выражение и строя асимптоту к полученной кривой имеем:

$$\ln|h_2(t)| \approx \ln H_2 - p_2 \cdot t \quad (1.10)$$

Отрезок, отсекаемый данной прямой на оси ординат, равен $\ln H_2$, что позволяет определить значение H_2 . Тангенс угла наклона равен p_2 (рис. 1.3). Знак коэффициента H_2 определяется по знаку $h_2(t)$ при $t \rightarrow \infty$. Следует иметь ввиду, что знаки коэффициентов H_i чередуются.

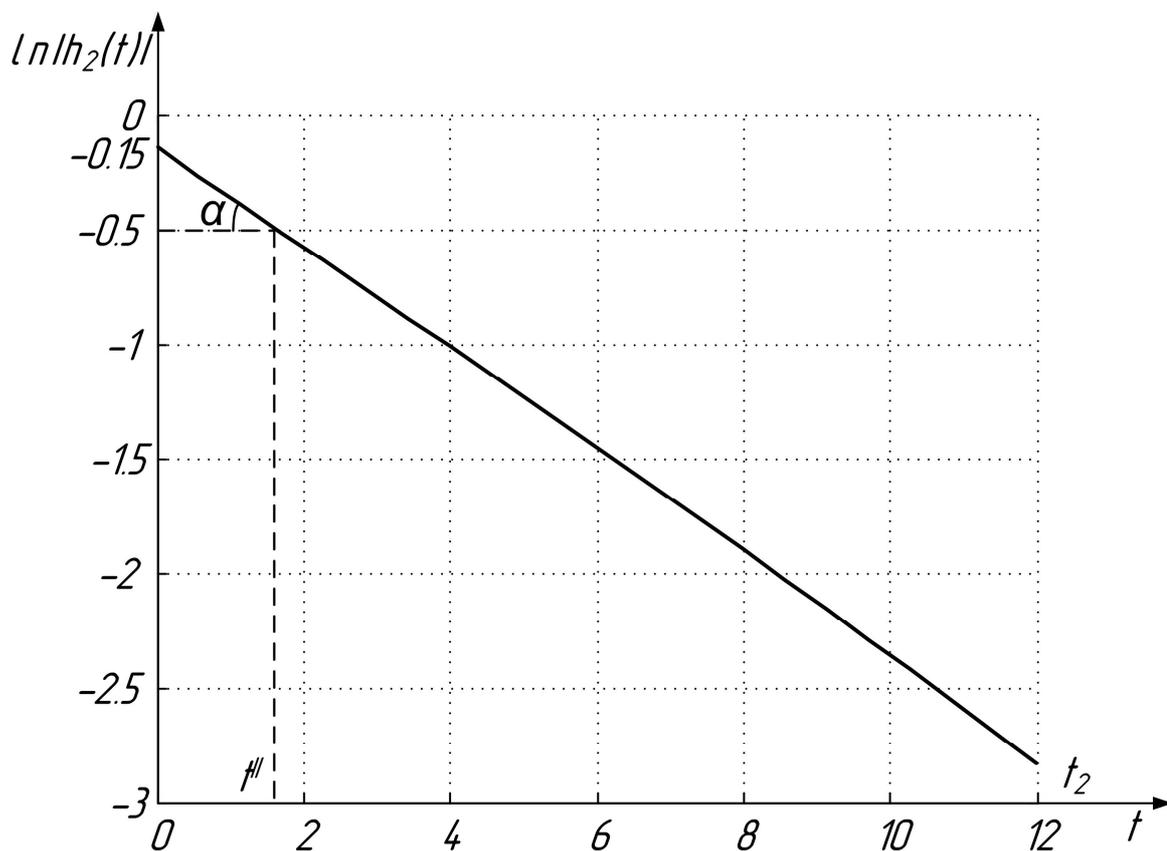


Рис. 3 – К определению второго слагаемого (1.3)

$$t = 0 \dots 12$$

$$\ln H_2 = 0,15$$

$$H_2 = 0,86, t'' = 1,66 \quad (1.11)$$

$$p_2 = \operatorname{tg} \alpha = \frac{0,5 - 0,13}{1,66} = \frac{0,37}{1,66} = 0,2231$$

Аналогичным образом определяются значения остальных корней характеристического уравнения и постоянных коэффициентов. Процесс аппроксимации останавливается, когда $h_n(t) \approx 0$ с точностью 1-2 % от $h(\infty)$. Практика показывает, что $h(t)$ можно аппроксимировать суммой из 2-4 экспонент.

Передаточная функция объекта находится при условии $T_k = 1/p_k$:

$$W(p) = \frac{1}{\prod_{k=1}^n (T_k p + 1)} \quad (1.12)$$

При выполнении условия (1.5) для рассмотренного примера, решение дифференциального уравнения (1.2) имеет вид

$$h_y(t) = k \cdot (1 - 1,86 \cdot e^{-0,0986t} + 0,86 \cdot e^{-0,2231t}) \quad (1.13)$$

Постоянные времени переходного процесса (1.13)

$$T_1 = 1/p_1 = 1/0,0986 \approx 10$$

$$T_2 = 1/p_2 = 1/0,2231 \approx 4,5$$

Передаточная функция соответственно

$$W(p) = \frac{k}{(10p + 1)(4,5p + 1)} \quad (1.14)$$

На исходной кривой разгона (рис. 1.1) присутствует запаздывание, поэтому его следует учесть следующим образом:

$$W(p) = \frac{k}{(10p + 1)(4,5p + 1)} \cdot e^{-\tau \cdot p} \quad (1.15)$$

2. Метод оценки параметров по кривой разгона

Рассмотрим нормированную передаточную функцию вида со структурой, изображенной на рис. 1.3

$$W(p) = \frac{1}{Tp + e^{-\lambda p}} \cdot e^{-\tau \cdot p}, \quad (1.16)$$

где T , λ , τ - параметры экспериментальной передаточной характеристики, значение которых определяется по рис. 1.5 (исходная кривая разгона на рис. 1.1).

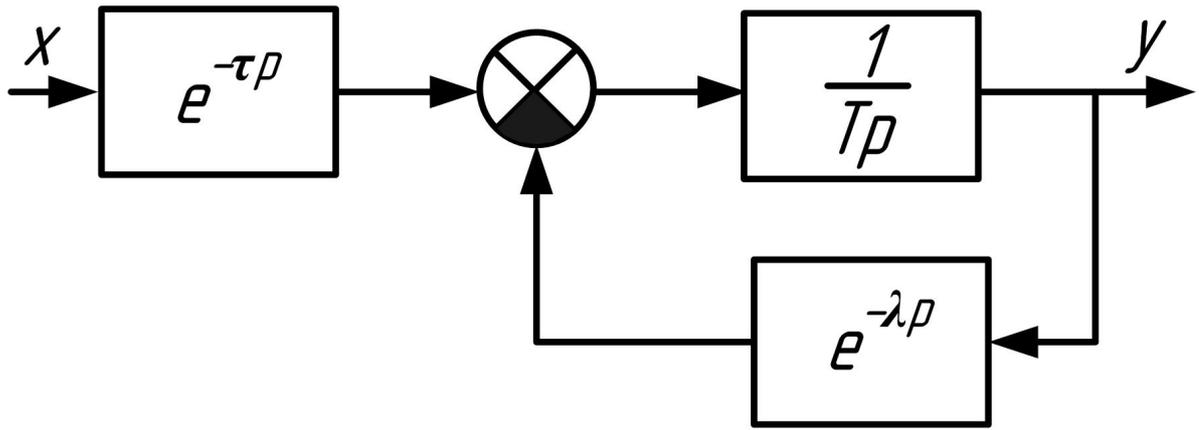


Рис. 4 Структура модели (1.15)

Параметр λ при фиксированных T и τ зависит от площади S , ограниченной осью ординат, прямой $y = 1$ и переходной характеристикой $h(t)$ и равен

$$\lambda = T + \tau - S \quad (1.17)$$

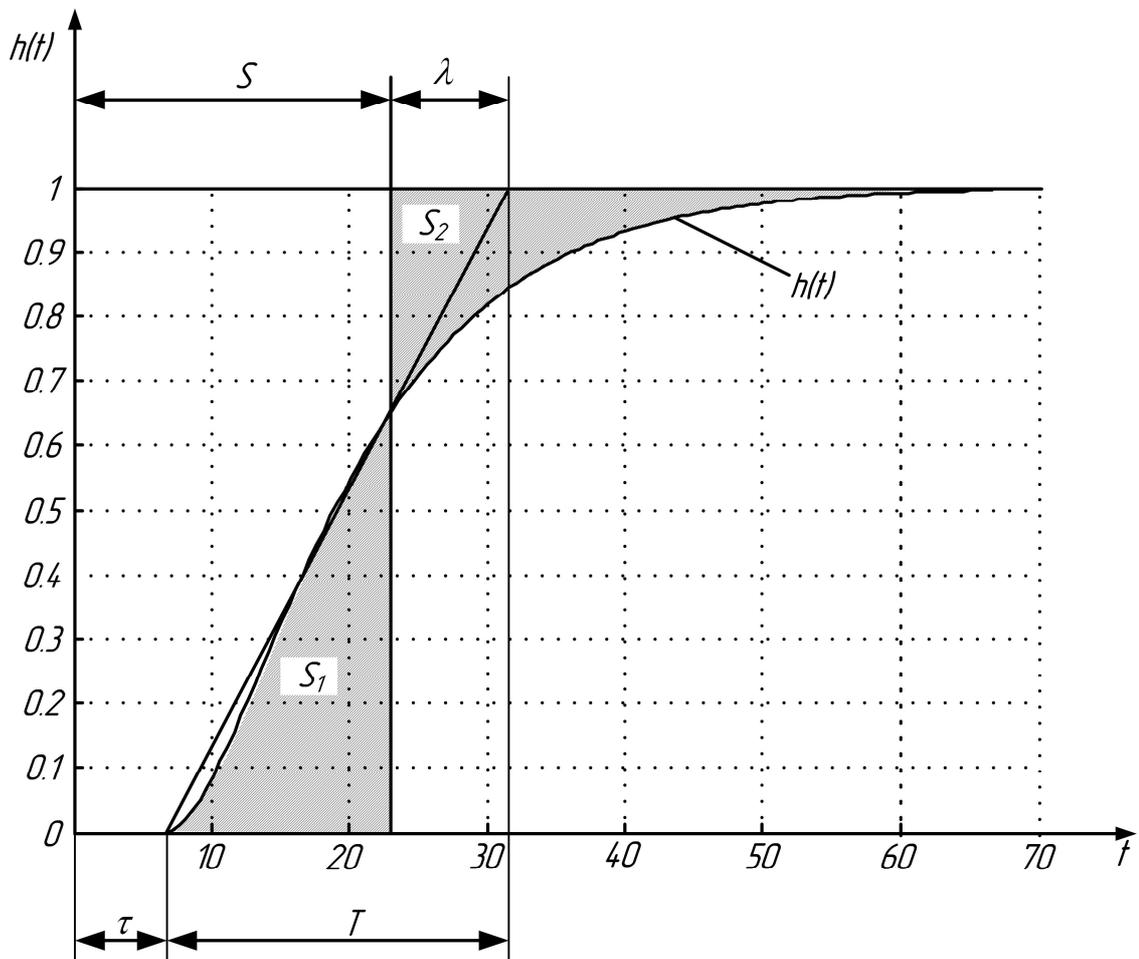


Рис. 1.5 – К методу определения параметров объекта по кривой разгона

Параметр S можно также определить как отрезок времени от начала нанесения возмущения до ординаты, отсекающей на переходной характеристике $h(t)$ равные площади S_1 и S_2 .

По рис. 1.5 неизвестные параметры передаточной функции (1.16):

$$\tau = 7$$

$$T = 31 - 7 = 24$$

$$S = 15,5$$

$$\lambda = T + \tau - S = 24 + 7 - 22,5 = 8,5$$

$$W(p) = \frac{1}{23p + e^{-8,5p}} \cdot e^{-7p}$$

(1.18)

Переходные характеристики экспериментальной кривой и кривой (1.18) показаны на рис. 1.6.

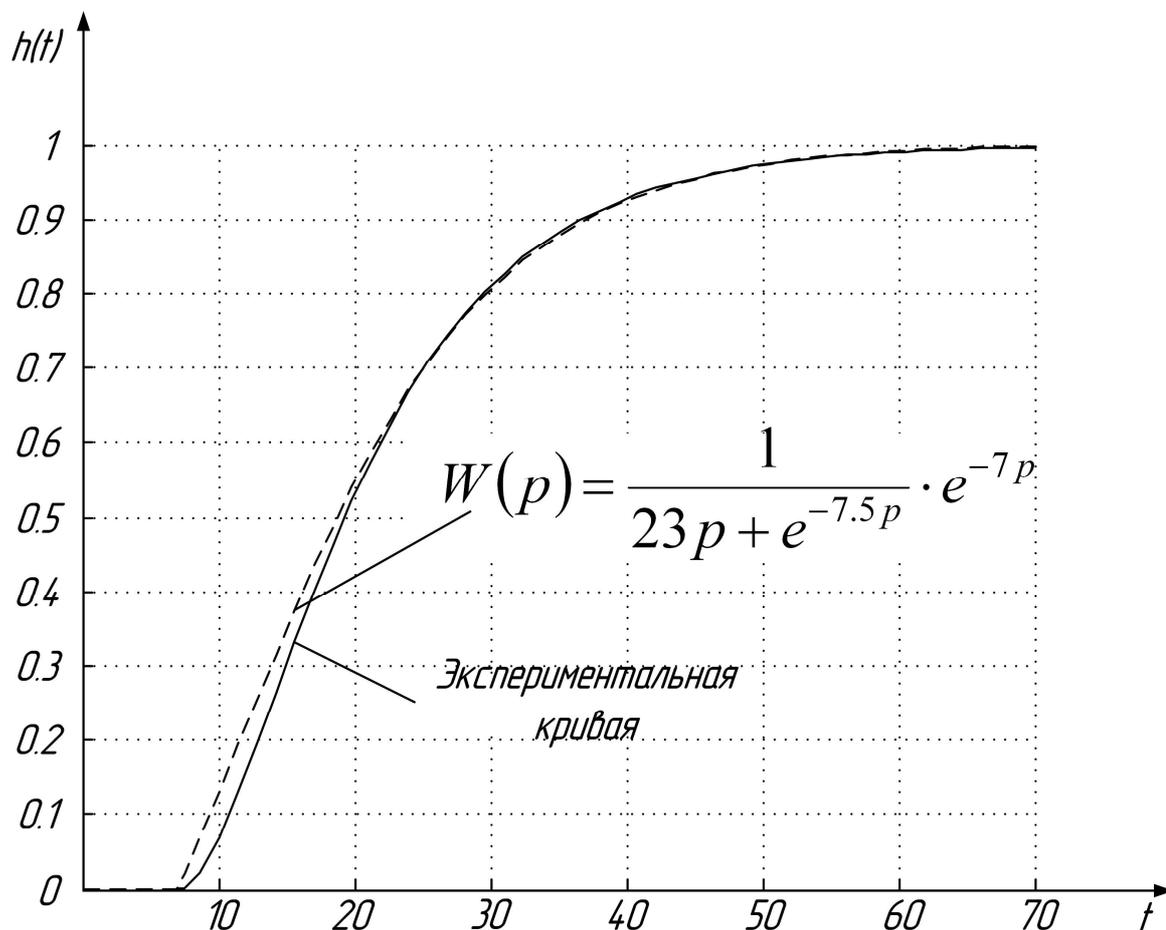


Рис. 1.6 Экспериментальная и аналитическая кривые разгона

Таким образом, модель (1.14) достаточно точно отражает динамику экспериментальной кривой.

3. Представление объекта управления в пространстве состояний

Линейный объект описывается в пространстве состояний уравнениями:

$$\begin{aligned} \dot{x}(t) &= A(t) \cdot x(t) + B(t) \cdot u(t) \\ y(t) &= C(t) \cdot x(t) + D(t) \cdot u(t) \end{aligned} \quad (1.19)$$

где $x(t) \in R^n$, $y(t) \in R^l$, $u(t) \in R^m$, а матрицы – функции $A(t)$, $B(t)$, $C(t)$, $D(t)$ имеют размеры $n \times n$, $n \times m$, $l \times n$, $l \times m$ соответственно.

Если $D(t) \equiv 0$, то объект (1.19) называется собственным (строго реализуемым), в противном случае несобственным.

Если матрицы $A(t)$, $B(t)$, $C(t)$, $D(t)$ постоянны (не зависят от времени t), то объект (1.19) называется стационарным, в противном случае нестационарным.

Для стационарных собственных систем уравнения в пространстве состояний записываются следующим образом

$$\begin{aligned} \dot{x}(t) &= A \cdot x(t) + B \cdot u(t) \\ y(t) &= C \cdot x(t) \end{aligned} \quad (1.20)$$

По уравнениям (1.20) записывается передаточная функция

$$W(p) = \frac{y(t)}{u(t)} = C(pI_n - A)^{-1} B = \frac{C(pI_n - A)^+ B}{\det(pI_n - A)}, \quad (1.21)$$

где через $(pI_n - A)^+$ - обозначена присоединенная к $(pI_n - A)$ матрица, I_n - единичная диагональная матрица размером $n \times n$.

4. Методы преобразования передаточных функций в уравнения в пространстве состояний

Для преобразования передаточной функции объекта управления с одним входом и скалярным выходом, например, (14), удобно использовать каноническое представление вида

$$W(p) = \frac{c_n p^{n-1} + c_{n-1} p^{n-2} + \dots + c_2 p + c_1}{p^n + a_n p^{n-1} + \dots + a_2 p + a_1}, \quad (1.22)$$

где $c_1 \dots c_n$ - коэффициенты числителя передаточной функции, $a_1 \dots a_n$ - коэффициенты знаменателя передаточной функции.

Матрицы уравнений в пространстве состояний (1.20), соответствующие передаточной функции (1.22) имеют вид

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -a_1 & -a_2 & -a_3 & \dots & -a_{n-1} & -a_n \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{pmatrix}, \quad (1.23)$$

$$C = (c_1 \quad c_2 \quad \dots \quad c_{n-1} \quad c_n), \quad (1.24)$$

Для получения канонического вида (1.22), исходная передаточная функция либо непосредственно преобразуется к нему, либо пользуясь известными методами пересчета, например методом прямого или последовательного программирования [Р]. Выбор того или иного метода зависит от формы представления исходной передаточной функции, по критерию минимизации вычислений.

5. Представление объекта управления с запаздыванием по управлению в пространстве состояний

Для стационарных собственных систем с запаздыванием по управлению уравнения в пространстве состояний записываются следующим образом

$$\begin{aligned} \dot{x}(t) &= A \cdot x(t) + B \cdot u(t - \tau) \\ y(t) &= C \cdot x(t) \end{aligned}, \quad (1.25)$$

где τ – известное запаздывание.

По уравнениям (1.26) записывается передаточная функция

$$W(p) = \frac{y(t)}{u(t - \tau)} = C(pI_n - A)^{-1} B = \frac{C(pI_n - A)^+ B}{\det(pI_n - A)} \cdot e^{-\tau \cdot p}. \quad (1.26)$$

6. Случай с зашумленной кривой разгона

В случае зашумленной кривой разгона, например, изображенной на рис. 1.7, необходимо использовать *сглаживание*.

Сглаживание применяется для выделения основного сигнала и удаления постоянных шумов, возникших, например, ввиду действия кого-либо возмущения на объект или наводках в цепях измерения.

Результатом сглаживания является гладкая разгонная кривая (рис.1.1), лишенная помех, пригодная для непосредственной аппроксимации выражениями (1.1) или (1.16). Далее рассматриваются методы, имеющие наибольшее практическое значение и позволяющие добиться приемлемого результата сглаживания без существенного искажения кривых разгона. Приводится реализация методов сглаживания в Matlab.

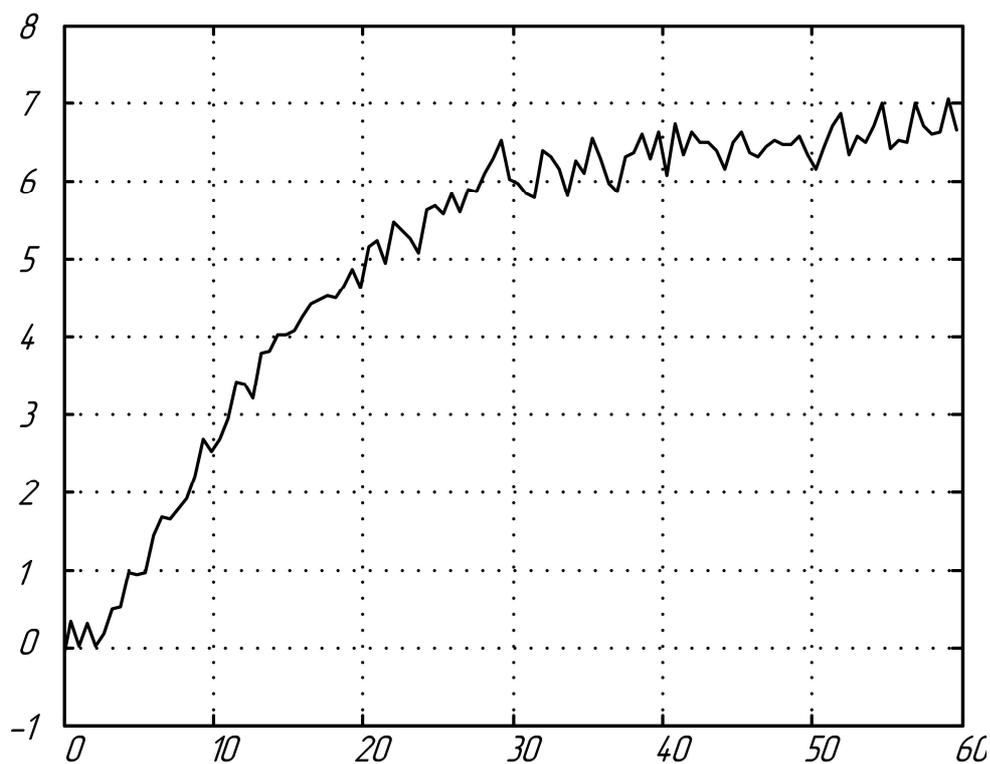


Рис. 1.7 Зашумленная кривая разгона

1. Методы сглаживания.

Суть *метода скользящего среднего* заключается в выравнивании (осреднении) экспериментальных данных путем вычисления средних арифметических значений на некотором интервале времени по небольшому числу измерений l

(то есть в применении гипотезы о близости небольших участков кривой к прямой линии). Число l удобно брать чётным. При общем числе измерений $n \geq 20 \dots 30$ первоначально следует выбрать $l = 2$. Если сглаживание оказалось недостаточным, то l увеличивают. При $l \geq 100$, первоначальное значение $l = 0,1 \cdot n$. Осреднение осуществляется по формуле:

$$h_i = \frac{1}{2l+1} \sum_{k=-l}^{k=l} z_{i+k} \quad (1.27)$$

где h_i – осредненное значение i -й ординаты, z_{i+k} – измеренное значение ординаты.

Основное внимание при использовании данного метода необходимо обращать на выбор числа l . Занижение l ведет к недостаточному выравниванию переходной характеристики, а завышение – к искажению существенных особенностей и потери части ординат. Так как начальный участок кривой разгона определяет структуру искомой передаточной функции, а конечный – передаточный коэффициент (коэффициент усиления), то для сглаживания данным методом, в идеальном случае, необходимо начинать регистрацию $z(t)$ несколько раньше нанесения возмущения и осуществлять несколько замеров $z(t)$ после выхода процесса в установившийся режим. На практике же, данные, расположенные в точках, близких к границам отрезка, не сглаживаются, так как не хватает точек слева и справа от текущей, в которой в данный момент производится сглаживание.

Метод четвертых разностей заключается в аппроксимации с помощью метода наименьших квадратов каждых пяти соседних значений z_i параболой второго порядка:

$$h_i = z_i - \frac{1}{12}(z_{i-2} - 4z_{i-1} + 6z_i - 4z_{i+1} + 4z_{i+2}), \quad i = 2, 3, \dots, (n-2) \quad (1.28)$$

Для определения двух первых и двух последних точек используют следующие формулы:

$$h_0 = \frac{1}{60}(53z_0 + 16z_1 - 6z_2 - 8z_3 + 5z_4) \quad (1.29)$$

$$h_1 = \frac{1}{35}(9z_0 + 13z_1 + 12z_2 + 6z_3 - 5z_4)$$

$$h_{n-1} = \frac{1}{35}(-5z_{n-4} + 6z_{n-3} + 12z_{n-2} + 13z_{n-1} + 9z_n) \quad (1.30)$$

$$h_n = \frac{1}{60}(5z_{n-4} - 13z_{n-3} - 6z_{n-2} + 16z_{n-1} + 53z_n)$$

Операцию сглаживания можно проводить несколько раз. Наилучшие результаты данный метод даёт, если сглаживаемая переходная характеристика соответствует решению дифференциального уравнения порядка выше первого.

2. Реализация на ЭВМ.

Встроенные средства Matlab предоставляют несколько методов сглаживания, в том числе и метод скользящего среднего.

Исходными данными во всех методах является функция, заданная таблично (запаздывание не учитывается) и коэффициент точности сглаживания.

Таблица 1.1

Примеры реализации сглаживания по методу скользящего среднего

<i>Реализация 1</i>	
Функция	Описание
$F = \text{ones}(n, m);$	Задаёт единичную матрицу $n \times m$, тем самым определяется размер «окна», то есть количество точек l , по которым происходит сглаживание.
$O_s = \text{filter2}(F, I_s, \text{'same'}) / (n + m);$	Функция сглаживания, где I_s и O_s – соответственно исходное и обработанное изображение (таблица значений функции).
<i>Реализация 2</i>	
$F = \text{ones}(n, m) / m;$	Количество точек осреднения
$O_s = \text{filtfilt}(F, 1, I_s);$	Функция сглаживания, где I_s и O_s – соответственно исходное и обработанное изображение (таблица значений функции).

Также Matlab содержит достаточно мощный пакет для подбора параметров и приближения данных – Curve Fitting Toolbox. Данный пакет содержит

множество инструментов для начальной фильтрации значений табличной функции; различные способы приближения, такие как параметрические, связанные с подбором параметров модели (одной из стандартных или пользовательскую); позволяет провести анализ результатов, включая экстраполяцию, интегрирование, дифференцирование. Все операции с данными, указанные выше производятся на уровне графического интерфейса.

Пакет инструментов Curve Fitting Toolbox предлагается изучить самостоятельно (вызов пакета осуществляется командой *cftool*).

Применение описанных выше методов проиллюстрирует следующий пример. Исходную зашумленную кривую (рис. 1.7) отфильтруем с помощью функции *filtfilt(...)* два раза. Результат первого этапа фильтрации представлен на рис. 1.8, второго – на рис. 1.9.

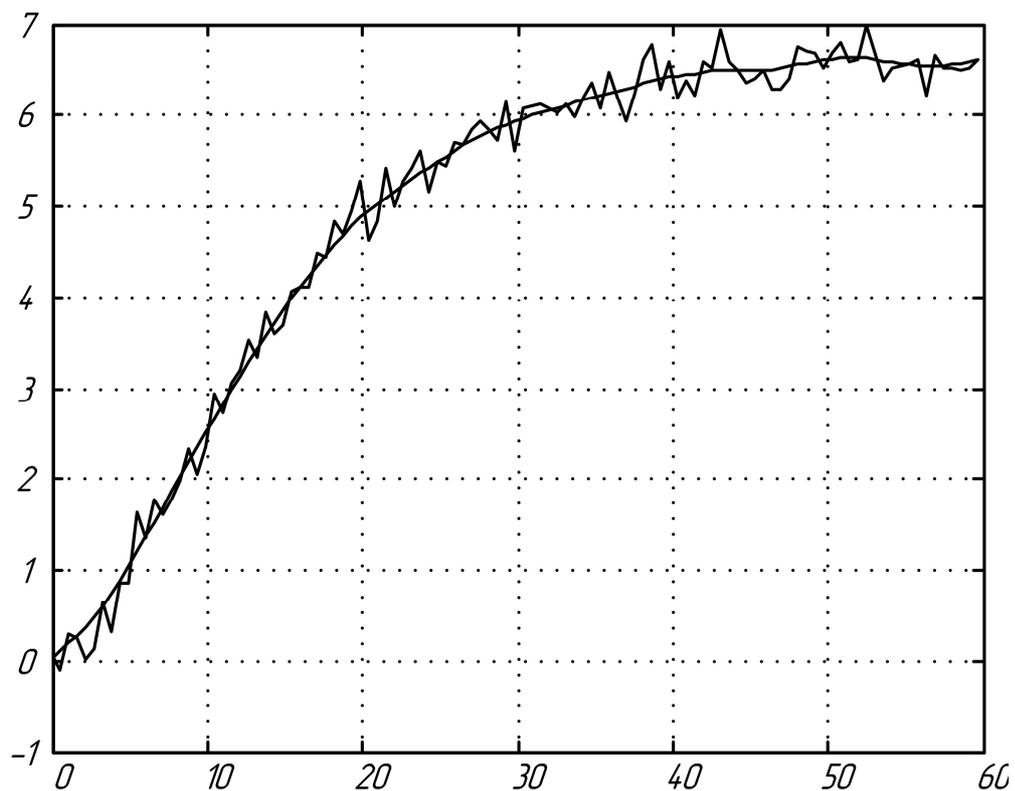


Рис. 1.8 – Первый этап фильтрации

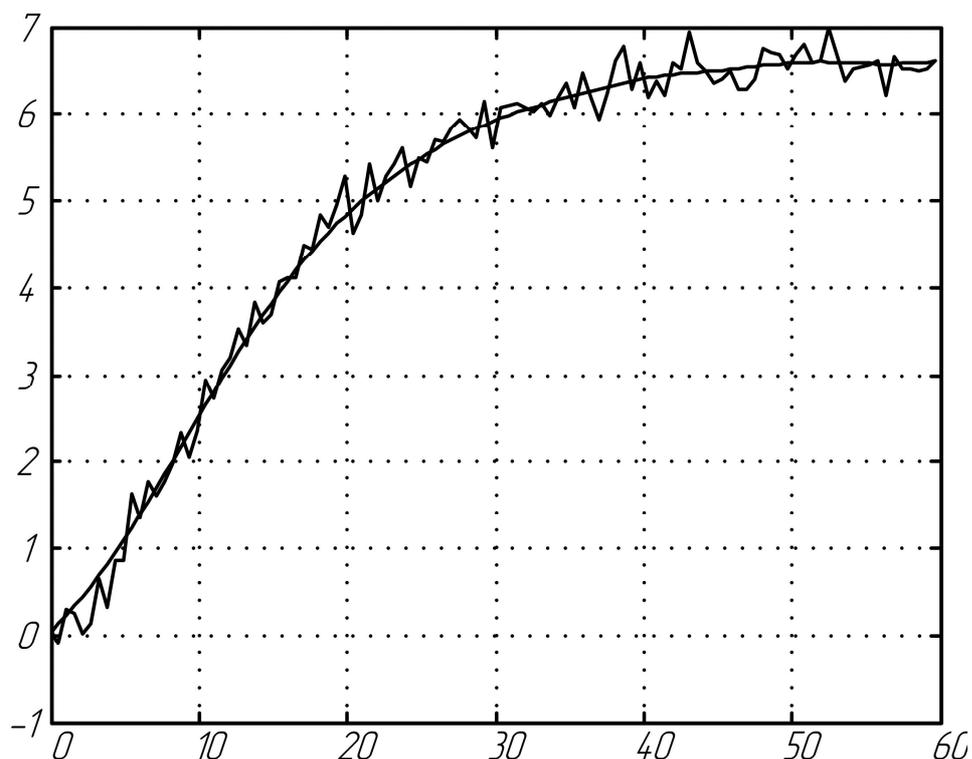


Рис. 1.9 – Второй этап фильтрации

Сглаженная кривая на рис. 1.9 практически идентична кривой на рис. 1.1.

Порядок работы

1. По заданной кривой разгона (согласно варианту в приложении А) получить математическое описание объекта управления. Использовать метод последовательного логарифмирования. Представить математическое описание в виде передаточной функции и в пространстве состояний.
2. По заданной кривой разгона получить математическое описание методом непосредственного определения параметров.
3. Сравнить полученные модели. Сделать выводы.

Содержание отчета

1. Исходные данные, кривые разгона, математическое описание.

ЛАБОРАТОРНАЯ РАБОТА № 2

КЛАССИЧЕСКИЕ СИСТЕМЫ РЕГУЛИРОВАНИЯ

Цель работы:

1. Решение вопроса о выборе системы регулирования для конкретного объекта управления
2. Рассмотрение классических законов управления.

Краткие теоретические сведения

1. Вопрос о выборе класса системы управления

Система управления по ошибке регулирования в общем виде представлена на рис. 2.1.

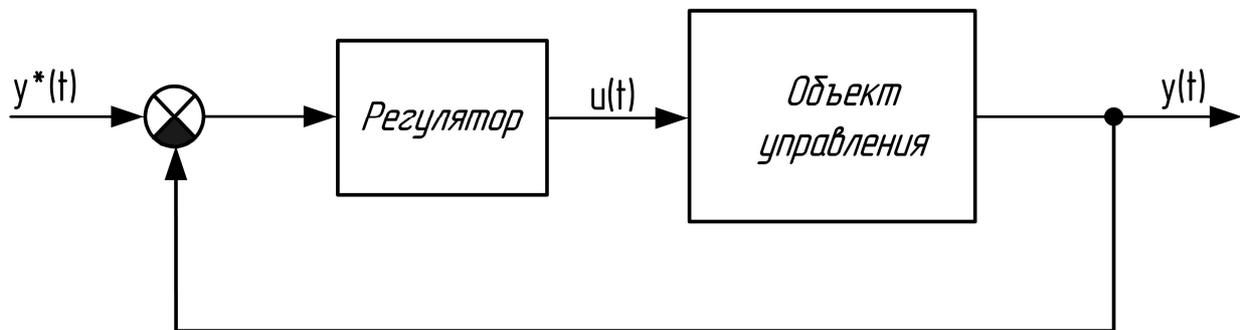


Рис. 2.1 Система регулирования. Общий вид.

Объект управления определяется частным случаем передаточной функцией (1.26) и с учетом транспортного запаздывания τ и коэффициента передачи k имеет вид:

$$W_{OY}(p) = \frac{k}{\prod_{k=1}^n (T_k p + 1)} \cdot e^{-\tau \cdot p}. \quad (2.1)$$

Зачастую в теплоэнергетических системах присутствует регулирующий орган (заслонка, задвижка и т.д.). Поэтому в состав объекта регулирования вводится исполнительный механизм с постоянной времени полного хода T_M , имеющий передаточную функцию:

$$W_{ИМ}(p) = \frac{1}{T_M \cdot p} \quad (2.2)$$

Совокупность объекта управления (2.1) и исполнительного механизма (2.2) называется *обобщенным объектом управления*:

$$W_{ООУ}(p) = W_{ОУ}(p) \cdot W_{ИМ}(p) = \frac{1}{T_M \cdot p} \frac{1}{\prod_{k=1}^n (T_k p + 1)} \cdot e^{-\tau \cdot p} \quad (2.3)$$

Исходными данными для выбора *регулятора* являются [инструкция]:

1. Характеристики объекта:

- чистое запаздывание τ ;
- постоянная времени T (как определяется, если постоянных две);
- их отношение τ/T ;
- коэффициент передачи k (если k изменяется в пределах обычных эксплуатационных режимов, то следует принимать при расчете его наибольшее из возможных значений).

2. Максимально возможные значения возмущений по нагрузке $f(t)$, процент хода регулирующего органа – пиковых, скачкообразно длительных и непрерывных монотонных (при непрерывных возмущениях должна быть известна также максимальная скорость возмущения $\dot{f}(t)$, проценты/сек.).

3. Требуемые показатели качества регулирования объекта:

- время регулирования;
- перерегулирование;
- максимальное динамическое отклонение и число колебаний регулируемой величины около линии установившегося значения за время регулирования.

Таким образом, для выполнения требований формируется область применения законов регулирования. В основном характер действия регулятора определяется по отношению τ/T :

- при $\tau/T < 0,2$ выбирается позиционное регулирование;

– при $0,2 < \tau/T < 1,0$ выбирается непрерывное регулирование, либо импульсное;

– при $\tau/T > 1,0$ применяют многоконтурные системы регулирования и принимают меры по компенсации влияния запаздывания, например, адаптивные [3], [4].

Таким образом, следует, что даже при оптимальной настройке регулятора качество процесса регулирования тем хуже, чем больше и само запаздывание и отношение запаздывания к постоянной времени объекта [РАН].

2. Описание классической системы управления

В современных системах управления технологическими процессами широко применяются регуляторы на основе микропроцессорных контроллеров. Отличительной особенностью таких регуляторов является переменная структура, устанавливаемая программно. Например, в контроллере КР-300 (Р-130, КР-500, МК-500 и т.д.) используется ПИД – регулятор, работающий в комплекте с исполнительным механизмом постоянной скорости, передаточную функцию и имеющий структуру:

$$W_{\text{ПИД}}(p) = K_{\text{П}} \left[1 + \frac{1}{T_{\text{И}} \cdot p} + \frac{K_{\text{Д}} \cdot T_{\text{И}} \cdot p}{\left(1 + \frac{1}{8} K_{\text{Д}} \cdot T_{\text{И}} \cdot p\right)^2} \right] \quad (2.4)$$

где $K_{\text{П}}$ – коэффициент пропорциональности;

$T_{\text{И}}$ – постоянная времени интегрирования;

$K_{\text{Д}}$ – коэффициент времени дифференцирования.

Ввиду присутствия в структуре обобщенного объекта управления (2.3) исполнительного механизма, ПИД - закон преобразуется в ПДД²-закон. Исключив из (2.4) звено, вводится следующая структура реального ПДД²-регулятора

$$W(p) = K_{\text{П}} \cdot \frac{T_{\text{М}}}{T_{\text{И}}} \left[1 + T_{\text{И}} \cdot p + \frac{K_{\text{Д}} \cdot T_{\text{И}}^2 \cdot p^2}{\left(1 + \frac{1}{8} K_{\text{Д}} \cdot T_{\text{И}} \cdot p\right)^2} \right] \quad (2.5)$$

Необходимо отметить, что элемент $T_{II} \cdot p$ в (2.5) физически не реализуем. Для «правильной» реализации дифференциальной составляющей необходимо подавление высокочастотных помех, присутствующих во входном сигнале. Эту функцию фильтрации выполняет апериодическое звено первого порядка с небольшой (относительно постоянных времени системы) постоянной времени:

$$W(p) = K_{II} \cdot \frac{T_M}{T_{II}} \left[1 + \frac{T_{II} \cdot p}{T_{\Phi} p + 1} + \frac{K_D \cdot T_{II}^2 \cdot p^2}{\left(1 + \frac{1}{8} K_D \cdot T_{II} \cdot p\right)^2} \right] \quad (2.6)$$

Разница между регуляторами (2.5) и (2.6) уменьшается с уменьшением T_{Φ} . Структурная схема регулятора (2.6) показана на рис. 2.2.

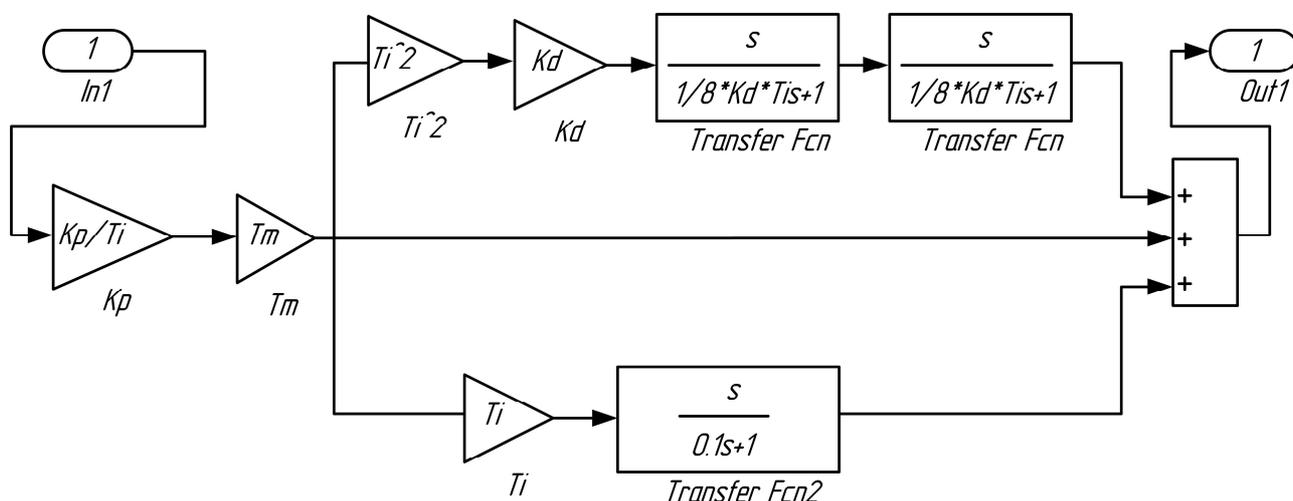


Рис. 2.2 Структурная схема ПДД²-регулятора

Для управления исполнительным механизмом постоянной скорости может быть использован блок широтно-импульсной модуляции, который в общем случае входит в блок регулирования. Структура блока широтно-импульсной модуляции может быть представлена в виде рис. 2.3. Такой блок может быть снабжен стандартными средствами интерфейса Matlab, как показано на рис. 2.4.

В данном диалоговом окне задаются основные параметры модулятора:

- частота модуляции;
- входной сигнал, соответствующий 100% скважности;
- минимальное время выходного сигнала (минимальное время импульса).

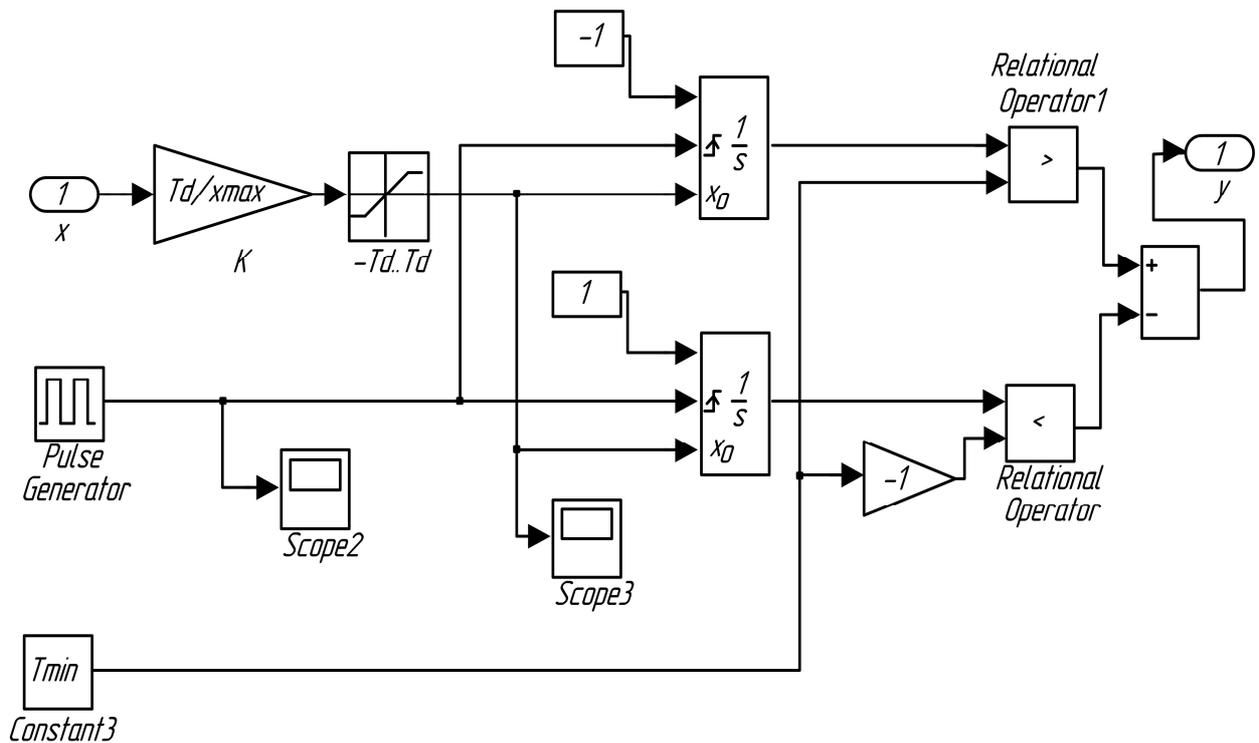


Рис.2.3 Принципиальная схема блока широтно-импульсной модуляции

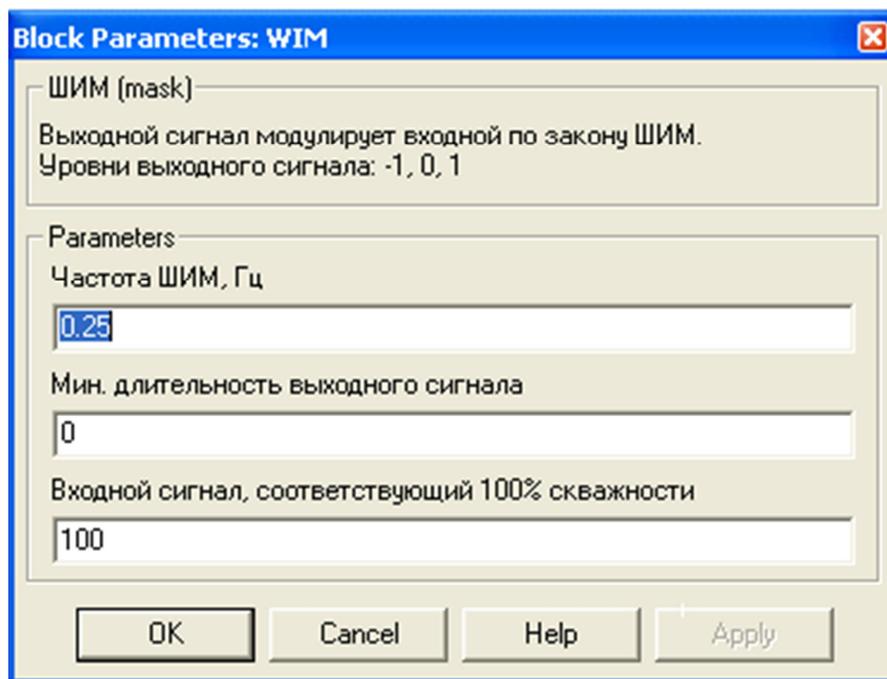


Рис. 2.4 Параметры блока широтно-импульсной модуляции

Принцип работы блока широтно-импульсной модуляции следующий. Генератор импульсов (Pulse Generator) с заданной частотой $f = 1/T_d$ (T_d - период импульсов) и скважностью $\gamma = T_u/T_d$ вырабатывает короткие тактовые импуль-

сы единичной амплитуды (рис. 2.5). На вход x подается сигнал управления. Модуль этого сигнала численно равен времени выходного импульса ШИМ, а полярность определяет направление движения механизма («больше»/«меньше»). В момент прихода каждого тактового импульса входящие в структуру интеграторы сбрасывают свое состояние и загружаются начальным значением с управляющего входа. После этого выходные сигналы интеграторов начинают изменяться с единичной скоростью. Выходной сигнал первого интегратора уменьшается, второго – увеличивается. Блоки сравнения сравнивают эти сигналы с нулем (или минимальным временем импульса T_{\min}), формируя единицы (истина) или нули (ложь) на своих выходах. На выходе y формируется результирующий импульс. Так, например, если интеграторы были загружены положительным начальным значением, то на выходе первого блока сравнения в течение заданного времени будет держаться единица, на выходе второго – ноль в течение всего периода модуляции. Таким образом, импульс на выходе y будет иметь заданную длительность.

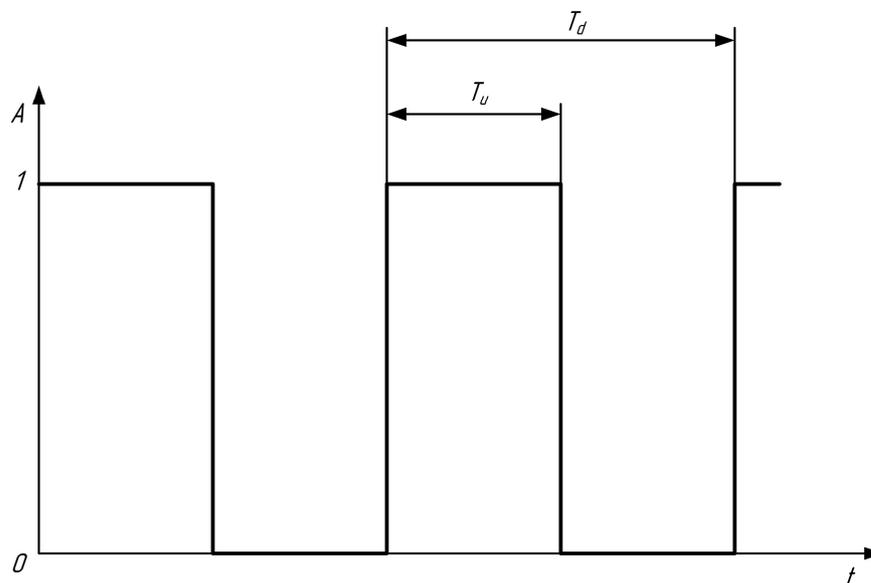


Рис. 2.5 Последовательность импульсов на выходе блока широтно-импульсной модуляции

Таким образом, общая структура классической системы регулирования показана на рис. 2.6.

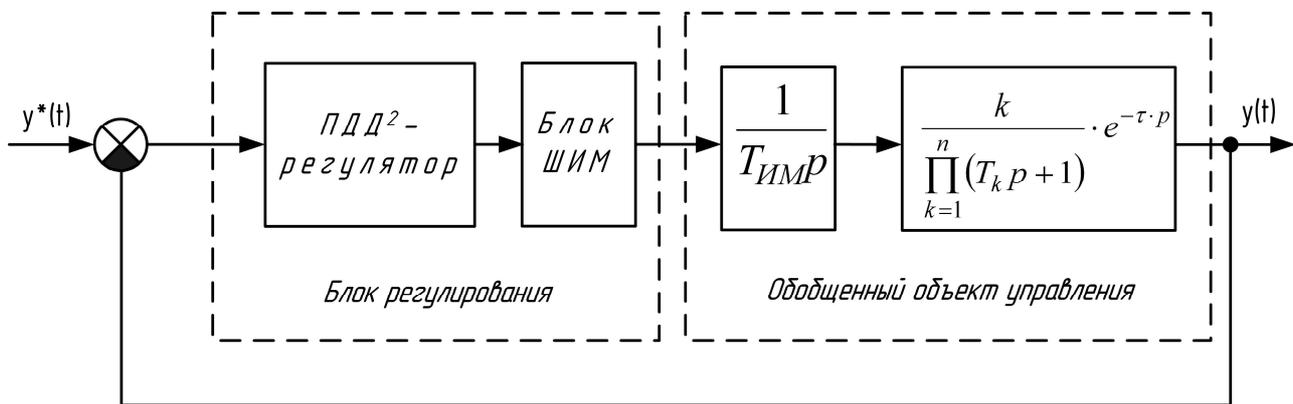


Рис. 2.6 Структура классической системы регулирования

3. Расчет классической системы управления

Существует достаточное количество методов расчета ПИД – регуляторов на какой-либо критерий оптимальности [19]. Одним из возможных вариантов является расчет системы с ПИД – регулятором и постоянным запаздыванием на заданный частотный показатель колебательности. Необходимо отметить, что в рамках системы, рассмотренной на рис. 2.6 необходимо пересчитать параметры полученного ПИД – регулятора в параметры ПДД²-регулятора.

Частотный показатель колебательности Λ есть отношение максимального значения амплитудно-частотной характеристики замкнутой системы к значению этой характеристики при нулевой частоте:

$$M = \frac{A_{зам}(\omega_{рез})}{A_{зам}(0)} = \frac{|W_{зам}(\omega_{рез})|}{W_{зам}(0)}, \quad (2.7)$$

где $\omega_{рез}$ – резонансная частота, при которой и наблюдается максимум АЧХ системы (рис. 2.7).

Методика расчета заключается в следующем. Исходную систему приводят к расчетной схеме, изображенной на рис. 2.8. Искомой величиной является коэффициент передачи регулятора k_p . Передаточная функция $W'_{раз}(p)$ представляет собой передаточную функцию последовательного соединения всех элементов САУ, в том числе и регулятора (с единичным коэффициентом передачи).

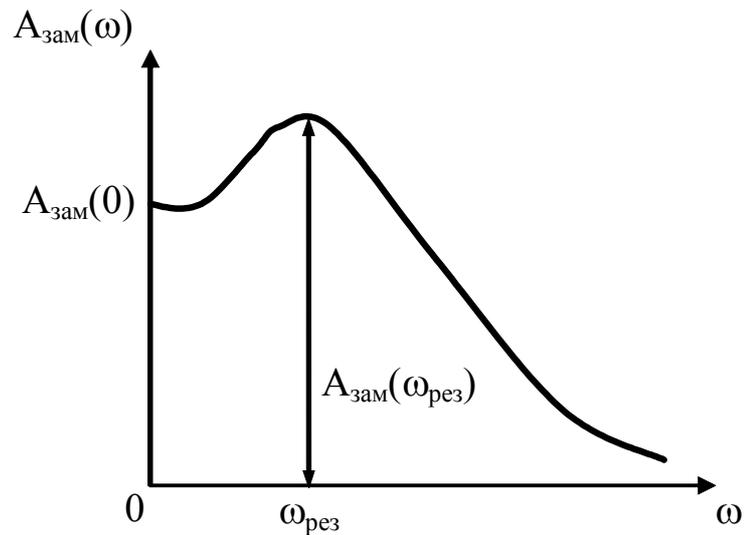


Рис. 2.7. К определению частотного показателя колебательности

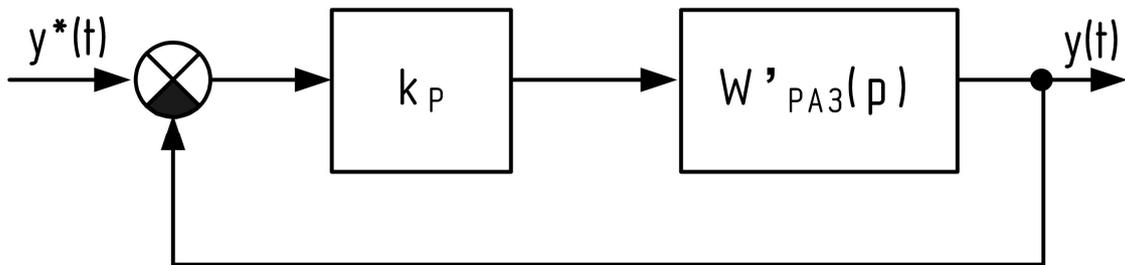


Рис. 2.8 Расчетная схема системы

При расчете ПИД – регулятора с передаточной функцией

$$W_{\text{ПИД}}(p) = k_p \left(1 + \frac{1}{T_{\text{из}} p} + T_d p \right) \quad (2.8)$$

подлежит определению только величина k_p . Постоянные времени $T_{\text{из}}$ и T_d зада-

ются, а множитель $\left(1 + \frac{1}{T_{\text{из}} p} + T_d p \right)$ вносится в $W'_{\text{раз}}(p)$.

Далее используется критерий максимального подавления низкочастотных возмущений, который устанавливает связь между постоянными времени: $T_d = 0,5T_{\text{из}}$. Задавшись несколькими значениями $T_{\text{из}}$, строят семейство характеристик, по которым определяют несколько вариантов k_p . Оптимальным будет вариант, для которого отношение $k_p/T_{\text{из}}$ максимально.

Для определения коэффициента передачи регулятора k_p существует графический метод расчета (приложение Г).

Для упрощения расчетов воспользоваться аналитическим методом в Matlab. Для этого необходимо решить задачу условной оптимизации.

Максимум амплитудно-частотной характеристики замкнутой системы равен заданному значению Λ :

$$\max(A_{\text{çàì}}(k_{\delta})) = \Lambda. \quad (2.9)$$

Соответствующая задача оптимизации

$$(\max(A_{\text{çàì}}(k_{\delta})) - \Lambda)^2 \rightarrow \min. \quad (2.10)$$

Код программы расчета показан в листинге на рисунке 2.9.

```
function s = f_mnk(k)

%Формирование передаточной функции объекта регулирования
w = zpk([], [-1/10 -1/5], 1/50);
w.InputDelay = 7; %Запаздывание

%Задание значения времени изодрома
Tiz = 10;
Td=0.5*Tiz;

%Формирование передаточной функции ПИД-регулятора
wpid=tf([Tiz*Td Tiz 1 ], [Tiz 0]);

%Заданный частотный показатель колебательности
l = 1.2;

%Формирование w`раз(p)
ww=w*wpid;

%Аппроксимация Паде
w1 = PADE(ww,7);
w2 = k*w1/(1+w1);

%Функция оптимизации
s=(max(bode(w2))-1)^2;
```

Рис. 2.9 Листинг программы расчета регулятора.

Для получения значения коэффициента передачи регулятора k_p необходимо вызвать функцию $fminbnd('f_mnk', k_0, k_\infty)$, где k_0, k_∞ границы интервала значений коэффициента передачи k_p .

Преобразование параметров регулятора (2.8) в постоянные регулятора (2.4) представлено в таблице 1.

Таблица 1

Таблица соответствия параметров регуляторов (2.8) и (2.4)

Параметр	Регулятор (2.8)	Регулятор (2.4)
Коэффициент передачи	k_p	K_P
Постоянная издрорма	$T_{из}$	T_I
Постоянная дифференцирования	T_δ	$K_D T_I$
Коэффициент времени дифференцирования	–	$K_D = \frac{T_\delta}{T_I}$

Значения параметры ПДД²-регулятора (2.6) полностью соответствуют приведенным в таблице 1.

Замечание 1. С целью сокращения объемов вычислений, в данной лабораторной работе показан *общий принцип* расчета классического регулятора на заданный частотный показатель колебательности, поэтому в качестве примера был взят «идеальный» ПИД – регулятор. В практических же условиях предпочтительнее рассчитывать систему управления с «реальным» ПИД-регулятором [20], в котором чистое дифференцирование заменено реальным дифференцированием, например, как это сделал при переходе от формулы (2.5) к (2.6). Вследствие выше указанных причин в ходе сеанса имитационного моделирования возможна корректировка рассчитанных значений ПИД (ПДД²) – регулятора.

Порядок работы

1. Исходя из характеристик заданного объекта, предложить вид системы управления, которая наилучшим образом обеспечила бы требуемые показатели качества.

2. Рассчитать ПИД-регулятор на заданный частотный показатель колебательности ($\Lambda = 1,1 - 1,5$).
3. Рассчитать ПДД² – регулятор (таблица 1).
4. Собрать в Simulink систему регулирования с объектом (2.1) и регулятором (2.4).
5. Собрать в Simulink систему регулирования с объектом (2.3) и регулятором (2.6).
6. Оценить качество работы собранных систем (перерегулирование, время переходного процесса).
7. В системе с ПДД² – регулятором для управления исполнительным механизмом применить ШИМ. Оценить качество работы системы, сопоставить управляющее воздействие регулятора с последовательностью импульсов, формируемой ШИМ.
8. Сравнить работу системы с применением ШИМ и в его отсутствие.

Содержание отчета

1. Оценка характеристик объекта и предложения по выбору системы регулирования.
2. Расчет настроек ПИД – регулятора (графический или аналитический).
3. Расчет параметров ПДД² – регулятора.
4. Simulink – модели классических систем регулирования (ПИД, ПДД², ПДД²+ШИМ).
5. Переходные характеристики основных параметров и сигналов моделируемых систем.

Лабораторная работа № 3

Адаптивная система регулирования с эталонной моделью

Цель работы:

1. Рассмотрение адаптивной системы управления с эталонной моделью.
2. Рассмотрение видов настроек, применяемых в адаптивных системах.

Краткие теоретические сведения

1. Область применения адаптивных систем управления

Как было показано в лабораторной работе №2, область применения адаптивных систем распространяется на случай, когда постоянная времени объекта сравнима с чистым запаздыванием в системе. Также применяя адаптивные алгоритмы настройки, можно скомпенсировать недостаток априорной информации об объекте, учесть его нестационарность и нелинейность, а также подавить возмущения, неучтенные на этапе синтеза системы управления [18].

2. Адаптивная система регулирования с эталонной моделью

В общем виде система управления имеет структуру (рис. 3.1), состоящую из следующих блоков: объект управления (ОУ), эталонная модель (ЭМ), адаптивный регулятор (АР), алгоритмы адаптации (АА). Блок алгоритмов адаптации формирует параметрические (ПН) и сигнальные (СН) алгоритмы настроек.

Объект управления в общем виде определяется выражениями (1.20)

В некоторых случаях [1], [20] параметры объекта управления (1.20) могут быть нестационарными, нелинейными, параметрически неопределенными, а на сам объект оказывают воздействие неучтенные возмущения.

Поэтому математическое описание объекта (1.20) необходимо дополнить следующим образом:

$$\begin{cases} \dot{X} = F(X, U, f, t) = (A(t) + a(X)) \cdot X + (B(t) + b(X)) \cdot U + f, \\ Y = C(t) \cdot X, \end{cases} \quad (3.1)$$

где $A(t)$ и $B(t)$ – нестационарные матрицы, определяющие линейную составляющую поведения объекта; $a(X)$, $b(X)$ – нелинейные составляющие описания; f – возмущения; C – стационарная матрица выхода.

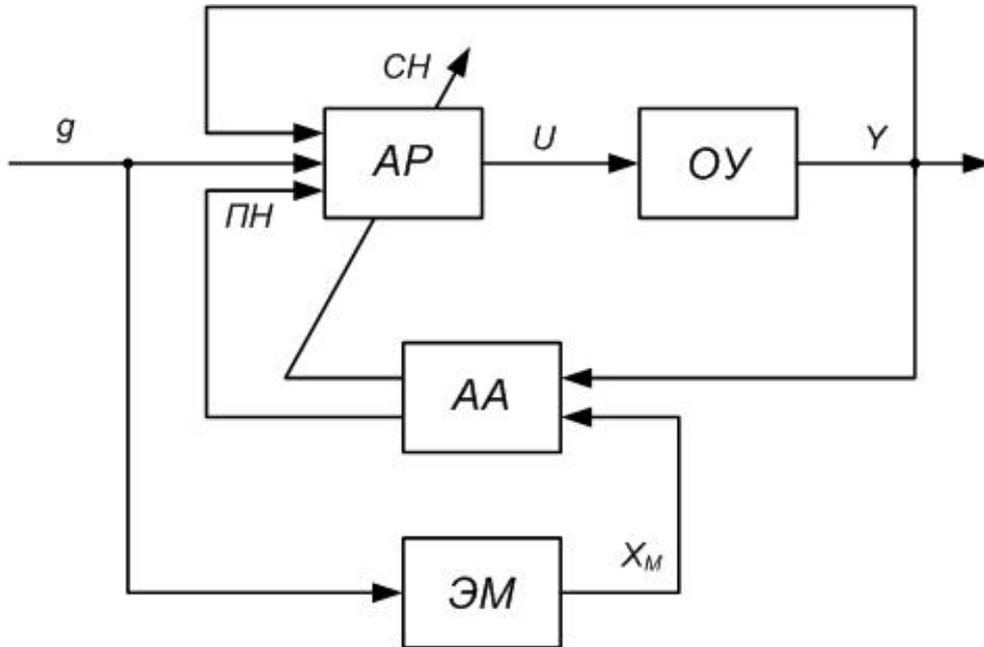


Рис.3.1 Адаптивная система регулирования с эталонной моделью

У объекта (3.1) выделяется стационарную часть и рассогласования:

$$\dot{X} = A_0 X + B_0 U + \sigma + \varphi \quad (3.2)$$

$$\begin{cases} \sigma = (A - A_0)X + (B - B_0)U \\ \varphi = f + a(X)X + b(X)U \end{cases} \quad (3.3)$$

где A_0 и B_0 – стационарные матрицы, σ – параметрические рассогласования, φ – сигнальные рассогласования.

Таким образом, при представлении объекта управления в виде (3.2) и (3.3) матрицы A_0 и B_0 описывают желаемое поведение объекта, то есть эталонную динамику, которой необходимо подчинить объект.

Эталонная модель формирует эталонную динамику движения:

$$\begin{cases} \dot{X}_M = A_M X_M + B_M g \\ Y_M = X_M \end{cases} \quad (3.4)$$

где A_M и B_M – матрицы эталонной динамики, g – задающее воздействие.

Цель управления, с учетом возможности измерения всех координат объекта управления (3.2), формируется следующим образом:

$$\lim_{t \rightarrow \infty} e = \lim_{t \rightarrow \infty} (X_M - X) \rightarrow 0 \quad (3.5)$$

где e – вектор рассогласования между моделью и объектом.

Адаптивный регулятор, обеспечивающий выполнение цели управления (3.5), должен выработать управляющее воздействие, способное подавить сигнальные φ и параметрические рассогласования σ :

$$U = K_a X + K_b (g + z) \quad (3.6)$$

где z – сигнальная настройка, K_a и K_b – матрицы настраиваемых параметров, формируемые блоком алгоритмов адаптации на основании сравнения координат модели и объекта:

$$\begin{cases} \dot{K}_a = B_M^T \cdot P \cdot e \cdot X^T \cdot R_a \\ \dot{K}_b = B_M^T P \cdot e \cdot (g + z) \cdot R_b \end{cases}, \quad (3.7)$$

$$z = h \cdot \text{sign}(B_M^T \cdot P \cdot e), \quad (3.8)$$

где R_a , R_b , h – диагональные положительные матрицы параметрической настройки и коэффициент сигнальной настройки соответственно, $P = P^T > 0$ матрица находится из решения уравнения Ляпунова:

$$PA_M + A_M^T P = -Q \quad (3.9)$$

где $Q = Q^T > 0$ – диагональная положительная матрица.

Из уравнений (3.7) и (3.8) следует, что коэффициенты параметрической настройки формируются по интегральному закону, а сигнальная настройка – знаковому алгоритму.

Вектор ошибок рассогласования между моделью (3.4) и объектом (3.2) может быть получен только при условии структурной согласованности модели и объекта, которое имеет вид:

$$\begin{cases} BK_a^0 = A_M - A \\ BK_b^0 = B_M \end{cases}, \quad (3.10)$$

где K_a^0 и K_b^0 – значения соответствующих коэффициентов после окончания процесса адаптации, характеризующегося целью адаптации:

$$\lim_{t \rightarrow \infty} [(A + BK_a - A_m)X + (BK_b - B_m)g + BK_b z + \varphi] = 0, \quad (3.11)$$

3. Область применения параметрической и сигнальной настроек

Выше рассмотрена адаптивная система с комбинированным типом настроек. В реальных системах с эталонной моделью может применяться, как только параметрическая, так и только сигнальная настройка. Использование лишь параметрической настройки целесообразно, когда параметры объекта изменяются в широких пределах, но скорость их изменения незначительна. Если параметры изменяются в небольшом диапазоне, но с высокой скоростью, либо объект подвержен сигнальным рассогласованиям (нелинейности и возмущения), предпочтительнее использование сигнальной настройки. В наиболее «тяжелых» случаях используется оба типа настройки; при этом «быстрые» рассогласования подавляются сигнальной настройкой, а «медленные» – параметрической (при этом сигнальная настройка не реагирует на небольшие рассогласования благодаря тому, что в алгоритм вводится зона нечувствительности).

В общем случае сигнальная настройка более универсальна благодаря простоте реализации и более высокому быстродействию. Однако ее возможности ограничены способностью регулирующих органов, непосредственно воздействующих на объект, формировать управляющие сигналы в скользящем режиме с заданной амплитудой сигнала. В реальных условиях всегда ограничены как уровень сигнала на выходе релейного блока, так и частота его переключения. В этом состоит один из недостатков системы с эталонной моделью.

Порядок работы

1. Задать динамику эталонной модели полиномом с распределением корней по Баттерворту с $\omega_0 = 1$ (приложение Б). Для решения уравнения (3.9) отно-

нительно матрицы P рекомендуется воспользоваться библиотечной функцией из Matlab:

$$P = \text{lyap}(Am', Q) \quad (3.12)$$

3. Для использования только параметрической настройки адаптивного регулятора (3.6) принять $z = 0$. При использовании только сигнальной настройкой (3.6) принять $K_b = 1$, а вектор K_a рассчитывается как вектор обратных связей модального регулятора [метод РАН] из условия:

$$B_0 K_a = A_0 - A_m. \quad (3.13)$$

Для изучения сигнала z предусмотреть его фильтрацию фильтром первого порядка с небольшой постоянной времени.

2. Собрать в Simulink систему регулирования, состоящую из:

- объекта управления (2.3), запаздывание принять равным нулю;
- адаптивного регулятора (3.6);
- параметрической настройки (3.7);
- сигнальной настройки (3.8);
- эталонной модели (3.4).
- задающего сигнала (ступенчатого, синусоидального, «белого шума»).

3. Предусмотреть возможность использования, как только параметрической или сигнальной настройки, так и комбинированной настройки.

4. Организовать контур параметрических и сигнальных рассогласований, а также возмущений, например, как на рис. 3.2.

При этом параметрические рассогласования моделируются изменением коэффициентов матрицы A , а сигнальные рассогласования можно задать в виде:

$$\varphi = a(X)X + f = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ k_1 x_2 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} 0 \\ 0 \\ k_2 \sin(\omega t) \end{bmatrix} \quad (3.14)$$

где k_1, k_2 будут определять влияние нелинейности и возмущения (последнее в данном случае задано в виде синусоидального сигнала небольшой частоты ω).

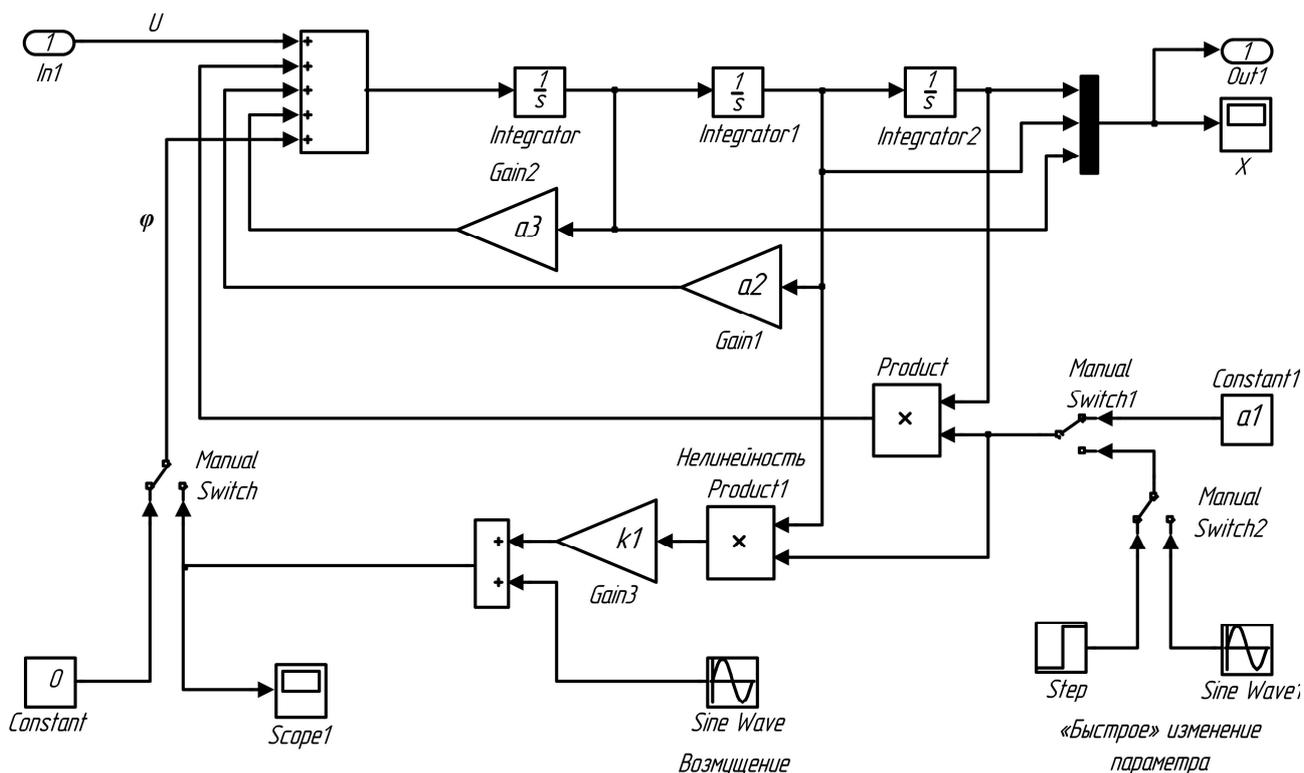


Рис. 3.2 Организация рассогласований и возмущений в системе

5. При отключенном регуляторе:

- исследовать влияние «быстрых» параметрических рассогласований на «поведение» объекта, для чего смоделировать ситуацию изменения его параметров во время обработки ступенчатого входного воздействия. В случае необходимости скорректировать модель так, чтобы это влияние было заметным;

- исследовать влияние нелинейности и возмущения на «поведение» объекта при ступенчатом входном воздействии. В случае необходимости подобрать коэффициенты, чтобы это влияние было заметным.

6. Настроить параметрическую и сигнальную составляющие адаптивного регулятора для подавления соответствующих рассогласований в отдельности.

7. Исследовать процессы адаптации при отработке системой ступенчатого и синусоидального входных воздействий и начальных рассогласований.

8. Оценить качество работы собранных систем (перерегулирование, время переходного процесса, скорость процесса адаптации, диапазон изменения сигналов настроек и управления).

9. Исследовать возможность подавления параметрической настройкой сигнальных рассогласований.
10. Ввести в систему запаздывание. Оценить качество ее работы (пункт 8).

Содержание отчета

1. Simulink – модель адаптивной системы.
2. Числовые значения расчетных и настраиваемых коэффициентов.
3. Переходные характеристики, соответствующие выполняемым заданиям. Оценка их качества.

Лабораторная работа № 4

Адаптивная система регулирования с компенсацией запаздывания

Цель работы:

1. Изучение адаптивной системы управления со стабилизирующим устройством, компенсирующим запаздывание.
2. Изучение диссипативной системы управления со стабилизирующим устройством.

Краткие теоретические сведения

1. Адаптивная система управления со стабилизирующим устройством

Структура адаптивной системы, представленная на рис. 4.1, состоит из следующих основных блоков: обобщенный объект управления (ООУ), адаптивный регулятор (АР), алгоритмы адаптации (АА), стабилизирующее устройство (СУ).

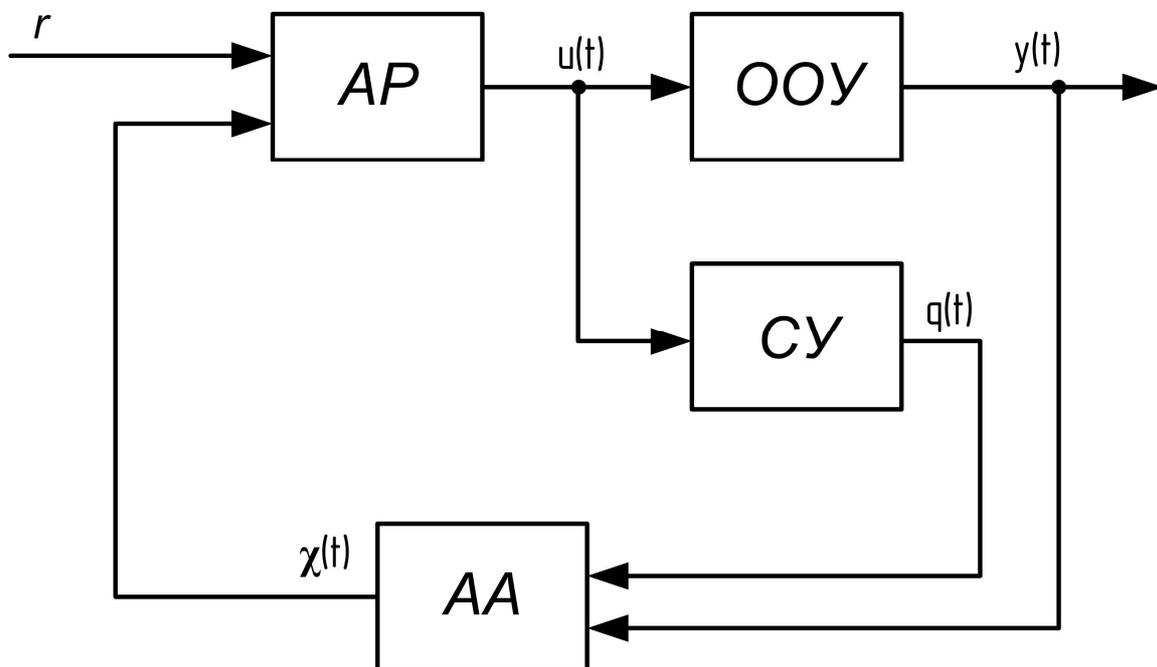


Рис. 4.1. Структура адаптивной системы со стабилизирующим устройством

Обобщенный объект управления (2.3) с запаздыванием по управлению с учетом форм записи (1.25) и (1.26) задается в операторной форме

$$p \cdot a(p) \cdot y(p) = b(p) \cdot u(t - \tau), \quad (4.1)$$

где $u(t) \in R$ – управляющее воздействие; τ – известное постоянное запаздывание, величина которого соразмерна с постоянной времени объекта; $y(t) \in R$ – выход объекта; $a(p)$, $b(p)$ – полиномы от оператора дифференцирования, заданные в виде:

$$\begin{cases} a(p) = p^n + a_{n-2}p^{n-1} + \dots + a_1p + a_0, & a_0 > 0, \\ b(p) = b_m p^m + b_{m-1}p^{m-1} + \dots + b_1p + b_0, & b_0 > 0, \\ p = \frac{d}{dt}, & \deg a(p) = n, \quad \deg b(p) = m, \quad l = n - m > 1 \end{cases}, \quad (4.2)$$

где $a(p)$, $b(p)$ – гурвицевы полиномы.

Функционирование объекта (4.1), (4.2) происходит в условиях априорной параметрической неопределенности, уровень которой задан в виде

$$a_i = a_i(\xi), \quad b_j = b_j(\xi), \quad \tau = \tau(\xi), \quad i = \overline{0, n-1}, \quad j = \overline{0, m}, \quad (4.3)$$

где ξ – набор неизвестных параметров, принадлежащих известному множеству Ξ .

С целью придания системе управления желаемых свойств, в основной контур вводится стабилизирующее устройство

$$W_{CV}(p) = \frac{q(t)}{u(t)} = \frac{k_c}{T_c \cdot p + 1}, \quad (4.4)$$

где k_c – коэффициент передачи; T_c – постоянная времени, $q(t)$ – выход стабилизирующего устройства.

Структура адаптивного регулятора задается в виде

$$\begin{cases} u(s) = \frac{k_p \cdot p}{T_p \cdot p + 1} \cdot g(p), \\ g(t) = \chi(t) \cdot r(t) \end{cases}, \quad (4.5)$$

где k_p и T_p – коэффициент передачи и постоянная времени линейной части адаптивного регулятора; $u(t)$ – управляющее воздействие; $g(t)$ – выход нелинейной части адаптивного регулятора; $r(t)$ – задающее воздействие, равное в

режиме стабилизации r_* ; $\chi(t)$ – настраиваемый параметр адаптивного регулятора, алгоритм адаптации которого подлежит определению.

Цели управления и адаптации:

$$\lim_{t \rightarrow \infty} y(t) = y_* = r_* = const \quad (4.6)$$

$$\lim_{t \rightarrow \infty} \chi(t) = \chi_0 = const. \quad (4.7)$$

Структура алгоритма настройки регулятора (4.5) задается таким образом, чтобы выполнялись цели управления и адаптации:

$$\chi(t) = \chi_{ИИ}(t) + \chi_{ИП}(t) \quad (4.8)$$

$$\frac{d\chi_{ИИ}(t)}{dt} = \alpha \cdot e(t) \cdot r_*, \quad \chi_{ИП}(t) = \beta \cdot e(t) \cdot r_* \quad (4.9)$$

$$e(t) = r_* - y(t) - q(t), \quad \alpha, \beta = const > 0$$

2. Диссипативная система управления

Рассмотренную выше задачу стабилизации обобщенного объекта управления можно свести к управлению объектом с кусочно-постоянным задающим воздействием, которое формируется как набор кусочно-постоянных значений, следующих друг за другом во времени

$$r(t) = r_{\theta_*} \cdot 1(t), \quad r_{\theta_*} = const, \quad t \in [t_{\theta_{нач}}, t_{\theta_{кон}}] \quad \theta = \overline{1, l}, \quad (4.10)$$

где r_{θ_*} – значение величины задающего воздействия на θ -ом интервале времени, l – число временных интервалов постоянства $r(t)$; $t_{\theta_{нач}}$, $t_{\theta_{кон}}$ – соответственно начало и конец θ -го интервала времени. В этом случае целевые условия (4.6) и (4.7) будут иметь вид:

$$\begin{cases} |y(t) - r_*| \leq \psi_1 = const, \\ |\chi(t) - \chi_0| \leq \psi_2 = const, \\ \forall t > t_*, t \in [t_{нач}, t_{кон}], \quad t_{нач} \leq t_* \leq t_{кон} \end{cases} \quad (4.11)$$

Модифицированные алгоритмы адаптации (4.8) и (4.9), с учетом соотношений

$$\begin{cases} |e_\theta(t)| \leq \psi = \text{const}, \quad \forall t_\theta > t_{\theta*}, \\ t \in [t_{\theta_{\text{нач}}}, t_{\theta_{\text{кон}}}], \quad t_{\theta_{\text{нач}}} \leq t_{\theta*} \leq t_{\theta_{\text{кон}}}, \quad \theta = \overline{1, l}, \end{cases} \quad (4.12)$$

примут вид

$$\chi(p) = \frac{\alpha + \beta \cdot p}{p} \cdot \mathcal{G}(p), \quad \mathcal{G}(t) = \begin{cases} e(t) \cdot r(t), & \forall |e(t)| > \psi, \\ 0, & \forall |e(t)| \leq \psi, \end{cases} \quad (4.13)$$

$$e(t) = r(t) - y(t) - q(t), \quad \alpha, \beta, \psi = \text{const} > 0.$$

Фактически для реализации алгоритмов настройки (4.13) и обеспечения диссипативности системы управления вводится зона нечувствительности по ошибке регулирования, то есть алгоритмы настройки «огрубляются».

Синтез адаптивной системы управления на основе критерия гиперустойчивости показан в [3].

Структура адаптивной системы со стабилизирующим устройством показана на рис.4.2.

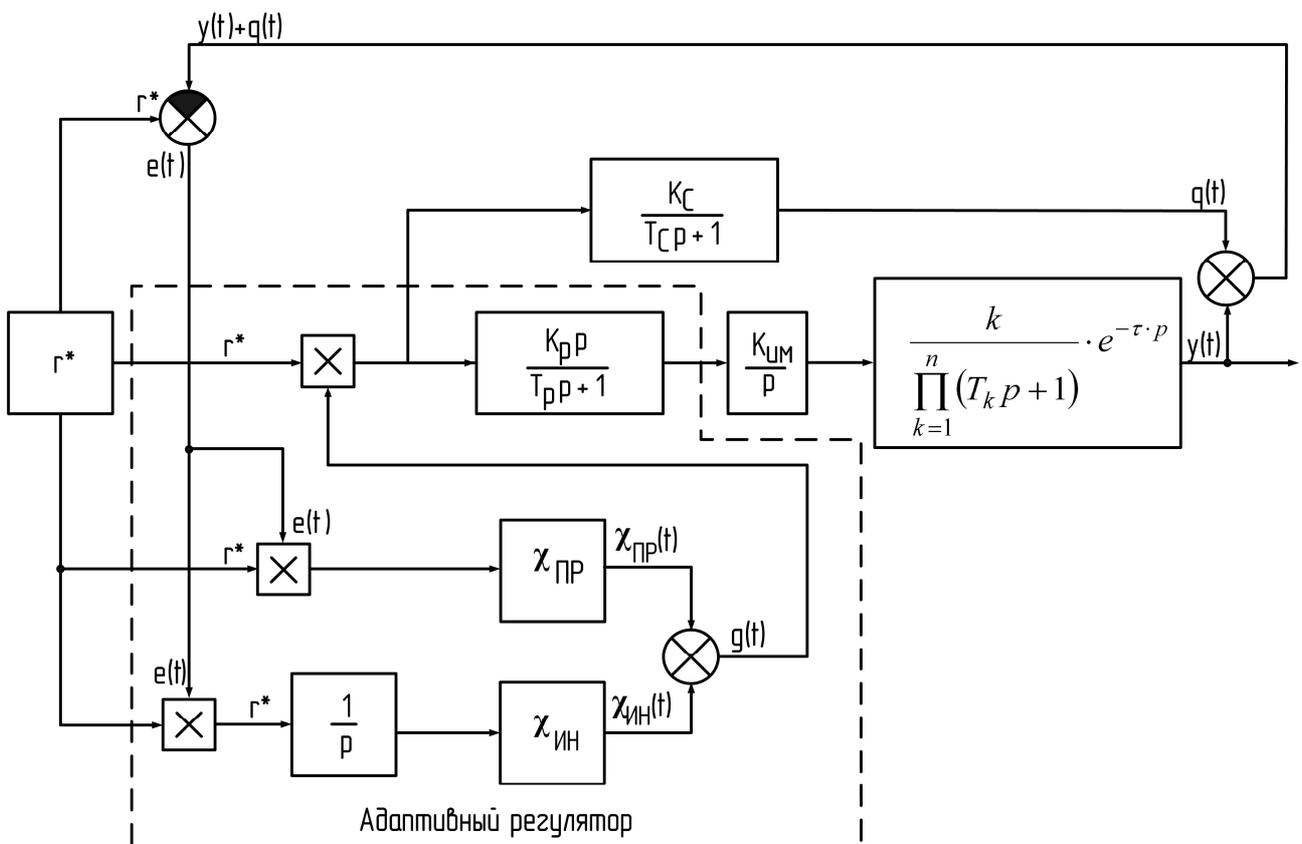


Рис. 4.2 Структура адаптивной системы со стабилизирующим устройством

Порядок работы

1. Построить предложенную на рис. 4.2 модель в Simulink.
2. При постоянном входном воздействии, настроить адаптивный регулятор, компенсатор.
3. Подключить ШИМ, оценить качество работы системы.
4. Проанализировать работу системы при использовании кусочно-постоянного воздействия (рис. 4.3), например:

$$r(t) = 0.143 \exp(-0.143 \cdot t) + H_1 + H_2 + H_3$$

$$H_1 = \begin{cases} 0 & \text{при } t \leq 0 \\ 1 & \text{при } t > 0 \end{cases}, H_2 = \begin{cases} 0 & \text{при } t \leq 100 \\ -0.5 & \text{при } t > 100 \end{cases}, H_3 = \begin{cases} 0 & \text{при } t \leq 200 \\ 1 & \text{при } t > 200 \end{cases} \quad (4.14)$$

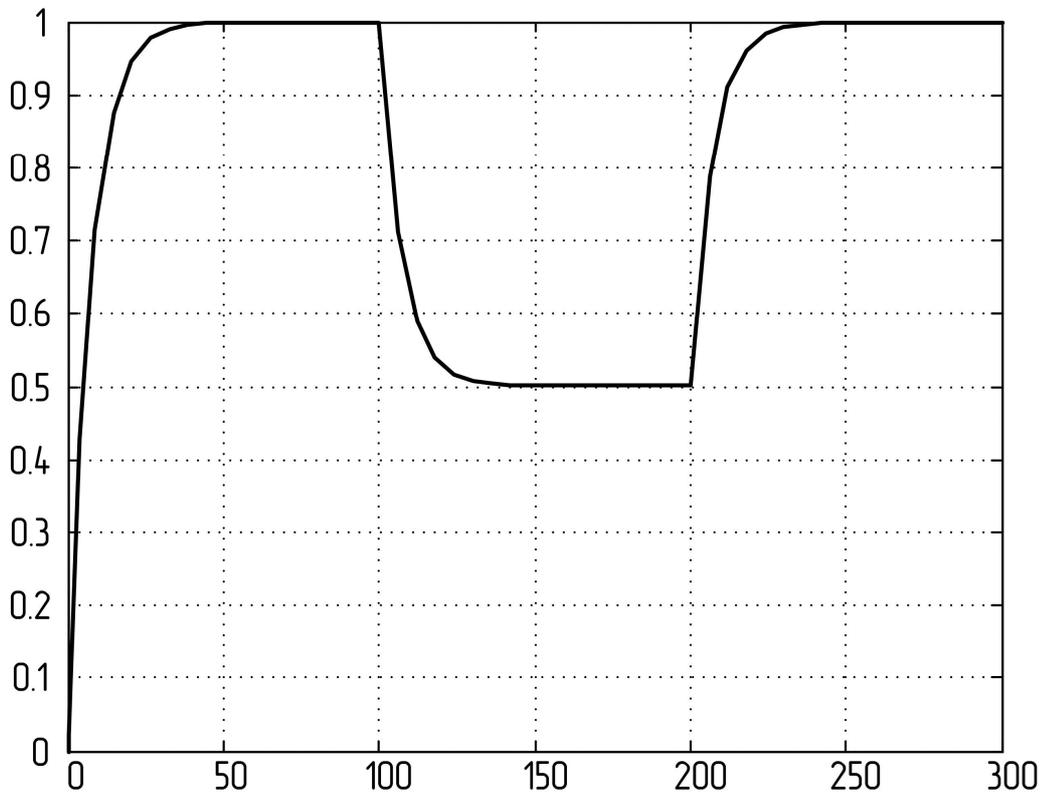


Рис. 4.3 Кусочно-постоянное воздействие

5. Огрубить алгоритмы адаптации.
5. Вызвать в объекте управления параметрические рассогласования. Исследовать границы множества параметров Ξ .

Содержание отчета

1. Simulink – модель адаптивной системы со стабилизирующим устройством. Настройки регулятора.
2. Переходные характеристики процессов, возникающие в системе с постоянным и кусочно-постоянным задающим воздействием.

Лабораторная работа № 5

Нечеткие системы управления

Цель работы:

1. Изучение принципов построения нечетких систем управления.
2. Сравнение алгоритмов нечеткого вывода. Создание базы правил.

Краткие теоретические сведения

1. Основные понятия нечеткой логики

Главным первичным понятием фаззи-логики является *лингвистическая переменная*. Лингвистическая переменная описывается пятью параметрами:

$$\langle x, T, U, G, M \rangle, \quad (5.1)$$

где x – имя переменной, T – терм-множество, каждый элемент которого задается нечетким множеством на универсальном множестве U , G – синтаксические правила, порождающие названия термов, M – семантические правила, задающие функции принадлежности нечетких термов, порожденных синтаксическими правилами из G .

Унифицированное терм-множество можно представить универсальным набором, состоящих из семи термов с треугольными и трапецидальными функциями принадлежности (рис. 5.1).

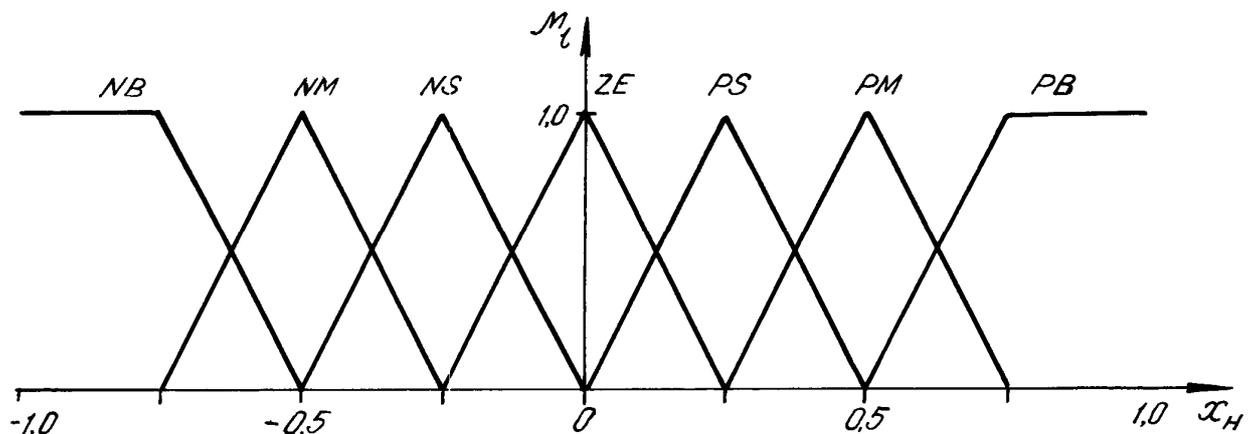


Рис. 5.1 Унифицированная система функций принадлежности

Наименование термов на рис. 5.1 образованы с помощью комбинаций слов «отрицательный» (англ. – negative), «положительный» (англ. – positive),

«большой» (англ. – big), «средний» (англ. – medium), «маленький» (англ.- small) и «приблизительно ноль» (англ. – approximately zero): negative big (*NB*), negative medium (*NM*), negative small (*NS*), approximately zero (*ZE*), positive small (*PS*), positive medium (*PM*), positive big (*PB*).

Нечеткое множество *A* является подмножеством базисного множества *U* и в общем виде может быть записано как совокупность пар различных значений *x* и соответствующих им значений функций принадлежности $\mu_A(x)$:

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad (5.2)$$

Процедуру определения значения функции принадлежности $\mu_A(x^*)$, соответствующего конкретному значению x^* переменной *x*, называют *фазификацией*.

Пусть задано нечеткое множество *A*, определяемое выражением (5.2), и нечеткое множество *B*, заданное следующим выражением:

$$B = \{(y, \mu_B(y)) \mid y \in Y\} \quad (5.3)$$

Фаззи-пересечение двух нечетких множеств *A* и *B*:

$$\mu_C(x, y) = \min\{\mu_A(x), \mu_B(y)\}, (x, y) \in A \times B. \quad (5.4)$$

Фаззи-объединение двух нечетких множеств *A* и *B*:

$$\mu_C(x, y) = \max\{\mu_A(x), \mu_B(y)\}, (x, y) \in A \times B. \quad (5.5)$$

Примеры фаззи – пересечения и фаззи-объединения показаны на рис. 5.2.

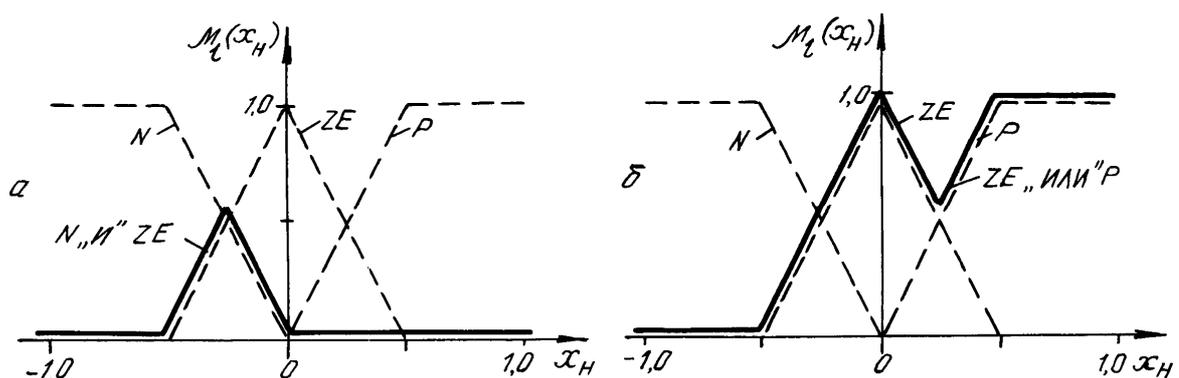


Рис. 5.2 Примеры фаззи – пересечения (а) и фаззи-объединения (б)

Нечеткий логический вывод основан на операции импликации. Импликация – логическая операция, заключающаяся в соединении двух нечетких высказаний

званий, например, A (посылка) и B (результат), в новое нечеткое высказывание

$$\langle \text{ЕСЛИ } x \text{ есть } A, \text{ ТО } y \text{ есть } B \rangle. \quad (5.6)$$

Импликация описывает взаимосвязь (*отношение*) между нечеткими множествами A и B .

Нечеткое отношение двух нечетких множеств, например, (5.2) и (5.3)

$$R = \{[(x, y), \mu_R(x, y)] \mid (x, y) \in A \times B\}, \quad (5.7)$$

где $\mu_R(x, y)$ – функция принадлежности, характеризующая одновременно принадлежность обоих элементов x и y , образующих пары (x, y) , к декартовому произведению $U \times Y$.

Функция принадлежности нечеткого отношения $\mu_R(x, y)$ определяется по формуле фаззи – пересечения двух нечетких множеств A и B , т.е. с помощью процедуры минимизации:

$$\mu_R(x, y) = \min_{A \times B} \{\mu_A(x), \mu_B(y)\} \quad (5.8)$$

Из (5.6) определяется функции принадлежности выходной переменной

$$\mu_B(y) = \max_{x \in A} \{\min[\mu_A(x), \mu_R(x, y)]\}, y \in B \quad (5.9)$$

Или при заданном значении $x = x^*$:

$$\mu_{B^*}(y) = \min_{x \in A} \{\mu_A(x^*), \mu_B(y)\} \quad (5.10)$$

Посылка правила (5.3) может состоять из нескольких высказываний, которые соединены между собой логическими операциями «И» и «ИЛИ». Тогда, например, для правила

$$\langle \text{ЕСЛИ } x_1 \text{ есть } A_1 \text{ И } x_2 \text{ есть } A_2 \text{ И } x_3 \text{ есть } A_3, \text{ ТО } y \text{ есть } B \rangle \quad (5.11)$$

нечеткое отношение запишется аналогично (5.7) с добавлением двух входных переменных x_1 и x_2 , а функция принадлежности этого нечеткого отношения задается выражением:

$$\mu_R(x_1, x_2, x_3, y) = \min_{A \times B} \{ \min_{A_1, A_2, A_3} [\mu_{A_1}(x_1), \mu_{A_2}(x_2), \mu_{A_3}(x_3)], \mu_B(y) \}. \quad (5.12)$$

Из выражения (5.13) видно, что логическая операция «И» заменяется процедурой минимизации (фаззи-пересечением).

В случае применения к отдельным высказываниям операции «ИЛИ», применяется процедура максимизации (фаззи-объединения) функций принадлежности этих высказываний.

Как правило, любая систему или объект описывается несколькими правилами нечеткого логического вывода. В этом случае формируется *нечеткая база знаний*, состоящая из j правил, $j=1\dots n$, которые связаны между собой союзом «ИЛИ».

2. Алгоритм нечеткого вывода Мамдани

Объединенная процедура определения функции принадлежности нечеткого правила «ЕСЛИ – ТО» по формуле (5.8) и агрегирования нескольких таких правил, связанных союзом «ИЛИ», называется *инференц-процедурой*, или *инференцией*.

Агрегирование, или композиция нескольких правил «ЕСЛИ – ТО», связанных союзом «ИЛИ», осуществляется путем максимизации функций принадлежности всех объединяемых правил согласно формуле (5.12).

Таким образом, результирующая функция принадлежности управляющего воздействия формируется по закону максимина:

$$\mu_{B^*}(y) = \max_j \{ \mu_{B_j^*}(y) \} = \max_j \{ \min[\mu_{A_j}(x^*), \mu_{B_j}(y)] \} \quad (5.13)$$

где $j=1, 2 \dots n$ – индекс j -го правила; n – количество объединяемых правил; $\mu_{A_j}(x^*)$ – значение функции принадлежности посылки A j -го правила, соответствующее фиксированному значению x^* входной переменной; $\mu_{B_j}(y)$ – функция принадлежности заключения B j -го правила; $\mu_{B_j^*}(y)$ – «усеченная» ФП заключения B j -го правила, ограниченная сверху на уровне $\mu_{A_j}(x^*)$.

Для того, чтобы по полученной таким образом результирующей функции принадлежности (5.13) найти конкретное значение выходной переменной y , применяют процедуру. При дефазификации используются различные методы. Наибольшее применение получил метод *центра тяжести*, согласно которому

значение y^* выходной переменной находят как абсциссу «центра тяжести» площади, расположенной под графиком функции принадлежности $\mu_{B^*}(y)$, по формуле средневзвешенного значения

$$y^* = \frac{\int y \cdot \mu_{B^*}(y) dy}{\int \mu_{B^*}(y) dy} \quad (5.14)$$

3. Алгоритм нечеткого вывода Сугено

Согласно концепции Сугено и его соавторов Такаги и Канга, заключение каждого j -го правила формируется не в виде лингвистических переменных, а записывается в виде алгебраической функции входных величин x_1, x_2, \dots, x_m :

$$y_j = c_{0j} + c_{1j}x_1 + c_{2j}x_2 + \dots + c_{mj}x_m, \quad (5.15)$$

где $c_{0j}, c_{1j}, \dots, c_{mj}$ – постоянные коэффициенты, устанавливаемые экспертами. Тем самым исключается необходимость обычной дефазификации, а четкое значение выходной переменной, соответствует фиксированным значениям x_i^* входных переменных x_i , вычисляется как средневзвешенная величина:

$$y^* = \frac{\sum_{j=1}^n y_j \cdot \mu_{A_j^*}}{\sum_{j=1}^n \mu_{A_j^*}}, \quad (5.16)$$

где $\mu_{A_j^*}$ – значение функции принадлежности всей посылки j -го правила, соответствующее значениям x_i^* .

Объединение составных частей посылок осуществляется так же, как и в методе Мамдани, с помощью процедур минимизации (5.11) и максимизации (5.12) соответствующих функций принадлежности.

Формула (5.15) соответствует алгоритму Сугено первого порядка. Часто применяют алгоритм Сугено нулевого порядка, согласно которому выходная переменная для j -го правила вычисляется по формуле

$$y_j = c_{0j} + c_{1j} + c_{2j} + \dots + c_{mj}, \quad (5.17)$$

где $c_{0j}, c_{1j}, \dots, c_{mj}$ – четкие значения выходной переменной, заданные в виде сингلتонов при формулировке исходных лингвистических правил.

4. Структура нечеткого регулятора

Типовая структура нечеткого регулятора, представленная на рис. 5.3, содержит следующие модули:

Фаззификатор, преобразующий фиксированный вектор входных параметров регулятора ε в вектор нечетких множеств, необходимых для нечеткого вывода;

Нечеткая база знаний, содержащая совокупную информацию о зависимостях между входами и выходами регулятора $u=f(z)$ в виде лингвистических правил, форма записи нечетких правил:

ЕСЛИ <посылка правила>, **ТО** <заключение правила>.

Функции принадлежности, используемые для представления лингвистических термов в виде нечетких множеств;

Машина нечеткого логического вывода, которая на основе правил базы знаний определяет значение выходной переменной в виде нечеткого множества \tilde{u} , соответствующего нечетким значениям входных переменных $\tilde{\varepsilon}$, используя при этом некоторые алгоритмы нечеткого логического вывода, которые рассмотрены выше;

Дефаззификатор, преобразующий выходное нечеткое множество \tilde{u} в четкое число u .

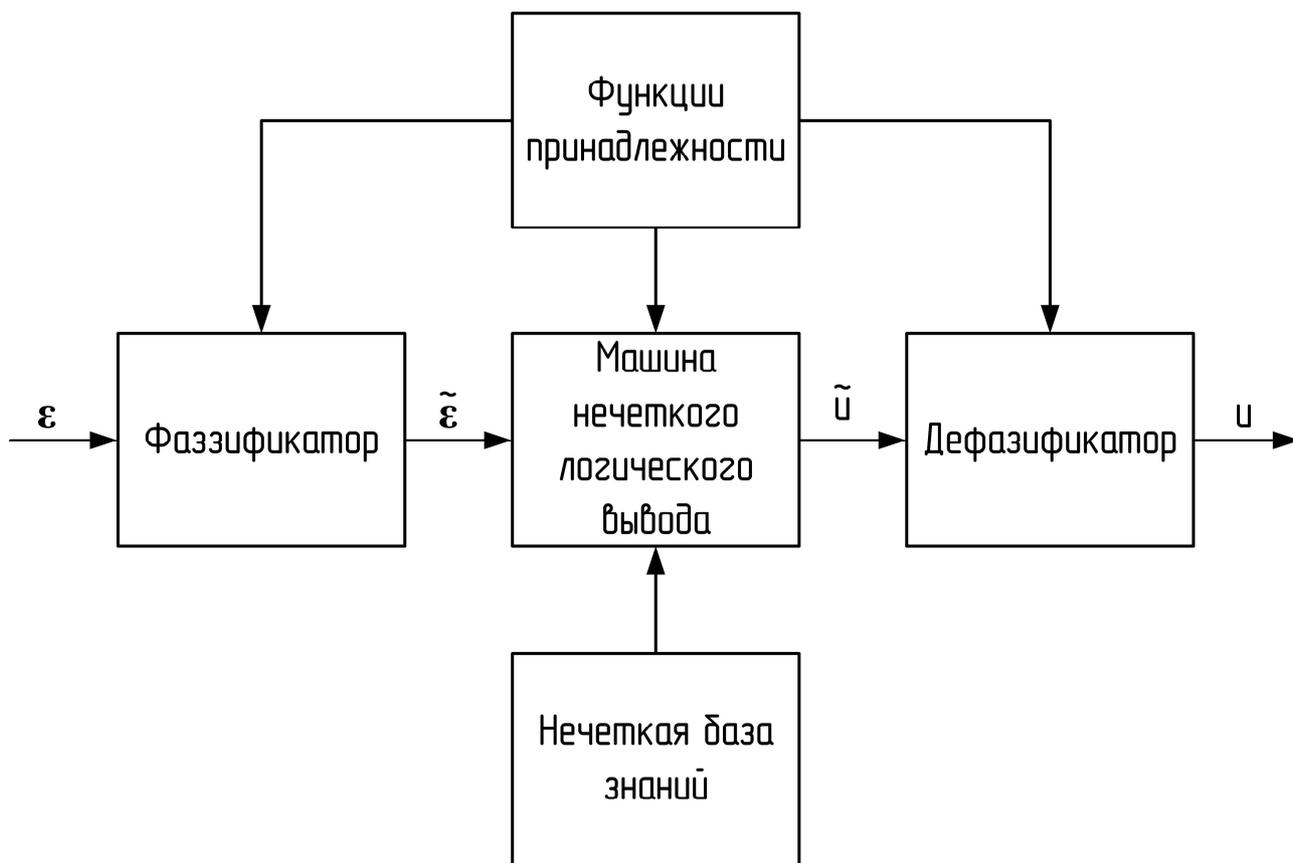


Рис. 5.3 Структура нечеткого регулятора

5 Порядок синтеза фаззи – регулятора

Несмотря на большое разнообразие подходов к построению систем фаззи-управления и на преобладание эвристических приемов алгоритмизации задач фаззи-управления, можно, тем не менее, выделить следующие обычно выполняемые процедуры синтеза фаззи-регулятора [5].

1. Предварительно изучают объект регулирования, его традиционную модель и опыт технологов по его эксплуатации.

2. Выбирают измеряемые (наблюдаемые) выходные переменные x_i и управляющие воздействия u_j , устанавливают их пределы изменения и подбирают масштабные коэффициенты, обеспечивающие нормирование всех переменных в определенном безразмерном диапазоне (например, $[-1, 1]$).

3. Вводят лингвистические переменные x_ℓ и u_ℓ и представляют их с помощью нескольких функций принадлежности, соответствующих отдельным лингвистическим значениям (термам). При этом весьма важным для обеспечения

качества и эффективности управления является выбор количества, формы и взаимного расположения функций принадлежности на нормированном интервале.

4. На основе интервью и анкетирования экспертов (операторов-технологов, инженеров-конструкторов объекта) формулируют логические правила (алгоритмы) регулирования. При этом полезно консультирование экспертов по объекту экспертом по фаззи-технологии. В случае необходимости корректируют выбранные в пункте 3 функции принадлежности.

Правила с одинаковым заключением с помощью союза «ИЛИ» объединяют в одно правило.

5. Сформулированные правила проверяют на их полноту, непротиворечивость и избыточность.

6. В случае двух входных переменных x_1 и x_2 составляют таблицы лингвистических правил, устанавливающие соответствие между лингвистическими значениями входных и выходных переменных.

7. Выбирают операторы импликации, инференц-процедуры и метод дефазификации.

8. С использованием специализированного пакета (например, «Fuzzy Logic», имеющегося в составе программной системы «MATLAB») программируют все функции принадлежности, правила, процедуры агрегирования посылок отдельных правил и всех правил между собой.

9. Проводят имитационное моделирование и испытание синтезированной стратегии управления в разомкнутом виде и в соединении с моделью объекта (четкой или нечеткой) и оценивают работоспособность, устойчивость и качества системы с фаззи-регулятором.

6. Иллюстрация процедур фаззи-логики

Ставится задача выработки управляющего воздействия u по двум входным сигналам. Входные переменные – ошибка регулирования $\varepsilon = \varepsilon_1$ и ее первая производная по времени $\dot{\varepsilon} = \varepsilon_2$ – характеризуются тремя функциями принадлеж-

ности (рис. 5.4, 5.5), а алгоритм нечеткого управления представлен следующими четырьмя правилами, соединенными между собой союзом «ИЛИ»:

- Правило 1: **ЕСЛИ** $\varepsilon_1 = N$ **И** $\varepsilon_2 = P$, **ТО** $u = P$,
 Правило 2: **ЕСЛИ** $\varepsilon_1 = N$ **И** $\varepsilon_2 = ZE$, **ТО** $u = ZE$, (5.18)
 Правило 3: **ЕСЛИ** $\varepsilon_1 = ZE$ **И** $\varepsilon_2 = P$, **ТО** $u = P$,
 Правило 4: **ЕСЛИ** $\varepsilon_1 = ZE$ **ИЛИ** $\varepsilon_2 = ZE$, **ТО** $u = ZE$.

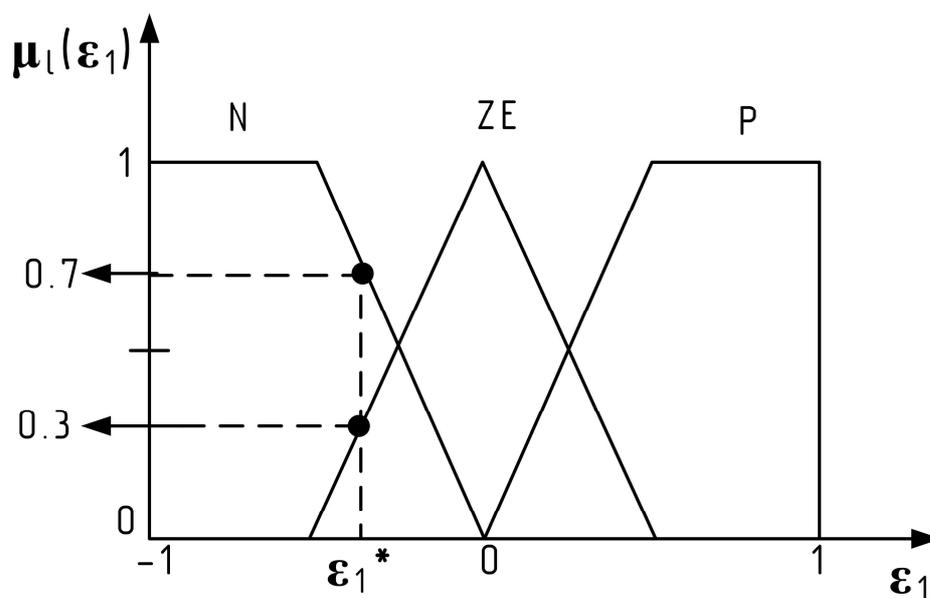


Рис. 5.4 Функции принадлежности, характеризующие термы ошибки регулирования ε_1

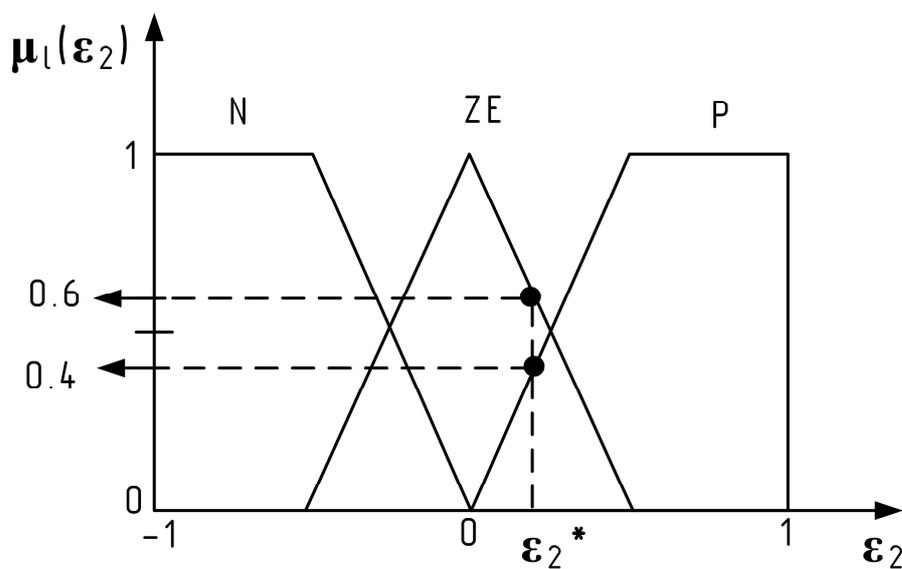


Рис. 5.5 Функции принадлежности, характеризующие термы производной ошибки регулирования ε_2

Пусть входные переменные имеют конкретные значения: $\varepsilon_1 = \varepsilon_1^* = -0,40$ и $\varepsilon_2 = \varepsilon_2^* = +0,23$. Тогда согласно формуле (5.4) посылки «ЕСЛИ» первых трех правил (5.18), содержащие внутри себя союз «И», дают следующие значения соответствующих фаззи – пересечений:

$$\begin{aligned}\mu_{A_1}(\varepsilon_1^*, \varepsilon_2^*) &= \min \{ \mu_N(\varepsilon_1^*), \mu_P(\varepsilon_2^*) \} = \min \{ 0,7; 0,4 \} = 0,4; \\ \mu_{A_2}(\varepsilon_1^*, \varepsilon_2^*) &= \min \{ \mu_N(\varepsilon_1^*), \mu_{ZE}(\varepsilon_2^*) \} = \min \{ 0,7; 0,6 \} = 0,6; \\ \mu_{A_3}(\varepsilon_1^*, \varepsilon_2^*) &= \min \{ \mu_{ZE}(\varepsilon_1^*), \mu_P(\varepsilon_2^*) \} = \min \{ 0,3; 0,4 \} = 0,3.\end{aligned}\tag{5.19}$$

Посылка в правиле 4 содержит союз «ИЛИ», и поэтому согласно формуле (5.5) функция принадлежности фаззи-объединения дает значение:

$$\mu_{A_4}(\varepsilon_1^*, \varepsilon_2^*) = \max \{ \mu_{ZE}(\varepsilon_1^*), \mu_{ZE}(\varepsilon_2^*) \} = \max \{ 0,3; 0,6 \} = 0,6.\tag{5.20}$$

Теперь в соответствии с алгоритмом (5.10) можно определить результирующие функции принадлежности, характеризующие правила (5.18):

$$\begin{aligned}\mu_{B_1^*}(u) &= \min \{ \mu_{A_1}(\varepsilon_1^*, \varepsilon_2^*), \mu_P(u) \}; \\ \mu_{B_2^*}(u) &= \min \{ \mu_{A_2}(\varepsilon_1^*, \varepsilon_2^*), \mu_{ZE}(u) \}; \\ \mu_{B_3^*}(u) &= \min \{ \mu_{A_3}(\varepsilon_1^*, \varepsilon_2^*), \mu_P(u) \}; \\ \mu_{B_4^*}(u) &= \min \{ \mu_{A_4}(\varepsilon_1^*, \varepsilon_2^*), \mu_{ZE}(u) \}.\end{aligned}\tag{5.21}$$

«Усеченные» функции принадлежности, соответствующие выражениям (5.21) и конкретным значениям функций принадлежности (5.19) и (5.20), приведены на рис. 5.5 и 5.6.

Выполняя для рассматриваемого примера процедуру агрегирования функций $\mu_{B_j^*}(u)$ по формуле (5.13), получаем результирующую ФП $\mu_{B^*}(u)$ (рис. 5.6), соответствующую агрегируемым функциям на рис.5.5. На рис. 5.6 показано также результирующее значение u^* управляющего воздействия u , найденное в результате процедуры дефазификации по методу центра тяжести площади – по формуле (5.14).

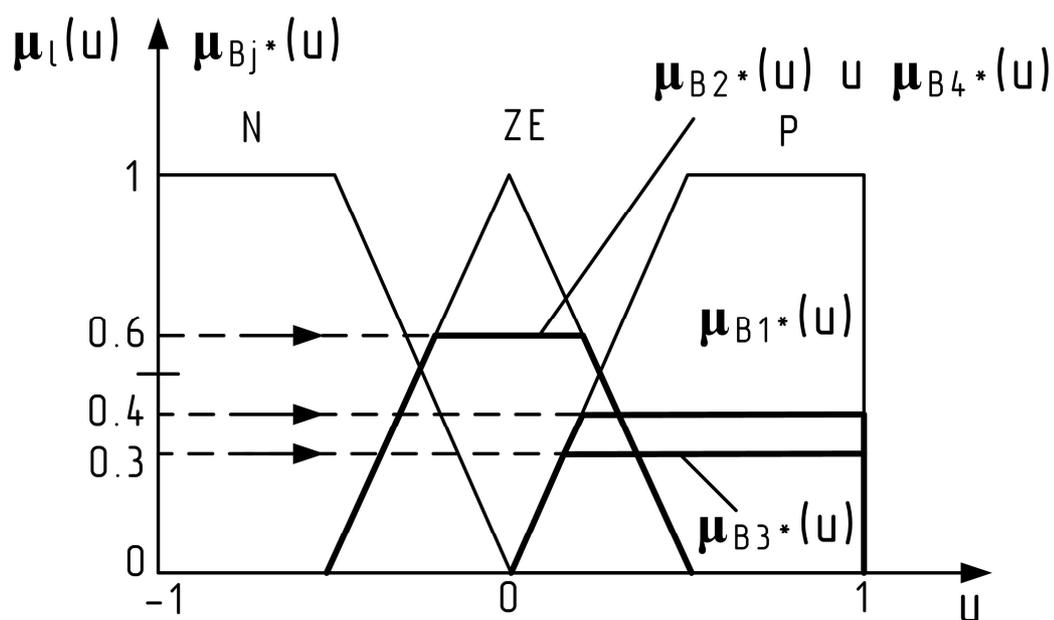


Рис. 5.5. Функции принадлежности заключений отдельных правил нечеткого алгоритма регулирования (5.18)

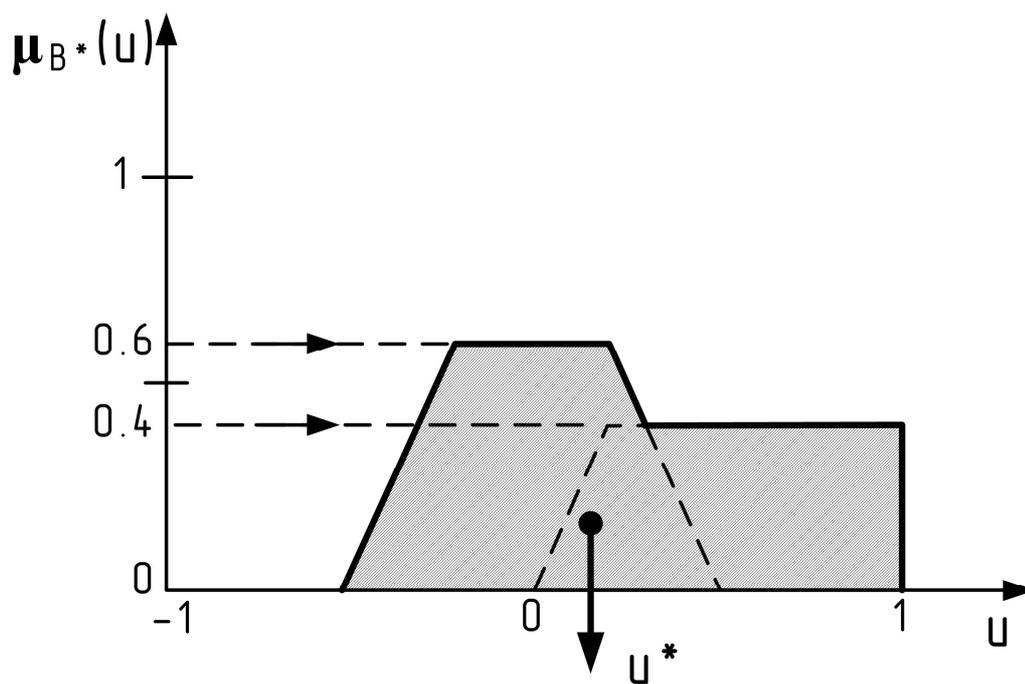


Рис. 5.6. Функции принадлежности заключений объединенных правил нечеткого алгоритма регулирования (5.18)

7. Системы регулирования уровня жидкости в резервуаре

В качестве примера [18] рассматривается задача регулирования уровня в подогревателе низкого давления (ПНД) турбоагрегата ТЭЦ (турбина Т-110/120-

130-4), работающего в общей паровой магистрали. Для получения математического описания объекта было снято семейство разгонных характеристик. Одна из полученных (для средней нагрузки турбоагрегата и при установившихся параметрах всех участвующих в работе котлоагрегатов) представлена на рис. 5.7 – слева – изменение уровня (в мм), а справа – вызвавшее это изменение отклонение управляющего клапана установленного на сливе (в процентах).

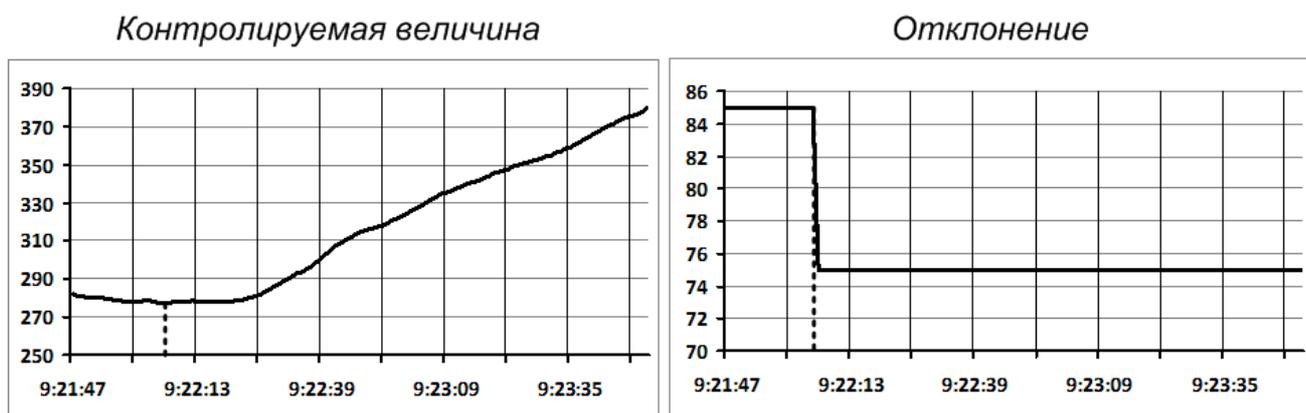


Рис. 5.7 Разгонная характеристика по уровню в ПНД №2

Передаточная функция объекта совместно с исполнительным механизмом, полученная для базового режима имеет вид:

$$W_{Oy}(s) = \frac{y(s)}{u(s)} = \frac{K_{oy} \cdot e^{-h \cdot s}}{a_1 s + 1} \cdot \frac{1}{s^2} = \frac{0.005 \cdot e^{-16 \cdot s}}{7 \cdot s + 1} \cdot \frac{1}{s^2}. \quad (5.22)$$

Структура системы, реализованная в Simulink, представлена на рис. 5.8. Нечеткий регулятор представляет собой звено с двумя входами и одним выходом. На вход регулятора заводится ошибка регулирования и ее производная, на выходе регулятора формируется четкое управляющее воздействие. Алгоритм работы нечеткого регулятора следующий:

1. **ЕСЛИ** ошибка нормальная, **ТО** положение задвижки не изменять.
2. **ЕСЛИ** ошибка низкая, **ТО** задвижку закрывать быстро.
3. **ЕСЛИ** ошибка высокий, **ТО** задвижку открывать быстро.
4. **ЕСЛИ** ошибка нормальная **И** она увеличивается (производная ошибки положительна), **ТО** задвижку открывать медленно.
5. **ЕСЛИ** ошибка нормальная **И** она уменьшается (производная ошибки отрицательна), **ТО** задвижку открывать быстро.

Для формирования результирующей функции принадлежности используется алгоритм нечеткого вывода типа Мамдани. Четкое управляющее воздействие формируется по методу центров тяжести.

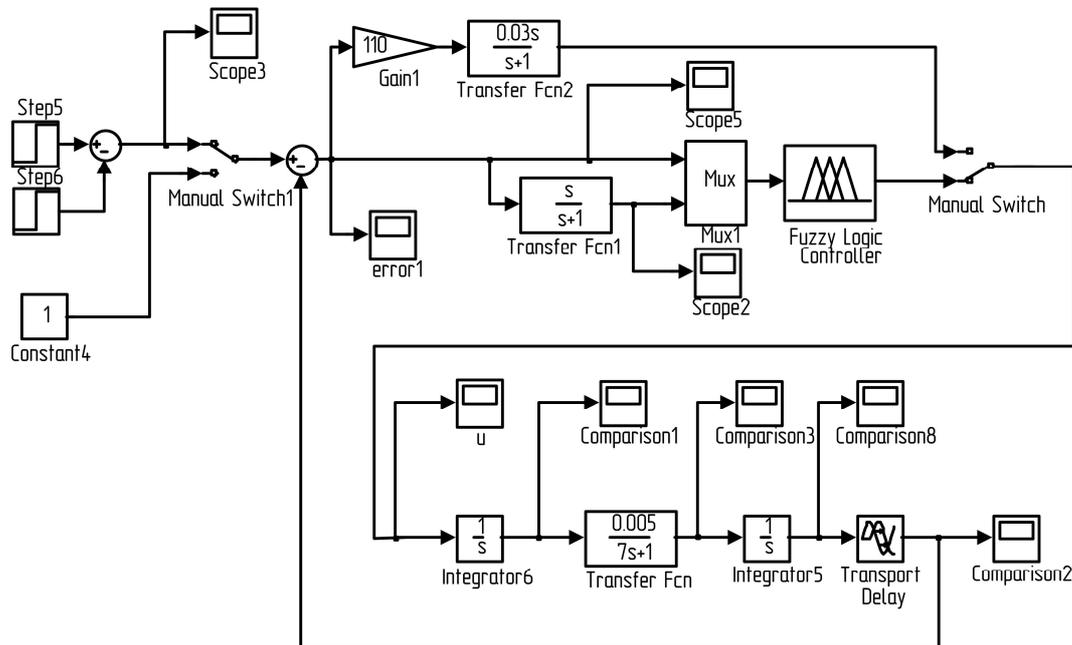


Рис. 5.8 Simulink – модель системы с нечетким регулятором

Функции принадлежности входных и выходной величин регулятора [6], границы их изменения показаны на рис. 5.9 – 5.11.

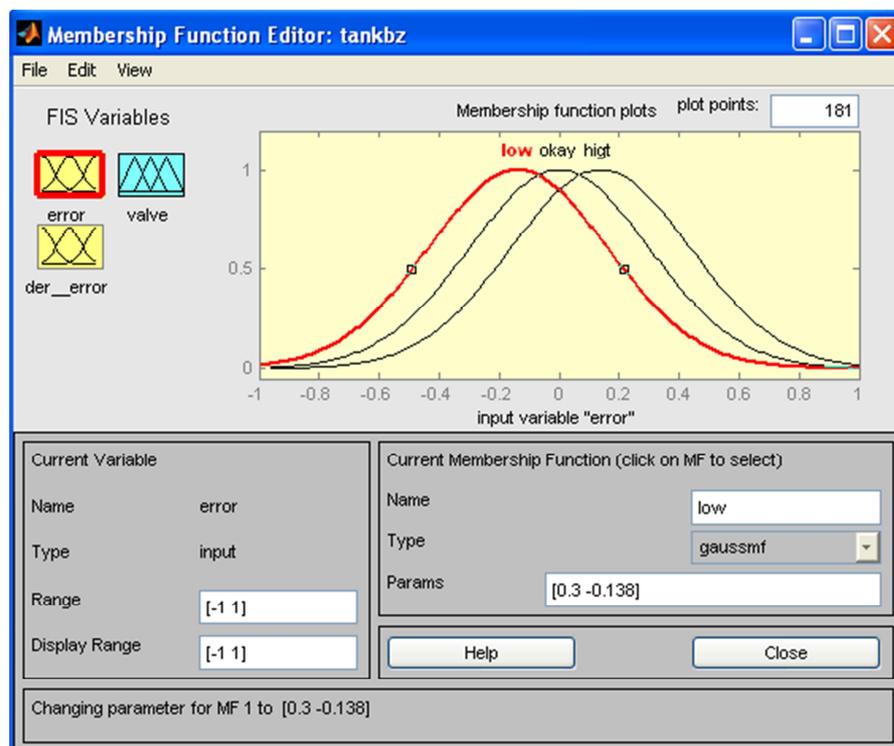


Рис. 5.9 Функции принадлежности для ошибки регулирования

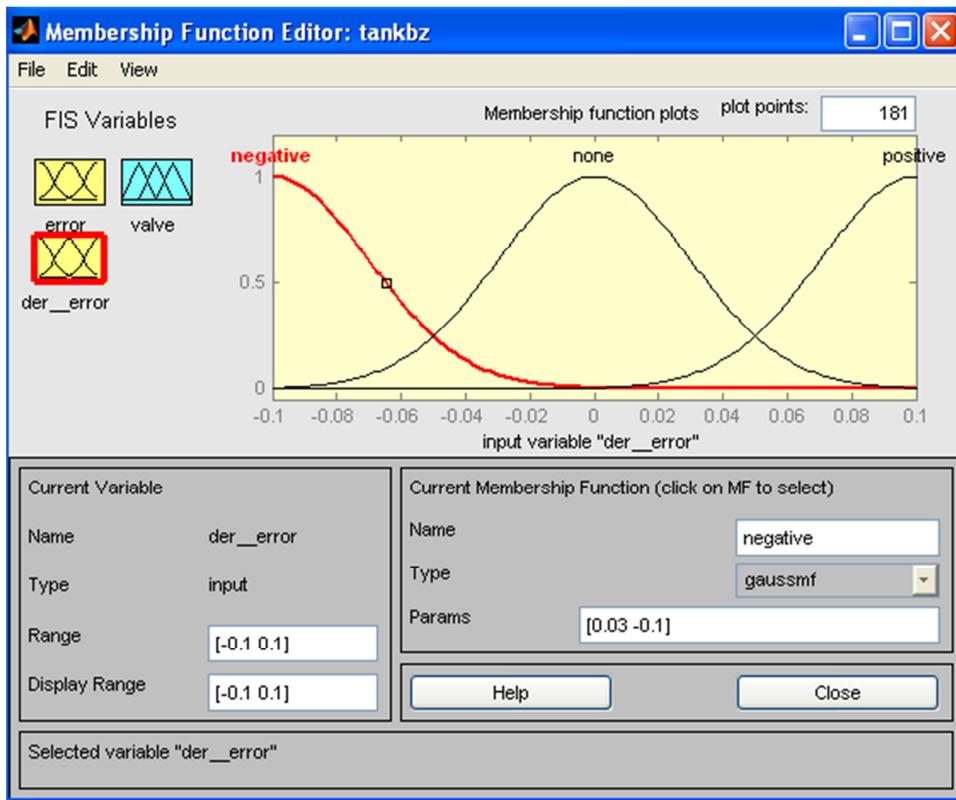


Рис. 5.10 Функции принадлежности для производной ошибки регулирования

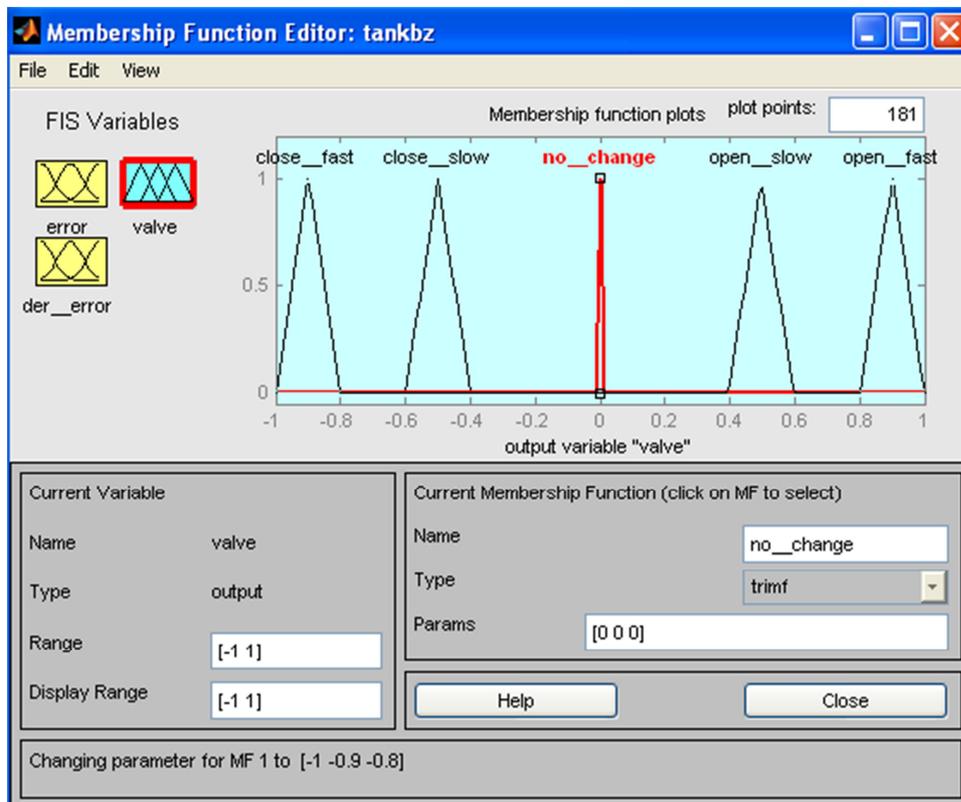


Рис. 5.11 Функции принадлежности для перемещения задвижки

Переходные процессы в системе на рис. 5.8 показаны на рис. 5.12.

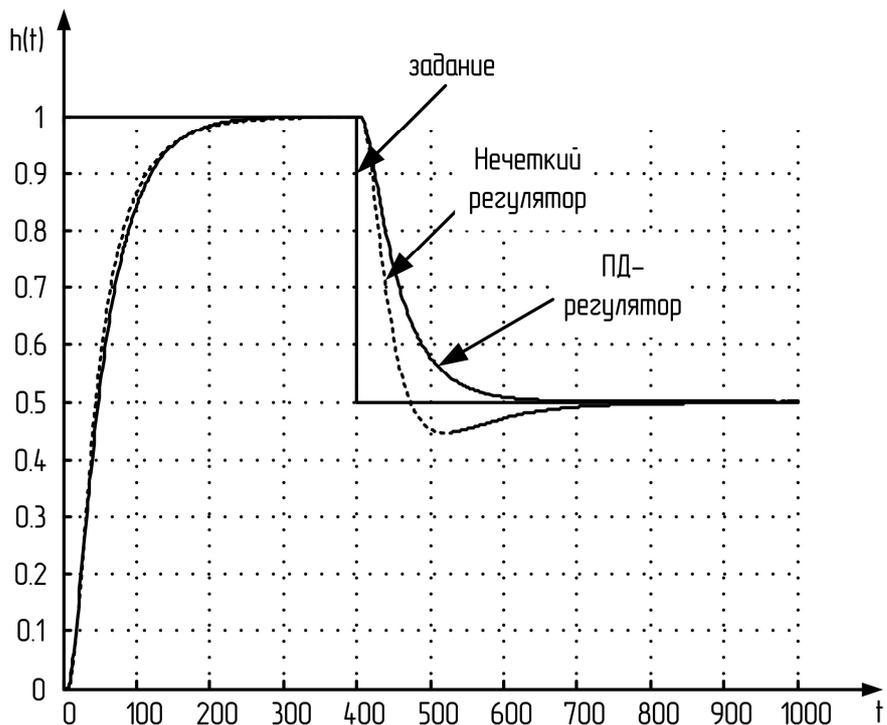


Рис. 5.12 Переходные процессы в системе на рис. 5.8

8. Недостатки систем с нечеткой логикой

Вообще говоря [15], системы с нечеткой логикой целесообразно применять для сложных процессов, когда нет простой математической модели; если экспертные знания об объекте или о процессе можно сформулировать только в лингвистической форме.

Данные системы применять нецелесообразно, когда требуемый результат может быть получен каким-либо другим (стандартным) путем, или когда для объекта или процесса уже найдена адекватная и легко исследуемая математическая модель.

Основные недостатки систем с нечеткой логикой связаны с тем, что:

- исходный набор постулируемых нечетких правил формулируется экспертом человеком и может оказаться неполным или противоречивым;
- вид и параметры функций принадлежности, описывающих входные и выходные переменные системы, выбираются субъективно и могут оказаться не вполне отражающими реальную действительность.

Порядок работы

1. Построить Simulink-модель приведенного в лабораторной работе примера системы регулирования жидкостью в резервуаре.
2. Параметры объекта управления взять согласно варианту.
2. Подобрать настройки ПД – регулятора.
3. Реализовать систему управления с нечетким регулятором.
4. Сравнить работу систем.

Содержание отчета

1. Simulink-модель нечеткой системы и системы с ПД – регулятором.
2. Переходные характеристики в системе.

Лабораторная работа №6

Реализация систем управления на микроконтроллерах

Цель работы:

1. Изучение основных этапов перехода от имитационной модели в Simulink к проектированию реальной системы.
2. Изучение принципов построения алгоритмических схем на микроконтроллерах.

Краткие теоретические сведения

1. Дискретно-непрерывное представление системы регулирования

Современные системы автоматического регулирования технологических процессов строятся на базе микропроцессорной техники. Их принципиальное отличие от традиционных аналоговых заключается в том, что они являются цифровыми (дискретными). Структура такой системы регулирования, состоящей из объекта управления, цифрового регулятора, а также блоков цифро-аналогового (ЦАП) и аналого-цифрового (АЦП) преобразования, изображена на рис. 6.1.

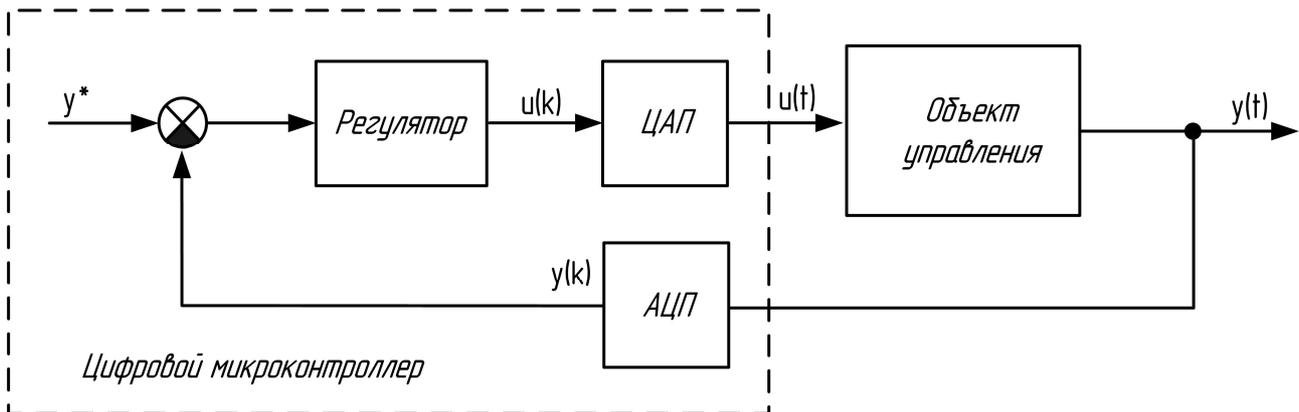


Рис. 6.1 Структурная схема дискретно-непрерывной системы регулирования

Из структурной схемы видно, что непрерывный во времени процесс $y(t)$ преобразуется в дискретный ряд чисел $y(k)$, т.е. квантуется по уровню и по времени.

После преобразования числовой информации в цифровом микроконтроллере согласно выбранному закону регулирования, совершается обратный процесс преобразования цифровой информации в аналоговую и на выходе регулятора формируется непрерывное во времени регулирующее воздействие $u(t)$.

Объект управления в совокупности с исполнительным механизмом остается непрерывным, а контур регулирования преобразуется в дискретный вариант с определенным значением времени дискретизации.

В основе математического анализа таких систем лежит дискретное преобразование Лапласа, либо Z – преобразование [16].

Для упрощения процедуры дискретизации управляющего контура рекомендуется воспользоваться встроенными средствами среды Matlab, например, функцией, которая переводит систему из непрерывной формы в дискретную:

$$SYSD = C2D (SYSC, Ts, METHOD), \quad (6.1)$$

где $SYSC$ – система, заданная в пространстве состояний или в виде передаточной функции, Ts – шаг дискретизации, $METHOD$ - метод Z – преобразования.

В процессе сборки исследуемой системы в Simulink в качестве аналого-цифрового преобразователя используется блок Zero-Order Hold, единственным параметром которого является шаг (время) дискретизации.

Роль цифро-аналогового преобразователя выполняет блок широтно-импульсной модуляции. (непрерывный элемент системы сглаживает цифровой сигнал, в данном случае это исполнительный механизм).

Например, дискретизацию регулятора и стабилизирующего устройства, заданных в виде передаточных функций, можно выполнить, как показано в листинге на рис. 6.2

```

%Регулятор
Kp = 450;
Tp = 100 ;
Wr = tf([Kp 0],[Tp 1]);
%Компенсатор
Kc= 0.03;
Tc=100 ;
Wc= tf([0 Kc],[Tc 1]);
%Дискретная модель
Wrd = c2d(Wr, tay,'zoh');
Wcd = c2d(Wc,tay,'zoh');
num = Wrd.num; num_r=num{1};
den = Wrd.den; den_r=den{1};
num = Wcd.num; num_c=num{1};
den = Wcd.den; den_c=den{1};

```

Рис.6.2 Листинг программы дискретизации непрерывных моделей

Результат выполнения программы (6.2) следующий:

$$W_P = \frac{4,5z - 4,5}{z - 0,9966} \quad (6.2)$$

$$W_P = \frac{7,191 \cdot 10^{-5}}{z - 0,9966} \quad (6.3)$$

2. «Физическая» часть системы управления

При реализации системы управления на *конкретном* цифровом микроконтроллере необходимо определиться со схемой подключения измерительных и исполнительных устройств.

Примером может служить система управления температуры за калорифером котла, изображенная на рис.6.3 [21].

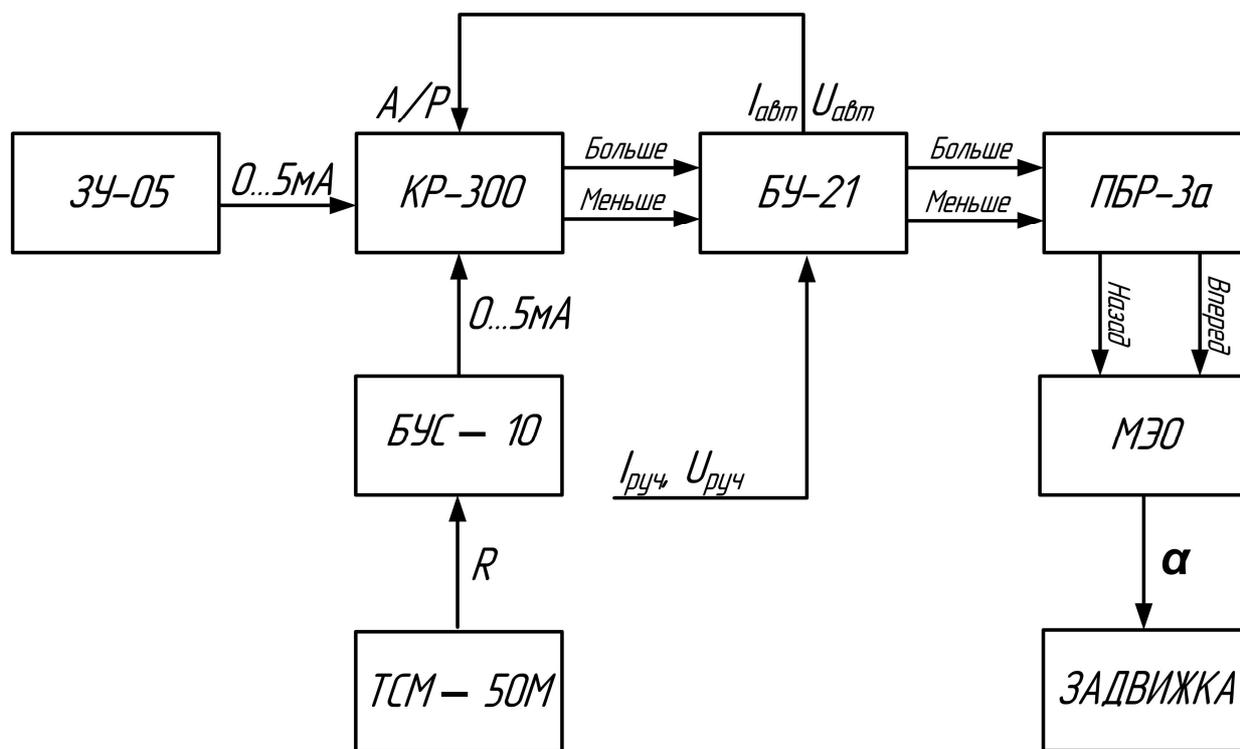


Рис.6.3 Структурная схема электрических соединений

Данная система реализована на микроконтроллере КР-300. Контролируемая величина – температура – измеряется термометром сопротивления, преобразуется в унифицированный электрический сигнал и усиливается блоком усиления сопротивлений. В регуляторе этот сигнал сравнивается с задающим сигналом, вырабатываемым задающим устройством. Далее регулятор вырабатывает необходимое управляющее воздействие и через блок управления передается сигнал на пускатель, который управляет механизмом постоянной скорости. Механизм постоянной скорости приводит в движение задвижку, изменение положения которой влияет на контролируемую величину.

Принципиальная электрическая схема, соответствующая структурной схеме, приведена на рис. 6.4 и 6.5.

Компоненты электрической схемы.

1. Блок усиления сопротивлений БУС-10 используется для преобразования изменение сопротивления резистивных датчиков в токовый сигнал 0-5 мА.
2. В качестве задающего токового устройства применяется ЗУ-05.

3. Блок управления БУ-21 обеспечивает управление системой в ручном и автоматическом режимах.

При переходе в ручной режим размыкается контакт автоматического регулирования, и контроллер отслеживает работу БУ-21 для последующего безударного переключения из ручного в автоматический режим.

4. Исполнительный механизм используется марки МЭО 100/25-0,25-99к.

Для регистрации самых крайних положений применяются 2 конечных выключателя, при достижении которых механизм должен быть немедленно отключен. Для обеспечения лучшей надежности введены два промежуточных контакта, которые находятся в замкнутом состоянии, пока механизм осуществляет движение в пределах заданной траектории. При размыкании этих контактов сообщается об аварийной ситуации и следует останов механизма.

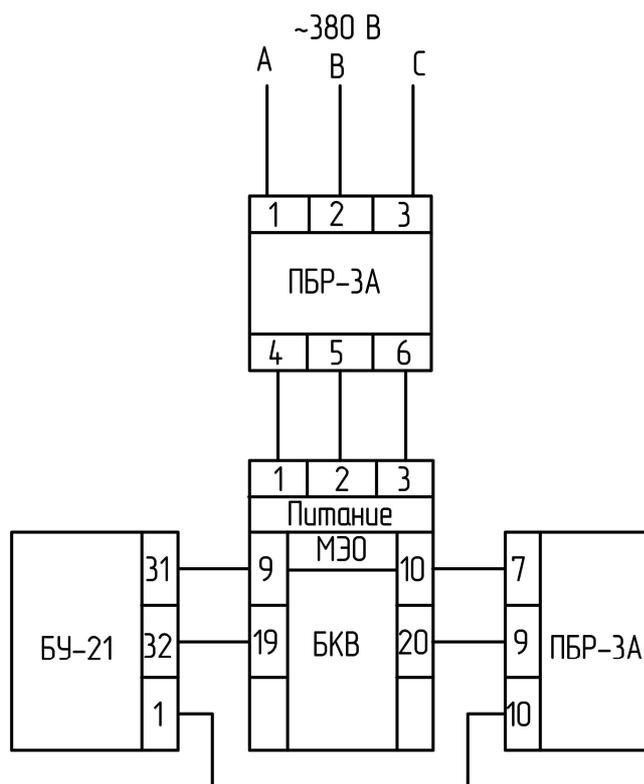


Рис. 6.4 Организация ручного управления

5. Пускатель предназначен для бесконтактного управления электрическими исполнительными механизмами.

6. Датчик температуры – термопреобразователь сопротивления модели TSM-50M

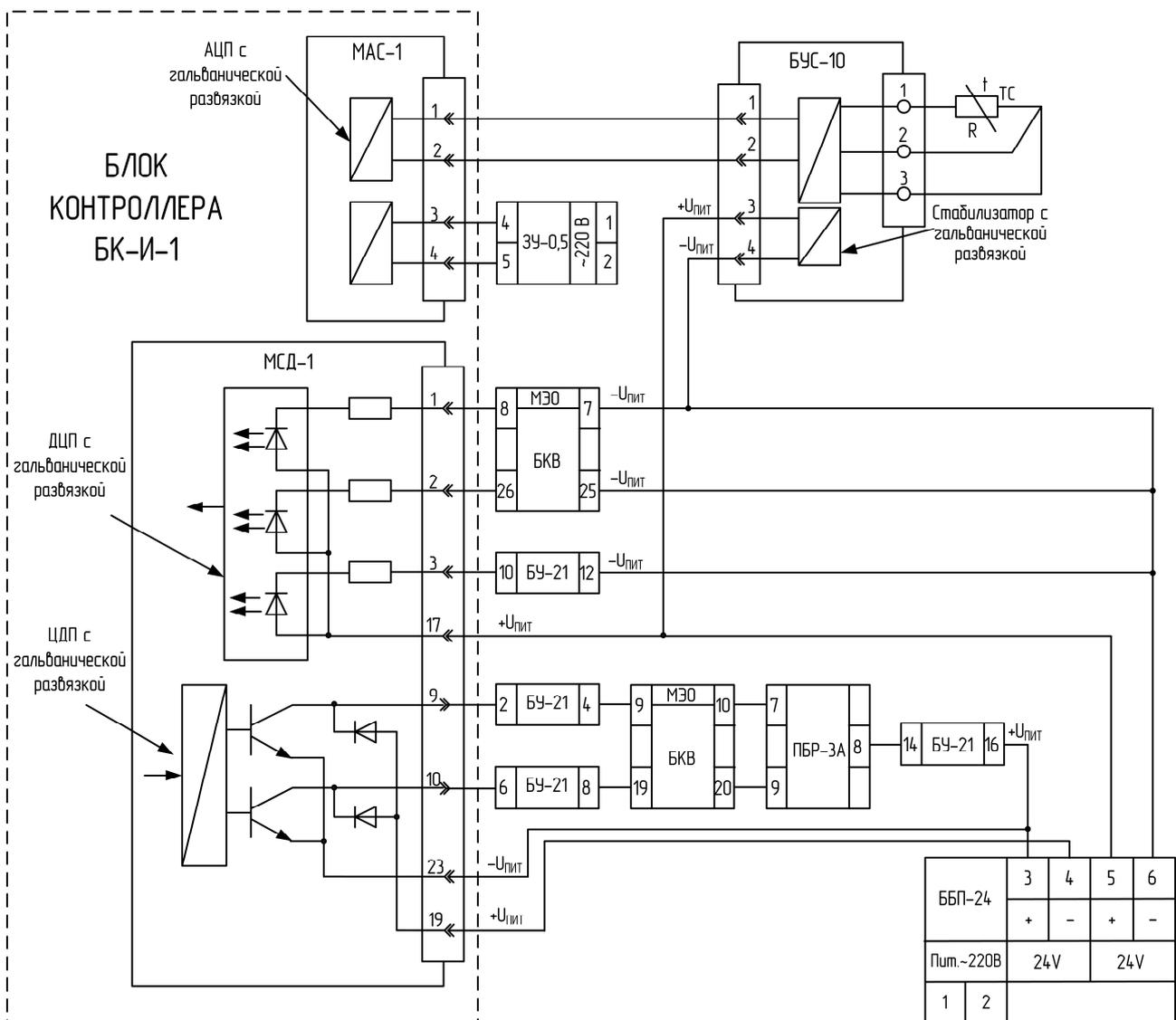


Рис. 6.5 Организация автоматического управления

7. Контроллер и платы ввода-вывода.

Для организации ввода аналоговых сигналов выбирается модуль МАС-1. Модуль аналоговых сигналов имеет 8 аналоговых входов и 2 аналоговых выхода. Каждый аналоговый вход подключается к “своему” аналого-цифровому преобразователю (АЦП). Аналоговые входы контроллера рассчитаны на сигнал 0-5 мА или 0-20 мА постоянного тока. Верхнее значение диапазона (5 мА или 20 мА) определяется номиналом резистора, расположенного в модуле. Настройка входа на диапазон 0-20 мА или 4-20 мА осуществляется в системе программи-

рования. Каждый аналоговый вход имеет гальваническую изоляцию от других аналоговых входов и других цепей контроллера.

Для реализации системы используется два аналоговых входа:

- первый канал используется для снятия сигнала с БУС-10.

- второй канал отведен для задающего воздействия.

Для реализации управляющих воздействий используется плата дискретного ввода-вывода МСД-1 8/8.

Модуль дискретных сигналов содержит 8 дискретных входов и 8 дискретных выходов. Все дискретные входы разбиты на группы по четыре входа. Дискретные входы одной группы имеют общую точку и являются пассивными. Для питания входов требуется внешнее нестабилизированное напряжение 24 В постоянного тока. Каждая группа входов имеет гальваническую изоляцию друг от друга и от других цепей контроллера.

Используются три дискретных входа и два дискретных выхода

– первые два входа использованы для отработки аварийной ситуации, возникающей при нарушении в работе исполнительного механизма – МЭО, а именно, когда размыкаются концевые выключатели. Через них заводится дискретный сигнал 24 В на входы контроллера. Если такой сигнал пропадает, то регулятор останавливает свою работу и сообщает о возникновении такой ситуации. Третий вход использован режима сигнала автоматического управления, подаваемого через контакт БУ-21 на вход регулятора.

– дискретные выходы предназначены для регулирования положения вала МЭО. Дискретный сигнал подается на вход 7 «больше» и вход 9 «меньше» МЭО.

Платы МАС-1 и МСД-1 устанавливаются в блок контроллера БК-И-1 в группы 1 и 2 соответственно.

В качестве источника питания системы использован блок бесперебойного питания ББП-24.

3. «Логическая» часть системы управления

1. Язык программирования КР-300.

Язык программирования КР-300 является непроцедурным. При программировании не задается порядок выполнения операций, а создается виртуальная (кажущаяся) структура, которая описывает информационную организацию контроллера и характеризует его как звено системы управления.

Часть элементов виртуальной структуры реализована аппаратно: аппаратура ввода-вывода информации, аппаратура оперативного управления и настройки, аппаратура интерфейсного канала. Часть реализовано программно в виде алгоритмических блоков (алгоблоков) и библиотеки алгоритмов. Алгоблок служит для хранения одного из библиотечных алгоритмов контроллера. Алгоблок с помещенным в него алгоритмом может рассматриваться как виртуальный прибор, выполняющий алгоритмическую обработку информации в соответствии с помещенным в него алгоритмом. Он обладает входами и выходами в количестве, присущем данному алгоритму. Алгоблоки соединяются друг с другом и с входами-выходами контроллера программным путем. В КР-3000 можно использовать до 999 алгоблоков.

Библиотека алгоритмов – это перечень алгоритмов управления, которые могут помещаться в алгоблоки. Библиотека насчитывает 251 алгоритм. В ее состав входят алгоритмы автоматического регулирования, динамических преобразований, логики, арифметических операций.

Часть библиотечных алгоритмов, которые называются специальными, выполняют особую задачу: они связывают аппаратуру контроллера с основной массой функциональных алгоритмов.

К специальным алгоритмам относятся: алгоритмы ввода и вывода аналоговых и дискретных сигналов; алгоритмы обслуживания лицевой панели; алгоритмы приема и передачи сигналов через интерфейсный канал.

Аппаратные средства виртуальной структуры (УСО, лицевая панель, интерфейсный канал) начинают выполнять свои функции после того, как в какие-либо алгоблоки будут помещены соответствующие алгоритмы.

В исходном состоянии в алгоблоках отсутствуют алгоритмы управления, и алгоблоки не связаны друг с другом и аппаратной частью виртуальной структуры. При программировании контроллера алгоритмы помещаются в алгоблоки и между алгоблоками программно устанавливаются связи.

При размещении алгоритмов в алгоблоках в большинстве случаев действуют два правила:

1. Любой алгоритм можно помещать в любой (по номеру) алгоблок, за исключением алгоритмов, обслуживающих лицевую панель. Эти алгоритмы могут быть помещены в первые четыре алгоблока (номер алгоблока определяет номер контура регулирования).

2. Один и тот же алгоритм можно помещать в разные алгоблоки, т.е. использовать многократно.

При размещении необходимо задать реквизиты (параметры) алгоритма: библиотечный номер, модификатор и масштаб времени.

Библиотечный номер представляет собой двухзначное число, под которым данный алгоритм хранится в библиотеке, и является основным параметром, характеризующим свойства алгоритма.

Модификатор задает дополнительные свойства алгоритма. В частности в алгоритме суммирования модификатор задает число суммируемых входных сигналов, в алгоритме программного задатчика – количество участков и т.д.

Масштаб времени имеется только в алгоритмах, чья работа связана с реальным временем, например, регулирование, программный задатчик, таймер и т.д. Масштаб времени задает одну из двух размерностей для временных сигналов или параметров. Если контроллер настроен на младший диапазон, то масштаб времени индивидуально в каждом алгоблоке задает масштаб «секунды» или «минуты». Для старшего диапазона масштаб времени задает «минуты» или «часы».

Соединение алгоблоков между собой и с аппаратной частью контроллера осуществляется операцией конфигурирования. В процессе конфигурирования для каждого входа алгоблока задается источники сигнала или параметры на-

стройки, т.е. каждый вход алгоблока находится в одном из двух состояний – связанном или свободном.

Вход считается связанным, если он соединен с выходом какого-либо алгоблока, в противном случае вход считается свободным.

Сигналы на свободных входах могут быть представлены в виде констант или в виде коэффициентов. Отличие между ними заключается в возможности их изменения: константы можно устанавливать и изменять только в режиме программирования, коэффициенты можно также устанавливать и изменять и в режиме работы.

Возможности конфигурирования не зависят от алгоритма, помещенного в алгоблок, и определяются тремя правилами:

1. Любой вход любого алгоблока можно связать с любым выходом любого алгоблока или оставить свободным.

2. На любом свободном входе любого алгоблока можно вручную задавать сигнал в виде константы или коэффициента.

3. На любом входе любого алгоблока сигнал можно инвертировать

4. Пример реализации «логической» части системы управления

Реализация алгоритмической схемы системы регулирования осуществляется после выбора и обоснования закона управления, а также после выполнения электрической схемы подключений.

Для реализации законов адаптивного управления, рассмотренных в лабораторной работе №2 с учетом электрической схемы подключений на рис.6.5, необходимы следующие алгоритмы.

Для ввода – вывода информации используются алгоритмы № 10 «Ввод аналоговый», № 11 «Ввод дискретный» и № 14 «Дискретный вывод». Эти алгоритмы применяются для связи функциональных алгоритмов с аппаратными средствами аналогового ввода (АЦП), дискретного ввода (ДЦП) и дискретного вывода (ЦДП). Номер слота контроллера, куда установлен модуль, указывается модификатором типа МТ, соответствующий размер – МР.

Для получения рассогласования используется алгоритм № 43 «Суммирование с масштабированием». Первый его вход является не масштабируемым и не используется. На два другие подается температура и задание с соответствующим смещением.

Алгоритм № 47 «Кусочно-линейная функция» реализует зона нечувствительности $\pm\delta$.

Алгоритм № 61 «Импульсатор» используется для периодического включения и выключения нагрузки в том случае, когда скважность включения должна быть пропорциональна непрерывному управляющему сигналу. Период следования импульсов задается сигналом на настроечном входе 2 алгоритма $T=0,24$. Каждая ячейка алгоритма представляет собой широтно-импульсный модулятор с заданным периодом и меняющейся длительностью включения.

«Переключатель» № 57 необходим для организации автоматического и ручного режимов, а также останова процесса регулирования в случае достижения исполнительного механизма конечных выключателей.

Алгоритм № 71 «Многовходовая логическая операция» представляет собой логическую операцию «ИЛИ», которая задается модификатором типа (MT=2).

При реализации нелинейной части адаптивного регулятора используются следующие алгоритмы:

– № 44 «Умножение – деление» для реализации умножителей с параметром деления 1;

– № 33 «Интегратор» - для получения интегральной составляющей управляющего воздействия;

Линейная часть адаптивного регулятора, а также компенсатор реализованы на алгоритме № 36 «Динамическое преобразование». Алгоритм применяется для динамической коррекции систем управления в тех случаях, когда требуется интегро-дифференцирующее преобразование сигнала.

Алгоритм реализует передаточную функцию:

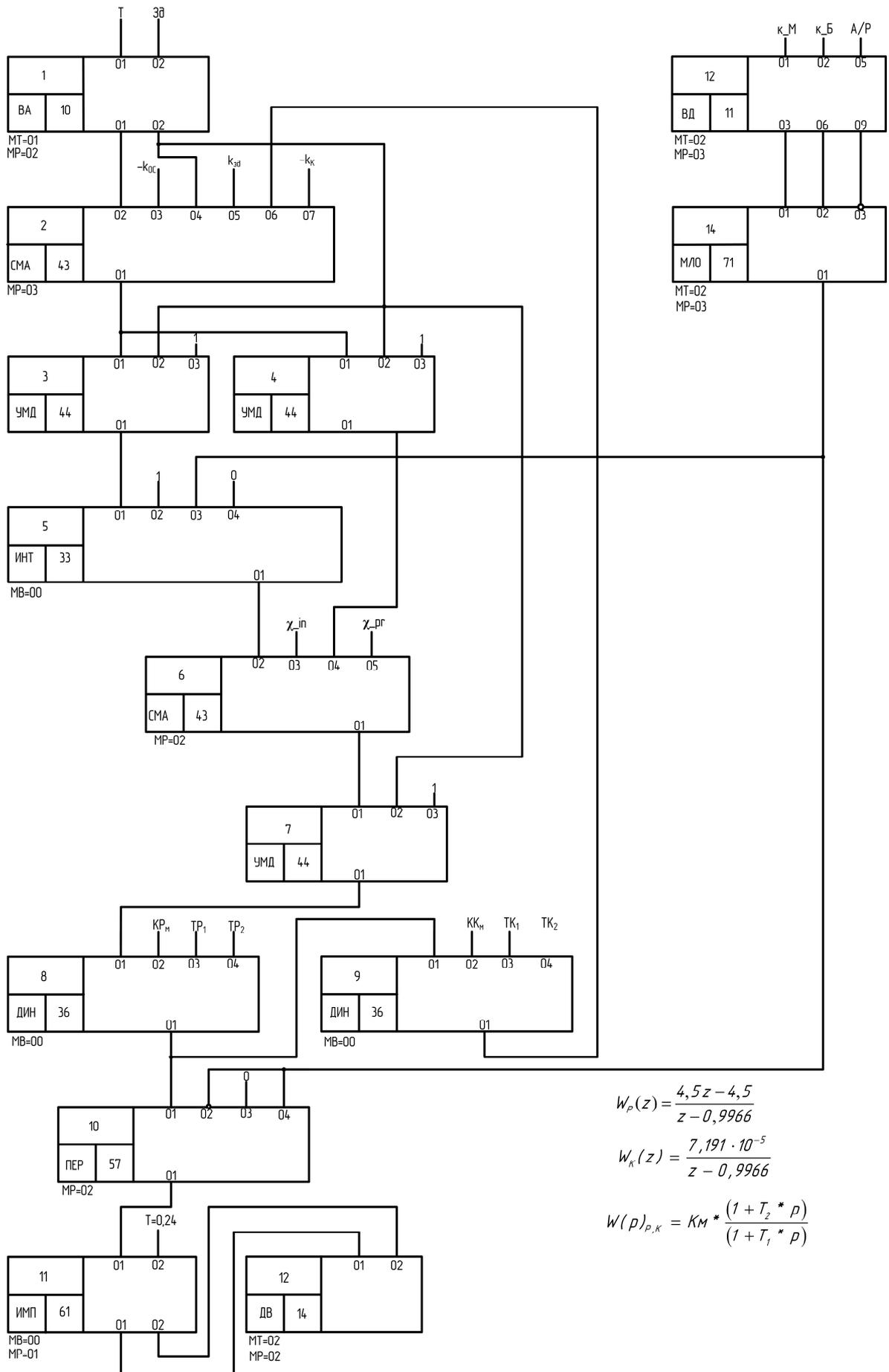
$$W(p) = \frac{Y(p)}{X(p)} = K_M * \frac{(1 + T_2 * p)}{(1 + T_1 * p)}, \quad (6.4)$$

где K_M — коэффициент усиления; T_1 и T_2 — постоянные времени.

Поскольку передаточные функции регулятора и компенсатора в дискретной форме имеют вид (6.2), (6.3), то данный алгоритм подходит для их реализации (K_M , T_1 и T_2 задаются соответственно).

Составленная алгоритмическая схема представлена на рис. 6.6 с указанием всех необходимых коэффициентов и модификаторов.

Для реализации классической системы регулирования используется ПДД² – регулятор, реализованный на алгоритме № 21 «Регулирование импульсное». На его входах 6, 7, 8, 9 задаются соответствующие настройки регулятора. Остальные коэффициенты принимаются в данном случае равными нулю.



$$W_p(z) = \frac{4,5z - 4,5}{z - 0,9966}$$

$$W_k(z) = \frac{7,191 \cdot 10^{-5}}{z - 0,9966}$$

$$W(p)_{p,k} = KM * \frac{(1 + T_2 * p)}{(1 + T_1 * p)}$$

Рис. 6.6 Алгоритмическая схема системы

Порядок работы

1. Получить дискретные модели классической системы управления, адаптивной системы управления со стабилизирующим устройством по вариантам.
2. Произвести сравнение аналоговой и дискретной систем по основным показателям качества.
3. Спроектировать принципиальную электрическую схему системы, согласно варианту. Реализовать автоматическое и ручное управление.
4. Реализовать предлагаемую систему в виде алгоритмической структуры на КР-300, либо на другой промышленном контроллере [ОВЕН].

Варианты заданий:

Содержание отчета

1. Simulink – модели непрерывной и дискретной систем.
2. Основные переходные характеристики в системах
3. Принципиальная электрическая схема.
4. Алгоритмическая схема (на выбор преподавателя – для классической или адаптивной системы управления).

Лабораторная работа №7

Учет шумов по каналу измерения выходной переменной

Цель работы:

1. Моделирование дискретно-непрерывной системы регулирования с шумленным каналом измерения контролируемой величины.
2. Подавление шумов в канале измерения с помощью фильтров.

Краткие теоретические сведения

1. Измерение выходной величины

Математическая модель редко отражает реальность – всякое описание строится на основе некоторых допущений. Во многих случаях модель не учитывает все тонкости реального процесса, а некоторыми состояниями просто пренебрегают. Поэтому при проектировании систему управления всегда необходимо задаваться вопросом о том, какая модель является более адекватной с позиции неучтенных факторов – неопределенностей.

Особенно важным для цифрового управления являются измерения и технология датчиков. Датчики должны отображать физические переменные технологического процесса, как в стационарных, так и в переходных режимах работы.

Моделирование возмущений и случайных ошибок в датчиках основывается на теории стохастических процессов. Стохастический процесс – это последовательность случайных величин. В принципе это означает, что в момент выборки к значению каждой переменной процесса добавляется случайная величина с известным распределением вероятности. Аналогично шум в измерениях можно смоделировать как случайную добавку к сигналу датчика

На практике не существует идеальных датчиков. Шумы и возмущения в датчике можно рассматривать как дополнительную случайную переменную e в уравнении состояния. Возмущения, влияющие на все датчики технологическо-

го процесса можно записать как вектор e , который добавляется в уравнение выхода системы в дискретном виде:

$$y(k \cdot h) = C \cdot x(k \cdot h) + e(k \cdot h), \quad (7.1)$$

где h – размер выборки, $k = 1, 2, \dots, n$ – порядковый номер шага, n – размер выборки.

Каждый компонент вектора шума $e(k \cdot h)$ моделируется как последовательность стохастических, то есть случайных, значений.

2. Фильтрация выходной величины

Фильтрация (рис. 7.1) применяется для уменьшения амплитуды определенных частотных составляющих сигнала. Фильтрация может быть как аналоговой – с помощью электронных цепей, так и цифровой, изменяющей дискретные значения сигнала, после аналогоцифрового преобразования. Для эффективной фильтрации необходимо, чтобы частотные диапазоны разделяемых сигналов не пересекались. Наиболее часто фильтрация применяется для устранения шума и помех и для усиления сигнала, насколько это возможно, до первоначального уровня. На практике, фильтрация имеет смысл только тогда, когда исходный сигнал с самого начала был защищен от влияния помех на этапе передачи от датчика к контроллеру [17].

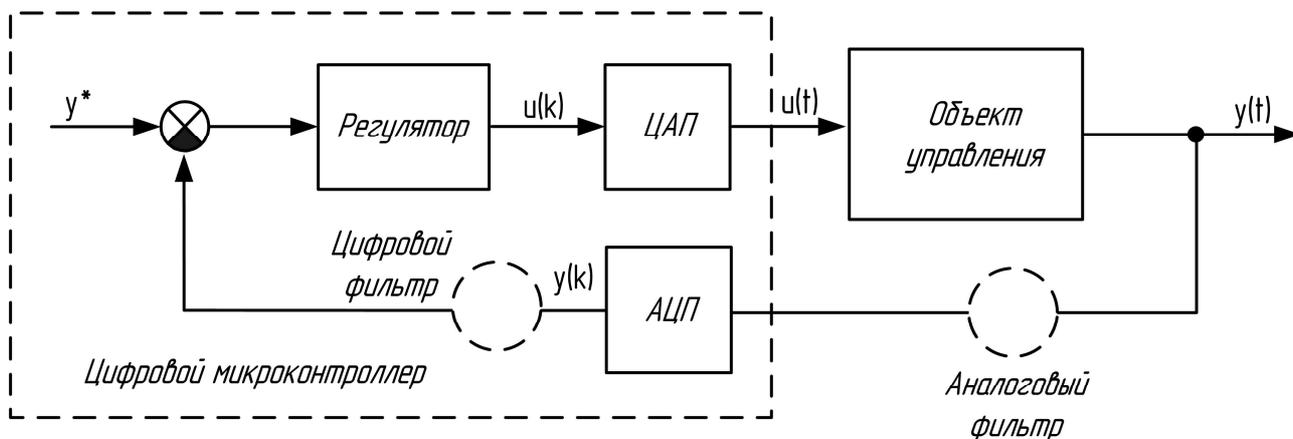


Рис. 7.1 Фильтрация измеряемой величины

Двумя основными параметрами аналоговой фильтрации являются ширина полосы пропускания и граничная частота. Полоса пропускания - это диапазон частот, проходящих фильтр без изменения. Граничная частота – это частота, на которой амплитуда ослабляется в $\sqrt{2}$ раз, что соответствует уменьшения мощности в 2 раза по сравнению с пропускаемыми частотами.

Фильтр низкой частоты пропускает частоты ниже граничной частоты и ослабляет компоненты с частотами выше этого значения. Фильтр высокой частоты пропускает высокие частоты и ослабляет низкие. Полосовые фильтры пропускают частотные компоненты, лежащие между двумя граничными отсекающими частотами.

Простейшим примером аналогового фильтра низкой частоты является апериодическое звено первого порядка (рис. 7.2):

$$W_{\Phi}(p) = \frac{1}{T_{\Phi}p + 1} \quad (7.2)$$

$$f_c = \frac{1}{2\pi \cdot T_{\Phi}}$$

где T_{Φ} - постоянная времени фильтра, f_c – частота среза.

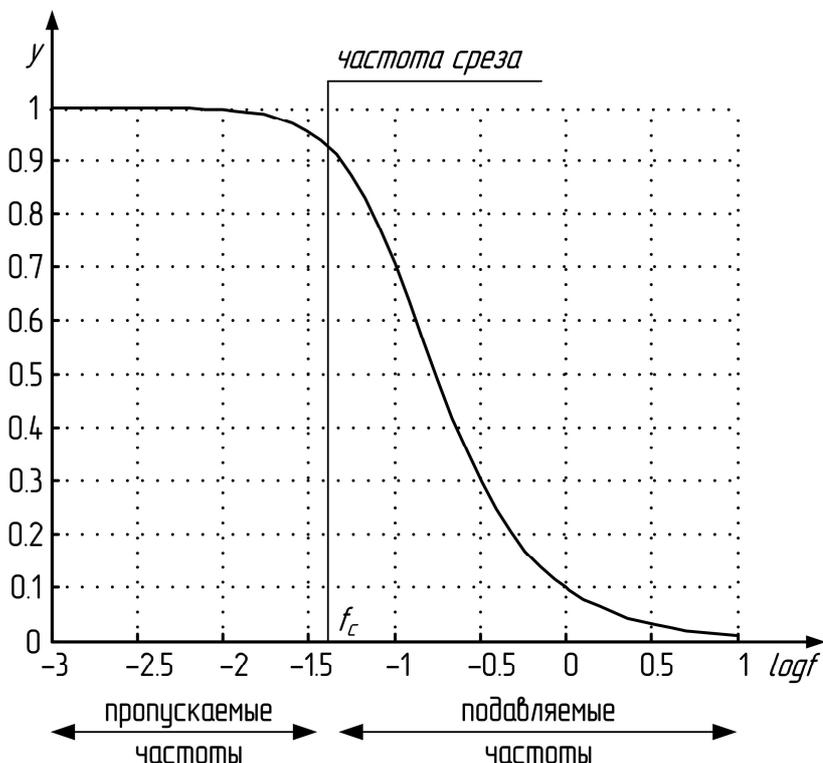


Рис.7.2 Частотная характеристика фильтра низкой частоты

Фильтр высокой частоты (рис.7.3) можно задать передаточной функцией реального дифференцирующего звена:

$$W_{\Phi}(p) = \frac{T_{\Phi} p}{T_{\Phi} p + 1}, \quad (7.3)$$

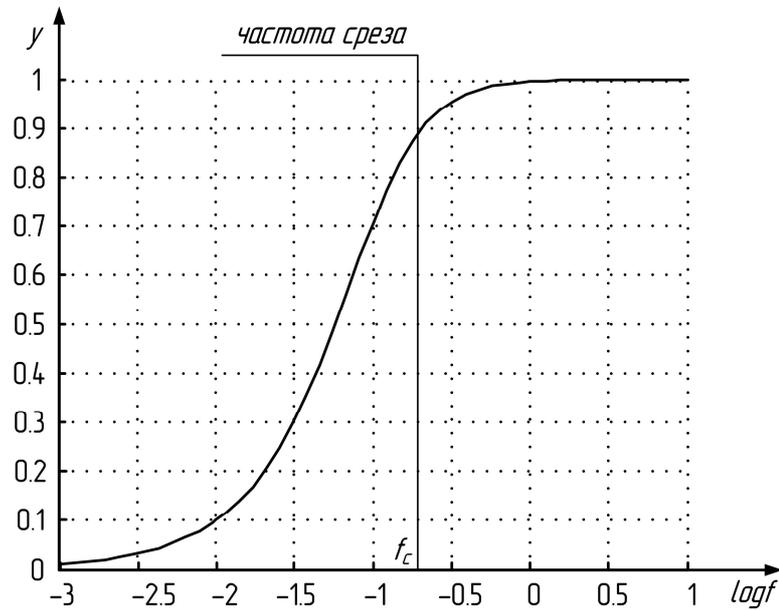


Рис.7.3 Частотная характеристика фильтра высокой частоты

Цифровая фильтрация обладает большей гибкостью по сравнению с аналоговой, поскольку характеристики фильтра можно изменить, просто задав новые параметры соответствующей ему программы. В отличие от аналоговых, цифровые фильтры хорошо работают с длительными постоянными сигналами.

В общем виде цифровой фильтра можно представить как:

$$\hat{y}(hk) = -a_1 \cdot \hat{y}([k-1]h) - a_2 \cdot \hat{y}([k-2]h) - \dots - a_n \cdot \hat{y}([k-n]h) + b_0 \cdot y(kh) + \dots + b_m \cdot y([k-m]h), \quad (7.4)$$

где h – интервал выборки, \hat{y} – отфильтрованный выход, y – вход.

Если все коэффициенты a_i в (7.4) равны нулю, то такой фильтр называется фильтром скользящего среднего с конечной импульсной характеристикой. Это означает, что если в течение некоторого времени все последовательные значения y_j , кроме одного, равны нулю, то на выходе сигнал будет отличен от нуля только на m временных интервалах. Если некоторые либо все коэффициенты a_i не равны нулю, то такой фильтр называется авторегрессивным и имеет бесконечную импульсную характеристику. Другими словами, входной сигнал,

отличающийся от нуля только на одном временном интервале, вызовет ан выходе появление сигнала отличного от нуля в течение бесконечно долгого времени. Обобщенный фильтр, описываемый уравнением (7.4), называется авторегрессивным фильтром скользящего среднего.

Порядок работы

1. В полученной ранее адаптивной системе управления со стабилизирующим устройством ввести зашумленность измерительного канала (вид шума задается преподавателем).

2. Путем подбора параметров шума добиться его значительного влияния на объект. В Simulink источник шума может быть задан, например, блоком «Белый шум», «синусоидой», другой какой-либо нелинейностью.

Содержание отчета

1. Структуры систем. Переходные процессы в них. Вид шума.
2. Выводы о влиянии шума на работу системы.

Лабораторная работа №8

Графический пользовательский интерфейс MATLAB

Цель работы:

1. Знакомство с пользовательским графическим интерфейсом пакета прикладных программ Matlab.
2. Создание собственного графического приложения для системы автоматического регулирования.

Краткие теоретические сведения

Приложения MATLAB с графическим интерфейсом являются графическими окнами, содержащими элементы управления (кнопки, списки, переключатели, флаги, полосы скроллинга, области ввода, меню), а также оси и текстовые области для вывода результатов работы (рис. 8.1). Создание приложений включает следующие основные этапы – расположение нужных элементов интерфейса в пределах графического окна и программирование событий, которые возникают при обращении пользователя к данным объектам, например, при нажатии кнопки. Процесс работы над приложением допускает постепенное добавление элементов в графическое окно, запуск, тестирование приложения и возврат в режим редактирования. Конечным результатом является программа с графическим интерфейсом пользователя, содержащаяся в одном или нескольких файлах, запуск которой производится указанием ее имени в командной строке или в другом приложении MATLAB.

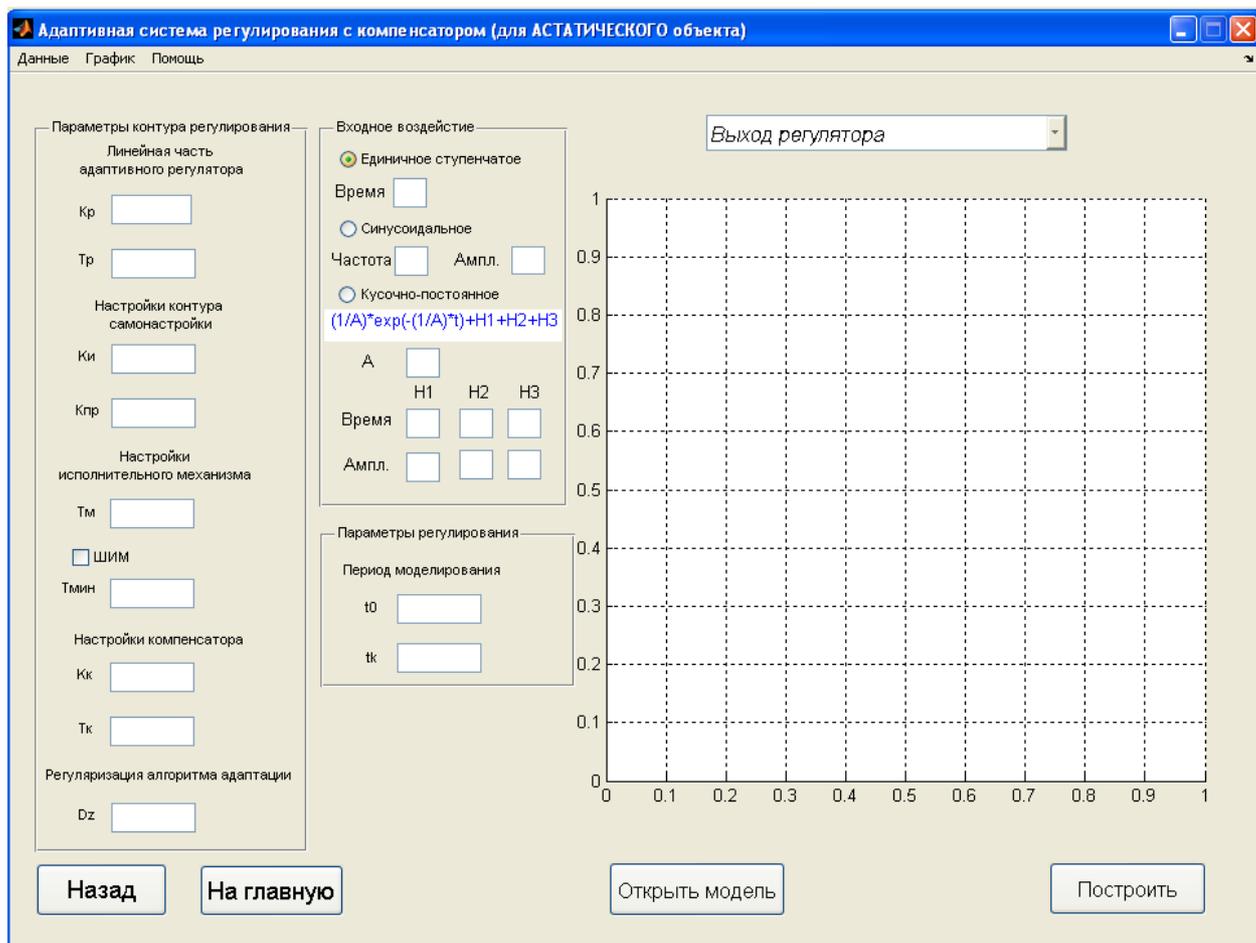


Рис. 8.1 Графический пользовательский интерфейс

1. Графический редактор

Запуск редактора среды GUIDE производится одноименной командой в командном окне MATLAB. Для создания нового приложения, в открывшемся диалоговом окне необходимо выбрать *Blank GUI* (рис. 8.2). Редактор среды GUIDE содержит строку меню; панель инструментов управления приложением; заготовку окна приложения с нанесенной сеткой; панель инструментов для добавления элементов интерфейса на окно приложения (рис. 8.3).

Меню *File* позволяет выполнять стандартные операции по созданию, сохранению или открытию приложения. Кроме того, в подменю *Preferences* можно изменить настройки редактора GUIDE.

При сохранении вновь созданного в редакторе GUIDE приложения, например, с именем *example*, создаются два файла – *example.fig* и *example.m*. Файл с расширением **.fig* содержит графическую структуру приложения, то есть

расположение элементов и их форматирование. Файл с расширением **.m* определяет перечень событий, который выполняется в зависимости от того или иного действия над элементами окна приложения в процессе его функционирования. Порядок такой структуры – *fig*-файл и соответствующий ему *m*-файл, можно изменить в *Tools – GUI Options*.

Меню *Edit* и меню *Layout* позволяют редактировать элементы в окне приложения (копирование, вставка, задний и передний планы, и другое).

Меню *View* содержит следующие элементы: *Property Inspector*, *Object Browser*, *M-file Editor* и *View Callbacks*. Подменю *Property Inspector* отображает свойства любого объекта окна приложения. Подменю *M-file Editor* открывает редактор *m*-файла, соответствующего данному *fig*-файлу. Подменю *View Callbacks* выводит список возможных событий в окне приложения, которые можно запрограммировать для его элементов.

Меню *Tools* содержит следующие подменю: *Run* (для запуска приложения), *Align Object* (выравнивание элементов по горизонтали и вертикали, и другое), *Grid and Rulers* (сетка и линейка), *Tab Order Editor* (редактор обхода элементов окна приложения клавишей **TAB**). Воспользовавшись инструментом *Tools – Menu Editor* можно создавать собственные главное и контекстное меню (рис. 8.4).

Меню *Help* содержит рекомендации и примеры по программированию в среде GUIDE.

Панель инструментов управления приложением дублирует функции главного меню.

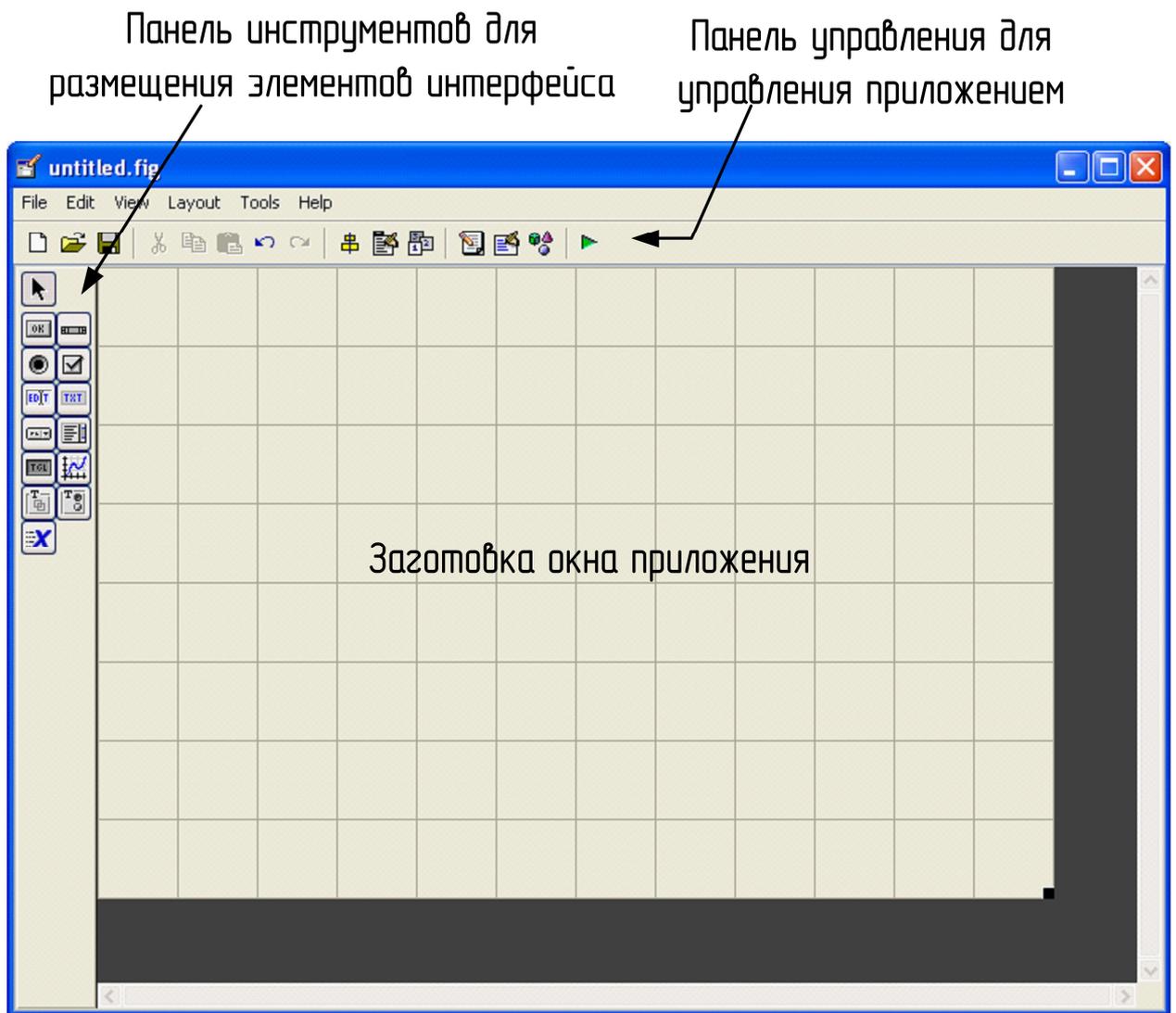


Рис. 8.2 Графический пользовательский интерфейс

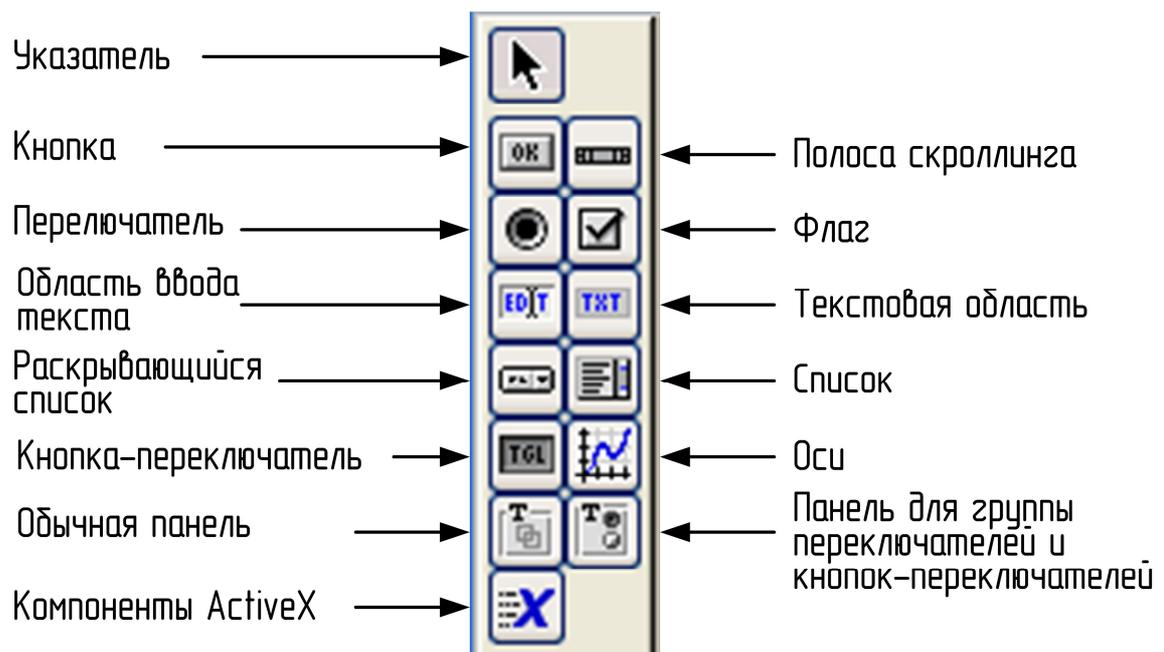


Рис. 8.3 Панель инструментов для добавления элементов интерфейса

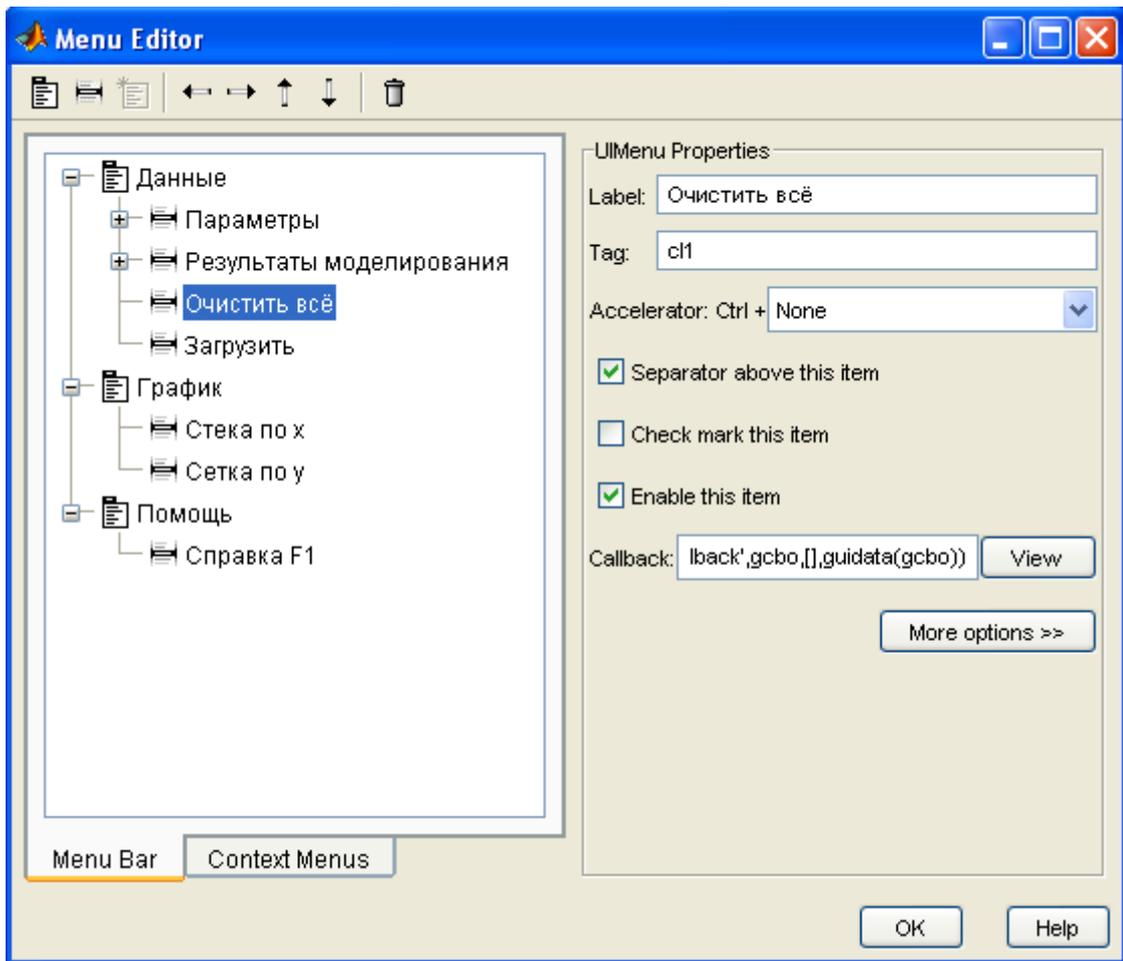


Рис. 8.4 Редактор меню

2. Создание приложения в среде GUIDE

Ставится задача создать GUI – приложение, управляющее сеансом имитационного моделирования в Simulink. В качестве Simulink-модели используется адаптивная система, рассмотренная в лабораторной работе №4. Предполагается, что обмен данными между GUI – приложением и Simulink-моделью происходит в двустороннем порядке. В GUI – приложении задаются основные настройки системы и контура управления, после чего выполняется имитационное моделирование. Результаты имитационного моделирования передаются обратно в GUI – приложение в виде необходимых для анализа системы характеристик (выход объекта, настройки регулирующего контура и другие).

Примером послужит GUI – приложение, изображенное на рис. 8.1.

После создания новой заготовки приложения, его наименования необходимо определиться с элементами интерфейса. Как показано выше, элементы интерфейса создаются с помощью панели инструментов.

Область ввода теста используется для задания параметров системы, моделируемой в Simulink. После добавления области ввода на окно приложения, необходимо вызвать инспектор свойств объекта (*Property Inspector*), показанный на рис. 8.5.

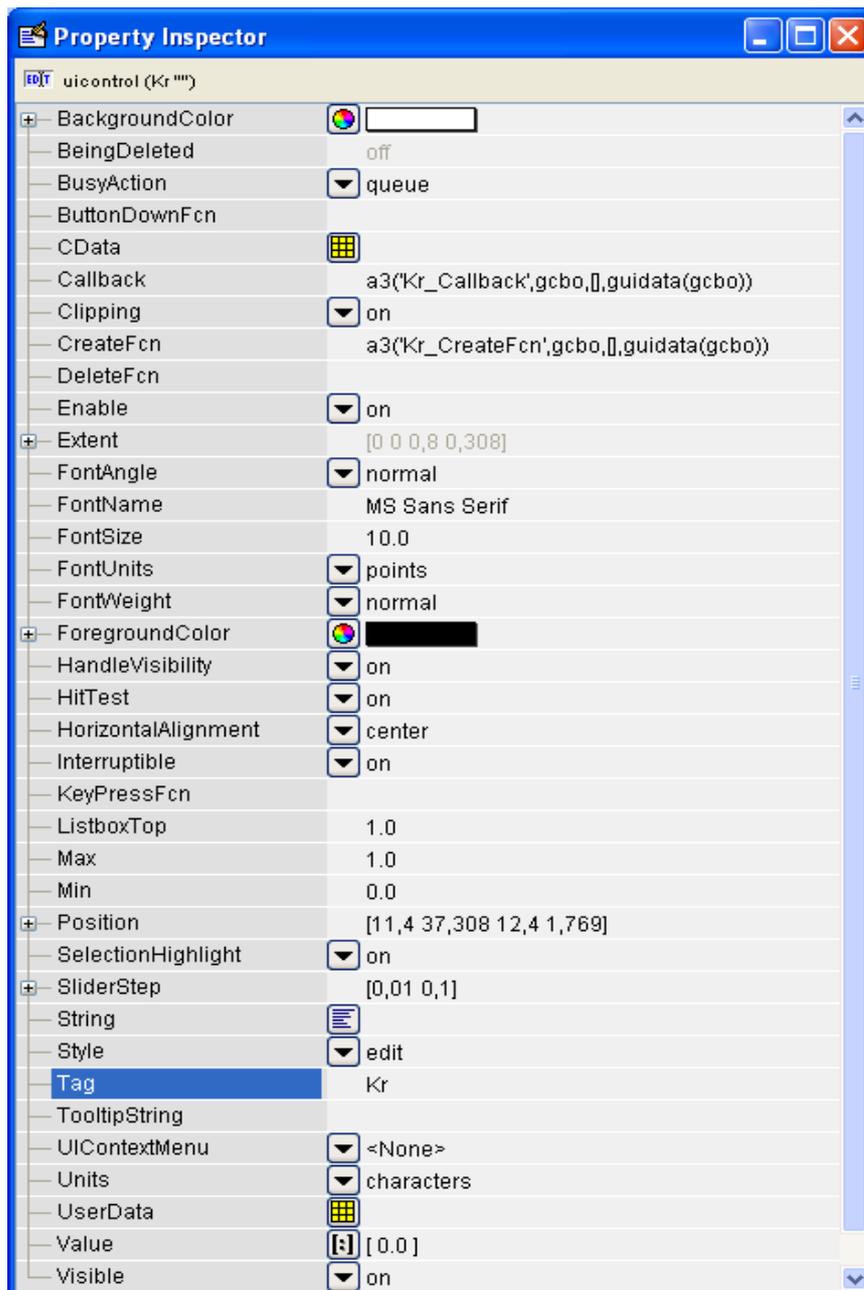


Рис 8.5 – Инспектор свойств

Инспектор свойств – это таблица названий свойств области ввода и их значений. Важнейшим свойством является имя элемента – *Tag*. Например,

пусть коэффициент усиления линейной части регулятора Kp обозначается через тэг Kr . В данном случае этому полю присвоено значение Kr . Значение поля Tag необходимо задавать максимально информативно.

Текстовая область используется для подписи областей ввода текста. После добавления на окно приложения текстовой области, также необходимо вызвать инспектор свойств. В данном случае, информативность значения поля Tag не важна, так как тестовая область, как правило, представляет собой статическую информацию. Важное значение имеет другое свойство – $String$. Значение поля $String$ задает подпись элемента в окне приложения, в данном случае это Kp .

Другие параметры системы на рис. 8.1 задаются аналогично. Кнопкам присваиваются также два значения – это ее имя и подпись в окне приложения. Оси (плоскость для построения графиков) имеет только значение свойства Tag .

Параметры, схожие по своему смыслу, для удобства использования можно объединить «обычной панелью». На рис. 8.1 все настраиваемые параметры объединены панелью с наименованием «Параметры контура регулирования».

Для анализа рассматриваемой системы регулирования удобно использовать несколько видов задающего воздействия. Каждый вид задающего воздействия – взаимоисключающий. Поэтому в данном случае необходимо использовать «панель для группы переключателей», то есть задание может быть либо ступенчатым, либо синусоидальным, либо кусочно-постоянным.

Присутствие/отсутствие в системе управления широтно-импульсного модулятора отражается в интерфейсе приложения как «флаг». При пустом флаге $ШИМ$, окно ввода значения $Tmin$ будет *неактивно*. Для этого в инспекторе свойств соответствующей области ввода необходимо свойству $Enable$ присвоить значение *off*. Если флаг $ШИМ$ будет активен, то и область ввода $Tmin$ станет также активной. Обратный процесс аналогичен, но реализуется он при обработке событий, наступающих при включении/выключении флага.

Для вывода результатов моделирования предусмотрены оси (плоскость для построения графиков) и список характеристик. Каждой характеристике со-

ответствует свой график. Такое соответствие также задается при обработке соответствующих событий в *m* – файле.

3. Обработка событий приложения GUIDE

Как правило, *m* – файл состоит из двух частей. Первая, заголовочная часть состоит из инициализационного кода, который редактировать не рекомендуется (рис. 8.6).

```
function varargout = a3(varargin)
%Инициализационный код
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @a3_OpeningFcn, ...
                  'gui_OutputFcn',  @a3_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

Рис. 8.6 Инициализационный код

Вторая часть – это подфункции вызова элементов интерфейса. При активации какого-либо элемента работающего интерфейса (ввод теста, нажатие кнопки) генерируется событие *Callback*. При этом вызывается подфункция в виде *имя_элемента_Callback*.

Для кнопки **ПОСТРОИТЬ** с тэгом *pl1* формируется следующая подфункция (рис. 8.7).

```

function pl1_Callback(hObject, eventdata, handles)
% hObject      handle to pl1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Проверка задания
if handles.vid_z==0
    %Диалоговое окно ошибки
    errordlg('Укажите вид задания!', 'Ошибка задания параметров', 'modal');
    % Возвращение фокуса на панель выбора задания
    uicontrol(handles.st);
else
M1=handles.M1;%Массив имён переменных
M=handles.M;%Массив указателей на переменные

for i=1:length(M1)
    %Если не все переменные заданы, им присваивается значение 1
    if length(get(M(i), 'String'))==0
        assignin('base', deblank(M1(i, :)), 1);
    else
        %Передача переменных в Симулинк с соответствующим значением
        assignin('base', deblank(M1(i, :)), str2num(get(M(i), 'String')));
    end
end

assignin('base', 'vid_z', handles.vid_z);
assignin('base', 'sh0', handles.sh0);
assignin('base', 'sh1', handles.sh1);

t0=handles.t0; %Начальное время моделирования
tk=handles.tk; %Конечное время моделирования
set(handles.grafik, 'Enable', 'on'); %Активизация графика
sim ('./sim/ada', [t0 tk]); %Запуск модели в Симулинк
%Вызов подфункции grafik_Callback для построения характеристик
grafik_Callback(hObject, eventdata, handles);
end

```

Рис. 8.7 Подфункция *pl1_Callback*

Как видно из листинга на рисунке 8.7, подфункция *pl1_Callback* содержит три входных параметра. Указателем на текущий объект является *hObject*, *handles* – это структура, содержащая указатели на все элементы интерфейса и пользовательские переменные, *eventdata* – параметр, зарезервированный системой (не используется).

Пояснения по листингу рис. 8.7:

– обращение, например, к полю *vid_z* структуры *handles* осуществляется как *handles.vid_z*;

– все переменные, объявленные в GUI – приложения и подфункциях являются локальными. Чтобы Simulink «увидел» эти переменные необходимо воспользоваться функцией

$$\text{assignin}(\text{'base'}, \text{имя_переменной}, \text{значение_переменной}), \quad (8.1)$$

которая присваивает переменной (типа *char*) какое-либо значение и передает данное равенство в рабочую область MATLAB. Таким образом локальные переменные становятся глобальными.

– чтобы установить какой-либо свойство элементы интерфейса, можно воспользоваться следующей функцией:

$$\text{set}(\text{указатель_на_переменную}, \text{свойство}, \text{значение_свойства}). \quad (8.2)$$

Например, $\text{set}(\text{handles.grafik}, \text{'Enable'}, \text{'on'})$. Чтобы получить значение свойства элемента интерфейса, используется функция

$$\text{get}(\text{указатель_на_переменную}, \text{свойство}). \quad (8.3)$$

– запуск процесса моделирования осуществляется командой *sim*. При этом должна существовать модель *ada.mdl* и должны быть заданы все переменные моделирования. Пусть к фалу *ada.mdl* задается относительно корневой директории.

При достаточно большом количестве переменных, их удобно объединять в массивы. Это действие производится при запуске приложения (рис. 8.8).

```

function a3_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

%Задание переменных для дальнейшего использования
handles.vid_z = 0; %Вид задания
handles.sh = 0; %Значение флага ШИМ
handles.sh0 = 0; %ШИМ выключен
handles.sh1 = 1; %ШИМ включен

%Массив указателей переменных
M = [handles.Kr,handles.Tr,handles.Ki,handles.Kpr,...
     handles.Kk,handles.Tk,...
     handles.Tm,handles.Tmin,handles.t1,handles.w2,...
     handles.Am2,handles.Am3,handles.t31,handles.t32,...
     handles.t33,handles.Am31,handles.Am32,handles.Am33,...
     handles.t0,handles.tk,handles.Dz];

%Массив имен переменных
M1 =char('Kr','Tr','Ki','Kpr','Kk','Tk','Tm','Tmin','t1','w2',...
        'Am2','Am3','t31','t32','t33','Am31','Am32','Am33','t0','tk','Dz');
handles.M = M;
handles.M1 = M1;
handles.param= './param/';
%Деактивация осей
set(handles.grafik,'Enable','off');
%Наполнение списка характеристик, предусмотренных для данного типа систем
cd init;
l = grafik45(handles.grafik,5);
cd ..;
%Сохранение объявленных переменных в структуре handles
guidata(hObject, handles);

```

Рис. 8.8 Инициализация переменных

4. Взаимодействие Simulink – модели с GUI – приложением

После проведения сеанса имитационного моделирования необходимо передать полученные характеристики переходных процессов обратно в GUI - приложение. Для этого в Simulink – модель добавляются библиотечные блоки *To File* из пакета *Sinks* (приёмники данных) как показано на рис. 8.9. Основным параметром блока *To File* является его имя и путь, по которому файл будет сохранен после сеанса имитационного моделирования. Путь задается также относительно корневой директории. То есть на рис. 8.9 характеристики выхода ком-

пенсатора (*59.mat*) и выхода объекта управления (*51.mat*) будут сохранены в папку *.\param*.

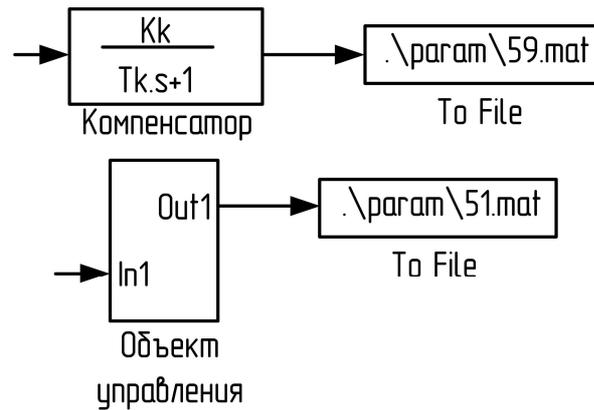


Рис. 8.9 Блоки передачи данных в GUI – приложение

В файлах типа **.mat* хранятся преимущественно массивы чисел. Чтобы считать данные из такого файла необходимо выполнить команды:

$$m = load (\text{имя_переменной}.mat) \quad (8.4)$$

$$t = m.ans (1, :) \quad (8.5)$$

$$y = m.ans (2, :) \quad (8.6)$$

Команда (8.4) загружает в рабочую область массив чисел из *mat* – файла, а команды (8.5) и (8.6) позволяют обратиться к столбцам массива данных, выделяя из них, например, массив времени *t* и массив значений *y*.

```
function grafik_Callback(hObject, eventdata, handles)

m = load (./param/59.mat); %загрузка файла из директории
cla; %Очистка осей перед построением
plot(m.ans(1,:),m.ans(2,:)); %построение графика
grid; %сетка
```

Рис. 8.10 Функция построение переходной характеристики *plot*

Полученные массивы времени *t* и значений *y* удобно использовать в функции *plot* (рис. 8.10) для вывода переходной характеристики на оси в окне GUI – приложения (рис. 8.11).

Рис. 8.11 GUI – приложение в работе

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 **Борцов Ю.А. и др.** Электромеханические системы с адаптивным и модальным управлением / Ю.А. Борцов, Н.Д. Поляхов, В.В. Путов. Л.: Энергоатомиздат. Ленингр. отд., 1984. 216 с.
- 2 **Косицын В.Ю., Рыбалев А.Н., Теличенко Д.А.** Д-адаптивная система для объекта с запаздыванием по управлению // Вестник Амурского гос. ун-та. – 2009. - №47
- 3 **Еремин Е.Л., Теличенко Д.А., Чепак Л.В.** Синтез адаптивных систем для скалярных объектов с запаздыванием по управлению. – Благовещенск: Амурский гос. ун-т, 2006. с. 50-54.
- 4 **Еремин Е. Л., Теличенко Д. А.** Адаптивное и робастное управление объектами теплоэнергетики. – Благовещенск: Амурский гос. ун-т, 2009. – 228 с.
- 5 **Лукас В.А.** Основы фазы-управления: Учебное пособие. – Екатеринбург: Изд-во УГГГА, 2000. – 62 с.
- 6 **Штовба С.Б.** Проектирование нечетких систем средствами MATLAB. – М.: Горячая линия – Телеком, 2007. – 288 с.
- 7 **Клюев А. С.** Проектирование систем автоматизации технологических процессов / А. С. Клюев, Б. В. Глазов, А. Х. Дубровский – М.: Энергия, 1980. – 512 с.
- 8 **Плетнев Г.П.** Автоматизированные системы управления объектами тепловых электростанций. – М.: Энергия, 1995. с. 350.
- 9 Контроллеры многофункциональные КР-300Ш, КР-300М, КР-300. Руководство по эксплуатации. Функциональные возможности. **КГЖТ.421457.001 РЭ 1.** / ЗАО «Волмаг». – Чебоксары: Издательство «Волмаг», 2004. – 68 с.
- 13 **Сметана А. З.** Автоматическая и автоматизированная настройка регуляторов теплоэнергетических процессов // Теплоэнергетика. – 2004. – № 11. – С. 47 – 52.
- 14 Экспериментальное исследование динамических характеристик физического объекта. Аппроксимация динамических характеристик: Методические

указания – М.: Изд-во Московского института химического машиностроения. Центральный институт комплексной автоматизации, 1992. – 16 с.

15. **Круглова В.В., Дли М.И., Голунова Р.Ю.** "Нечеткая логика и искусственные нейронные сети". – М.: Физматлит, 2001. - 224 с.

16. **Садомцев Ю.В.** Модели систем автоматического управления. Дискретные системы. – Саратов : Изд-во Сарат. техн. ун-та, 1998. – 100с.:ил.

17. **Олссон Г., Пиани Д.** Цифровые системы автоматизации и управления. – СПб.: Невский диалект, 2001 г. – 557 с.: ил.

18. **Еремин Е.Л., Косицын В.Ю., Теличенко Д.А.** Адаптивные алгоритмы для одного класса динамических объектов с запаздыванием по управлению

// ИСУ

19. Методичка по Современным системам управления

20. Ротач В.Я. Теория автоматического управления. – М.: Издательский дом МЭИ, 2008 г. – 396 с.

21. Монография ДА об автоматизации объектов теплоэнергетики

22. Ануфриев, Смирнов МАТЛАБ

23. Дьяков Учебный курс, 2001

24. Дьяков Симулинк

ПРИЛОЖЕНИЕ А
ВАРИАНТЫ ЗАДАНИЙ

Таблица А.1

Варианты заданий

Номер узла	Номер варианта									
	1	2	3	4	5	6	7	8	9	10
1	0.02	0.09	0.41	0.70	0.62	0.19	0.06	0.24	0.03	0.07
2	0.13	0.06	0.08	0.52	0.07	0.30	0.10	0.38	0.65	0.14
3	0.38	0.03	0.20	0.93	0.73	0.15	0.01	1.49	0.93	0.04
4	0.58	0.20	0.26	0.71	2.96	0.28	0.19	3.13	1.07	0.91
5	0.76	0.36	0.78	1.67	5.17	0.25	0.22	4.02	1.08	1.43
6	0.83	0.58	1.52	3.72	5.65	0.35	0.38	4.55	1.02	1.62
7	0.92	0.71	2.29	4.92	6.42	0.27	0.45	5.52	1.06	1.63
8	1.01	0.90	3.00	6.88	7.28	0.71	0.58	5.68	1.02	1.58
9	1.02	0.92	3.40	7.17	6.92	0.98	0.61	5.67	1.06	1.60
10	1.02	1.05	4.01	7.57	7.23	1.01	0.74	6.10	1.03	1.62
11	1.04	1.09	4.30	8.82	7.14	1.24	0.72	6.20	1.07	1.52
12	1.07	1.20	4.51	9.39	7.44	1.65	0.80	5.95	1.09	1.50
13	1.00	1.19	4.97	9.12	7.65	1.84	0.88	5.92	1.06	1.58
14	1.06	1.26	5.05	9.34	7.53	2.01	0.91	6.10	1.10	1.50
15	1.00	1.22	5.24	10.22	7.46	2.20	0.88	6.03	1.08	1.57
16	1.04	1.27	5.11	9.58	7.09	2.32	0.92	6.02	1.02	1.53
17	1.03	1.30	4.88	10.51	7.07	2.51	0.94	6.49	1.08	1.62
18	1.09	1.30	5.21	10.37	7.01	2.43	0.92	6.10	1.02	1.59
19	1.00	1.32	5.35	10.59	7.20	2.57	0.94	6.01	1.06	1.50
20	1.08	1.34	5.44	10.01	7.57	2.59	0.99	6.44	1.07	1.63

Таблица А.2

Варианты заданий

№ п/п	Номер варианта										
	11			12				13			
1	71.39	26	76.08	1	0	26	18.23	1	54.62	26	63.25
2	71.79	27	75.97	2	0	27	19.10	2	54.02	27	63.14
3	71.93	28	75.88	3	0.09	28	18.40	3	55.00	28	63.13
4	72.21	29	76.36	4	1.73	29	18.73	4	56.11	29	63.45
5	72.40	30	75.99	5	3.04	30	18.47	5	56.12	30	63.80
6	72.72	31	76.38	6	4.45			6	56.80	31	63.91
7	72.78	32	76.48	7	6.37			7	57.01	32	63.32
8	73.26	33	75.76	8	6.94			8	57.97	33	63.71
9	73.94	34	76.79	9	8.80			9	58.21	34	63.67
10	74.59	35	76.48	10	9.58			10	58.76	35	63.73
11	74.31	36	76.44	11	10.60			11	58.81	36	63.92
12	75.00	37	76.09	12	10.88			12	59.55	37	63.64
13	74.73	38	76.21	13	12.61			13	60.22	38	63.72
14	74.74	39	76.01	14	12.79			14	60.45	39	
15	75.08	40	76.04	15	14.07			15	60.63	40	
16	74.97	41	76.12	16	14.71			16	60.86	41	
17	75.20	42	76.43	17	15.31			17	60.95	42	
18	75.43	43	76.69	18	15.98			18	61.45	43	
19	75.04	44	76.60	19	16.36			19	61.81	44	
20	74.78	45	76.24	20	16.56			20	62.15	45	
21	75.57	46	75.91	21	17.11			21	62.04	46	
22	75.40	47	76.40	22	17.31			22	62.77	47	
23	75.85	48	76.22	23	17.48			23	62.81	48	
24	76.06	49	76.02	24	18.28			24	63.17	49	
25	75.60	50	76.47	25	18.47			25	63.16	50	

Для вариантов *1-10* разгонные кривые сняты на промежутке времени [0 100] с шагом 5 секунд.

Для варианта *11* - [0 700] с шагом 14 секунд.

Для варианта *12* - [0 450] с шагом 15 секунд.

Для варианта *11* - [0 252] с шагом 6 секунд.

ПРИЛОЖЕНИЕ Б

РАСПРЕДЕЛЕНИЕ КОРНЕЙ ПО БАТТЕРВОРТУ

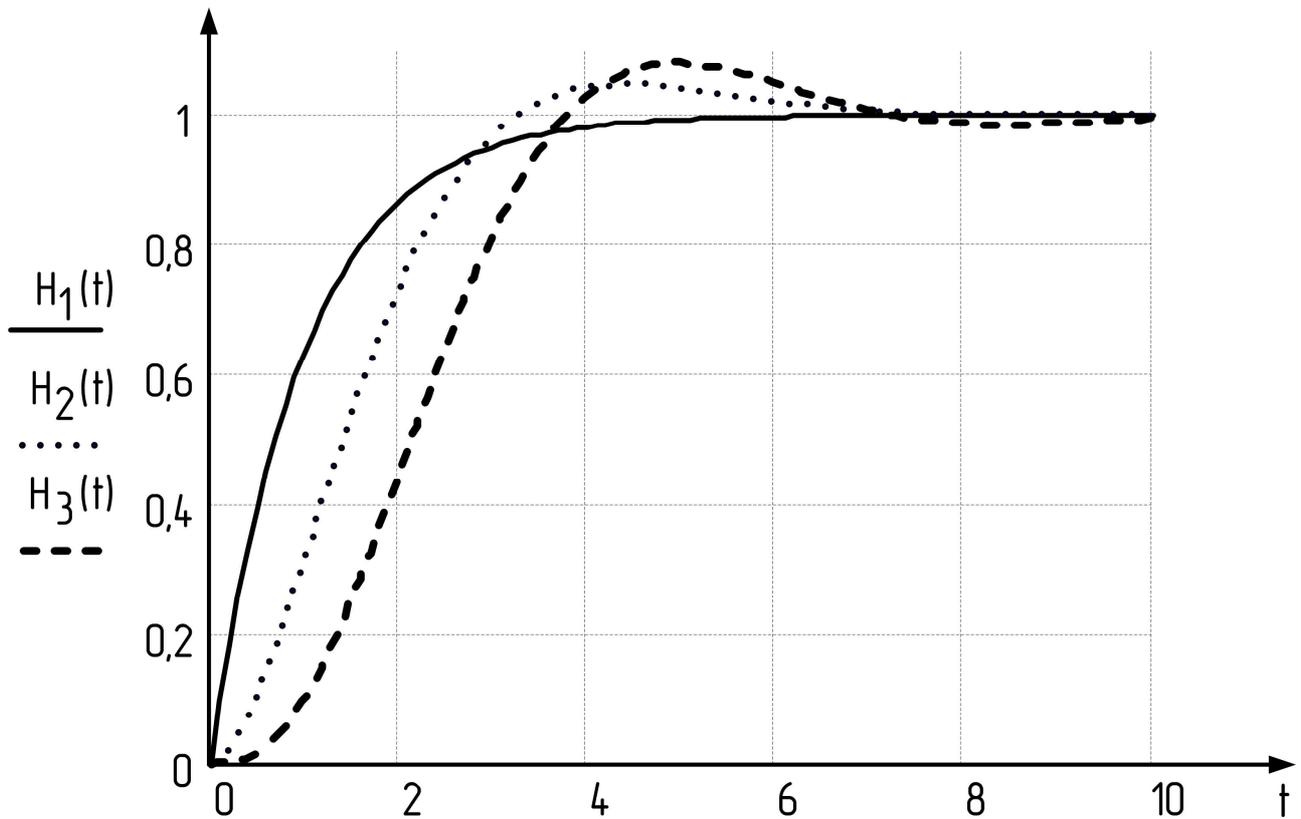


Рис. Б1 Распределение по Баттерворту

Таблица Б.1

Полиномы Баттерворта

Порядок полинома	Полином
$n = 1$	$H_1(p) = p + \omega_0$
$n = 2$	$H_2(p) = p^2 + 1,4 \cdot \omega_0 p + \omega_0^2$
$n = 3$	$H_3(p) = p^3 + 2 \cdot \omega_0 p^2 + 2 \cdot \omega_0^2 p + \omega_0^3$

ПРИЛОЖЕНИЕ В

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ №6

1. Система управления температурой за калорифером котла.
2. Система управления разряжения в топке котла.
3. Регулятор перегретого пара.
4. Регулятор общего воздуха.
5. Регулятор аэросмеси.
6. Регулятор уровня жидкости в резервуаре
7. Регулятор уровня в барабане котла
8. Регулятор топливно-подающего тракта.
9. Химическое производство
10. Регулятор общей нагрузки.

Для данных систем регулирования строится принципиальная электрическая схема. Тип датчика и его подключение задается преподавателем.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ЛАБОРАТОРНАЯ РАБОТА №1	
Математическое описание объектов управления.....	4
Краткие теоретические сведения	
1 Метод последовательного логарифмирования	5
2 Метод оценки параметров по кривой разгона.....	9
3 Представление объекта управления в пространстве состояний.....	12
4 Методы преобразования передаточных функций в уравнения в пространстве состояний.....	12
5 Представление объекта управления с запаздыванием по управлению в пространстве состояний.....	13
6. Случай с зашумленной кривой разгона.....	14
ЛАБОРАТОРНАЯ РАБОТА №2	
Классические системы регулирования.....	19
Краткие теоретические сведения	
1. Вопрос о выборе класса системы управления.....	19
2. Описание классической системы управления.....	21
3. Расчет классической системы управления.....	25
ЛАБОРАТОРНАЯ РАБОТА №3	
Адаптивная система регулирования с эталонной моделью.....	30
Краткие теоретические сведения	

1. Область применения адаптивных систем управления.....	30
2. Адаптивная система регулирования с эталонной моделью.....	30
3. Область применения параметрической и сигнальной настроек.....	33

ЛАБОРАТОРНАЯ РАБОТА №4

Адаптивная система регулирования с компенсацией запаздывания.....	37
---	----

Краткие теоретические сведения

1. Адаптивная система управления со стабилизирующим устройством.....	37
2. Диссипативная система управления.....	39

ЛАБОРАТОРНАЯ РАБОТА №5

Нечеткие системы управления.....	43
----------------------------------	----

Краткие теоретические сведения

1. Основные понятия нечеткой логики.....	43
2. Алгоритм нечеткого вывода Мамдани.....	46
3. Алгоритм нечеткого вывода Сугено.....	47
4. Структура нечеткого регулятора.....	48
5. Порядок синтеза фаззи – регулятора.....	49
6. Иллюстрация процедур фаззи-логики	50
7. Системы регулирования уровня жидкости в резервуаре.....	53
8. Недостатки систем с нечеткой логикой.....	57

ЛАБОРАТОРНАЯ РАБОТА №6

Реализация систем управления на микроконтроллерах.....	59
--	----

Краткие теоретические сведения	
1. Дискретно-непрерывное представление системы регулирования...	59
2. «Физическая» часть системы управления.....	61
3. «Логическая» часть системы управления.....	66
4. Пример реализация «логической» части системы управления.....	68
ЛАБОРАТОРНАЯ РАБОТА №7	
Учет шумов по каналу измерения выходной переменной.....	73
Краткие теоретические сведения	
1. Измерение выходной величины.....	73
2. Фильтрация выходной величины.....	74
ЛАБОРАТОРНАЯ РАБОТА №8	
Графический пользовательский интерфейс MATLAB.....	78
Краткие теоретические сведения	
1. Графический редактор.....	79
2. Создание приложения в среде GUIDE.....	82
3. Обработка событий приложения GUIDE.....	85
4. Взаимодействие Simulink – модели с GUI – приложением.....	88
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	91
ПРИЛОЖЕНИЕ А. Варианты заданий	93
ПРИЛОЖЕНИЕ Б. Распределение корней по Баттерворту	96
ПРИЛОЖЕНИЕ В. Задание к лабораторной работе №6	97
ПРИЛОЖЕНИЕ Г. Определению частотного показателя колебательности ..	98

Рыбалев Андрей Николаевич

зав. кафедрой АППиЭ , доцент кафедры АППиЭ АмГУ, канд. техн. наук

Денис Алексеевич Теличенко

доцент кафедры АППиЭ АмГУ, канд. техн. наук

Косицын Василий Юрьевич

«СОВРЕМЕННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ».

Пособие к выполнению лабораторных работ.

Изд-во АмГУ. Подписано к печати 16.11.10. Формат 60x84/16. Усл. печ. л. 6.28,
уч. - изд. л. 6.5. Тираж 100. Заказ 143.
Отпечатано в типографии АмГУ.