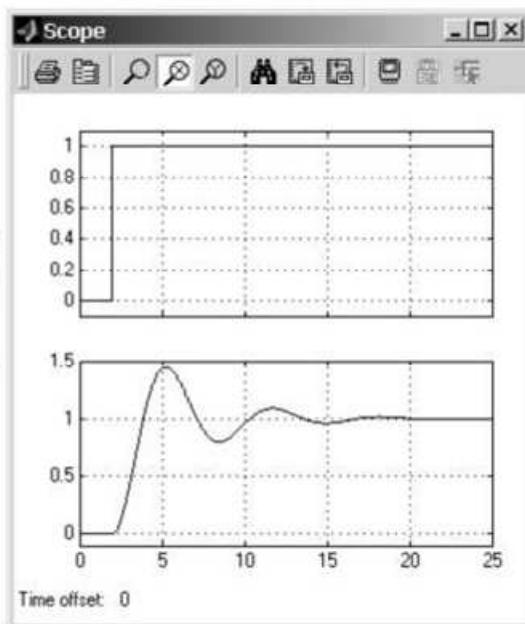
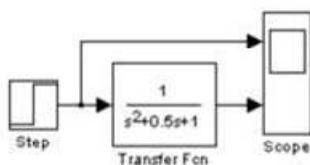


И.А. Елизаров  
Ю.Ф. Мартемьянов  
А.Г. Схиртладзе  
А.А. Третьяков

# МОДЕЛИРОВАНИЕ СИСТЕМ



Издательство ФГБОУ ВПО «ТГТУ»

Учебное издание

ЕЛИЗАРОВ Игорь Александрович,  
МАРТЕМЬЯНОВ Юрий Федорович,  
СХИРТЛАДЗЕ Александр Георгиевич,  
ТРЕТЬЯКОВ Александр Александрович

# МОДЕЛИРОВАНИЕ СИСТЕМ

Учебное пособие

Редактор Т.М. Г л и н к и н а  
Инженер по компьютерному макетированию М.А. Ф и л а т о в а

Подписано в печать 19.09.2011  
Формат 60 × 84/16. 5,58 усл. печ. л. Тираж 100 экз. Заказ № 384

Издательско-полиграфический центр ФГБОУ ВПО «ТГТУ»  
392000, г. Тамбов, ул. Советская, д. 106, к. 14

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Тамбовский государственный технический университет»

**И.А. ЕЛИЗАРОВ, Ю.Ф. МАРТЕМЬЯНОВ,  
А.Г. СХИРТЛАДЗЕ, А.А. ТРЕТЬЯКОВ**

# **МОДЕЛИРОВАНИЕ СИСТЕМ**

*Допущено Учебно-методическим объединением вузов  
по образованию в области автоматизированного машиностроения  
в качестве учебного пособия для студентов высших учебных заведений,  
обучающихся по направлению подготовки  
«Автоматизация технологических процессов и производств»*



---

Тамбов  
Издательство ФГБОУ ВПО «ТГТУ»  
2011

УДК 519.87  
ББК 22.18  
Е511

Рецензенты:

Кандидат технических наук, доцент ФГБОУ ВПО «ТГТУ»

*О.Г. Иванова*

Кандидат технических наук, доцент, первый заместитель  
генерального директора ОАО «Корпорация «Росхимзащита»

*С.Б. Путин*

Е511 Моделирование систем : учебное пособие / И.А. Елизаров,  
Ю.Ф. Мартемьянов, А.Г. Схиртладзе, А.А. Третьяков. – Тамбов :  
Изд-во ФГБОУ ВПО «ТГТУ», 2011. – 96 с. – 100 экз.  
ISBN 978-5-8265-1022-3

Содержит серию лабораторных работ для практического освоения методов математического моделирования и приобретения навыков проведения вычислительных (компьютерных) расчётов при создании систем и средств моделирования и соответствующих математических расчётов, работы по построению математических моделей на основе данных активного эксперимента, исследованию линейных стационарных систем управления, построению аналитических и экспериментально-аналитических моделей ряда типовых технологических процессов. Сопровождается необходимыми теоретическими положениями для выполнения лабораторных работ с использованием системы автоматизированных расчётов MATLAB+SIMULINK.

Предназначено для студентов высших учебных заведений, обучающихся по направлениям подготовки «Автоматизация технологических процессов и производств».

УДК 519.87  
ББК 22.18

**ISBN 978-5-8265-1022-3**

© Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет» (ФГБОУ ВПО «ТГТУ»), 2011

## ПРЕДИСЛОВИЕ

В подготовке студентов, специализирующихся в области построения автоматизированных систем управления технологическими процессами, одно из ведущих мест занимает курс «Моделирование систем». Эффективность освоения этого курса в значительной мере зависит от содержания и постановки лабораторного практикума.

В общем случае под термином «математическое моделирование автоматических систем» понимаются процессы отыскания их математических моделей, а также непосредственного исследования и анализа этих моделей на основе методов теории автоматического управления аналитически, графоаналитически или с использованием компьютеров. По мере развития вычислительной техники разработано достаточно много прикладных программ, обеспечивающих исследование переходных и установившихся процессов в автоматических системах любой сложности и практически в любых режимах работы при изменении их параметров и структуры. Среди них – пакеты MATLAB/SIMULINK, MATHCAD, VISSIM, LABVIEW и др.

Система MATLAB разработана фирмой The MathWorks, Inc. (США) и является системой инженерных и научных расчётов для различных областей науки и техники. Среди них:

- математика и вычисления;
- разработка алгоритмов;
- вычислительный эксперимент,
- имитационное моделирование;
- анализ данных, исследование и визуализация результатов;
- исследования в области автоматического управления;
- статистическая обработка сигналов и процессов и др.

Система MATLAB – это:

1. Интерактивная система, которая позволяет производить вычисления с помощью непосредственного ввода команд с клавиатуры;
2. Огромная библиотека готовых функций и алгоритмов, реализующих наиболее распространённые методы вычислений, которые для облегчения их поиска разбиты на специализированные разделы (Toolboxes);
3. Язык программирования, позволяющий пользователю создавать собственные функции, а также законченные программные приложения, в том числе и с использованием библиотечных функций системы.

Программа SIMULINK является расширением пакета MATLAB, достаточно самостоятельным его инструментом. При работе с SIMULINK практически не требуется знать сам MATLAB и остальные его приложения.

С другой стороны, доступ к функциям MATLAB и другим его инструментам остаётся открытым и их можно использовать в SIMULINK. Часть входящих в состав пакетов имеет инструменты, встраиваемые в

SIMULINK (например, LTI-Viewer приложения Control System Toolbox – пакета для разработки систем управления).

Как программное средство SIMULINK – типичный представитель визуально-ориентированных языков программирования. На всех этапах работы, особенно при подготовке моделей систем, пользователь практически не имеет дела с обычным программированием. Программа в кодах автоматически генерируется в процессе ввода выбранных блоков компонентов, их соединений и задания параметров компонентов.

При работе с SIMULINK пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков. При моделировании пользователь может выбирать метод решения дифференциальных уравнений. В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки SIMULINK. Результаты моделирования могут быть представлены в виде графиков или таблиц.

## 1. КРАТКОЕ ОПИСАНИЕ РАБОТЫ В MATLAB

### 1.1. РАБОТА В КОМАНДНОМ РЕЖИМЕ

Работа MATLAB возможна в двух режимах: командном режиме и режиме выполнения m-файла.

Работа системы в командном режиме напоминает работу в качестве калькулятора. В командной строке MATLAB после символа » можно сразу вводить исходные данные для вычисления с помощью встроенного текстового редактора. Ввод данных завершается нажатием клавиши ENTER.

Результаты вычислений выводятся с новой строки (без маркера »). Для блокировки вывода результатов вычислений некоторого выражения его надо закончить знаком ; (точка с запятой). Это бывает удобным для скрытия результатов промежуточных вычислений.

В качестве знака присваивания используется знак равенства =. Например, ввод строки  $x = 1$  означает объявление и инициализацию переменной  $x$ .

Имя переменной может содержать до 30 символов и должно не совпадать с именами функций, процедур системы и системных переменных. При этом система различает большие и маленькие буквы в переменных, т.е. переменные **aaa** и **AAA** в MATLAB обозначают разные переменные.

Система MATLAB имеет несколько имен переменных, которые используются самой системой и входят в состав зарезервированных:

- **i, j** – мнимая единица (корень квадратный из  $-1$ );
- **pi** – число  $\pi$  (сохраняется в виде 3.141592653589793);

- **inf** – обозначение машинной бесконечности;
- **NaN** – обозначение неопределённого результата (например, типа 0/0 или inf/inf);
- **eps** – погрешность операций над числами с плавающей запятой;
- **ans** – результат последней операции без знака присваивания;
- **realmax** и **realmin** – максимально и минимально возможные величины числа, которые могут быть использованы.

Эти переменные можно использовать в математических выражениях.

Встроенные функции записываются строчными буквами, а их аргументы приводятся в круглых скобках, например,  $\sin(x)$ .

В одном сеансе работы системы MATLAB можно определить несколько переменных и/или вычислить несколько выражений, например:

```

» a=1, b=5, c=a+b           Вводится с клавиатуры
a=
    1
b=
    5                       Ответ системы
c=
    6

```

Для вывода значения конкретной переменной достаточно ввести её имя, например:

```

» a                         Вводится с клавиатуры
a=                           Ответ системы
    1

```

Если вывод результата вычисления только что введённого выражения не заблокирован и не указана переменная, значение которой необходимо вывести, то система MATLAB сама назначает такую переменную с именем **ans**, присваивает ей значение последнего выражения и выводит её как результат вычислений, например:

```

» sin(pi/2)                Вводится с клавиатуры
ans=                       Ответ системы
    1

```

### 1.1.1. Числа, матрицы и векторы

Система MATLAB изначально создавалась как «матричный вычислитель» и оптимизирована для проведения вычислений с векторами, матрицами и массивами. Более того, в системе по умолчанию предполагается, что каждая переменная – это вектор или матрица. Например, если задано  $X = 1$ , то система воспринимает это как задание вектора с единственным элементом, значение которого равно 1. В свою очередь, вектор – это матрица, число столбцов которой равно 1.

Для задания вектора с большим числом элементов, их значения надо перечислить в квадратных скобках, используя в качестве разделителя пробел или запятую. Задание матриц осуществляется аналогично, для разделения строк используется знак ; (точка с запятой) Например:

» <b>X</b> = [1 2 3 4]	Задание вектора из 4 элементов
X =	Ответ системы
	1 2 3 4
» <b>Y</b> = [1 2; 4 5]	Задание матрицы 2×2
Y =	Ответ системы
	1 2
	3 4

В MATLAB предусмотрено наличие большого количества операций, которые можно совершать с матрицами и векторами. Среди них:

- сложение, вычитание (+,-);
- умножение (\*);
- обращение (**inv**);
- деление слева направо (/) и справа налево (\);
- возведение в степень (^);
- транспонирование (');
- создание нижней треугольной матрицы A: **tril(A)**;
- создание верхней треугольной матрицы A: **triu(A)**;
- формирование единичной матрицы заданного размера n: **eye(n)**;
- формирование матрицы единиц заданного размера  $n \times m$  : **ones(n,m)**. Для создания квадратной матрицы: **ones(n)**;
- формирование матрицы нулей заданного размера  $n \times m$  : **zeros(n,m)**. Для создания квадратной матрицы: **zeros(n)**;
- извлечение диагонали заданной матрицы A: **diag(A)**;
- вычисление следа матрицы A: **trace(A)**;
- собственные числа матрицы A: **eig(A)**;

### 1.1.2. Математические функции

В MATLAB предусмотрено наличие большого числа встроенных математических функций [1]. Среди них:

– *элементарные функции:*

- **sin(Z), cos(Z), tan(Z), cot(Z), sec(Z), csc(Z)** – синус, косинус, тангенс, котангенс числа Z;
- **sinh(Z), cosh(Z), tanh(Z), coth(Z), sech(Z), csch(Z)** – гиперболические синус, косинус, тангенс, котангенс, секанс, косеканс числа Z;
- **asin(Z), acos(Z), atan(Z), acot(Z), asec(Z), acsc(Z)** – арксинус (в радианах в диапазоне от  $-\pi/2$  до  $+\pi/2$ ), арккосинус (в диапазоне от 0 до  $\pi$ ),

арктангенс (в диапазоне от  $-\pi/2$  до  $+\pi/2$ ), арккотангенс, арксеканс, арккосеканс;

• **asinh(Z)**, **acosh(Z)**, **atanh(Z)**, **acoth(Z)**, **asech(Z)**, **acsch(Z)** – обратные гиперболические синус, косинус, тангенс, котангенс, секанс, косеканс;

- **exp(Z)** – экспонента числа Z;
  - **log(Z)** – натуральный логарифм;
  - **log10(Z)** – десятичный логарифм;
  - **sqrt(Z)** – квадратный корень из числа Z;
  - **abs(Z)** – модуль числа Z.
  - **round(Z)** – обычное округление числа Z к ближайшему целому;
  - **mod(X,Y)** – целочисленное деление X на Y;
  - **rem(X,Y)** – вычисление остатка от деления X на Y;
  - **sign(Z)** – вычисление сигнум-функции числа Z;
- *функции комплексного аргумента.*

Практически все вышеперечисленные элементарные математические функции вычисляются при комплексных значениях аргумента и получают в результате этого комплексные значения результата. Например, функция **sqrt** вычисляет, в отличие от других языков программирования, квадратный корень из отрицательного аргумента, а функция **abs** при комплексном значении аргумента вычисляет модуль комплексного числа.

В MATLAB есть несколько дополнительных функций, рассчитанных только на комплексный аргумент:

- **real(Z)** – выделяет действительную часть комплексного аргумента Z;
- **imag(Z)** – выделяет мнимую часть комплексного аргумента;
- **angle(Z)** – вычисляет значение аргумента комплексного числа Z (в радианах в диапазоне от  $-\pi$  до  $+\pi$ );
- **conj(Z)** – выдает число, комплексно сопряжённое относительно Z.

### 1.1.3. Решение системы линейных алгебраических уравнений

Рассмотрим систему линейных алгебраических уравнений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1; \\ a_{21}x_1 + a_{22}x_2 = b_2. \end{cases}$$

В матричной форме эта система может быть записана в виде  $\mathbf{AX} = \mathbf{B}$ , где матрицы **A**, **B**, **X** имеют вид:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}; \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Решение системы уравнений в матричной форме записи представляется выражением:

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B} ,$$

где  $\mathbf{A}^{-1}$  – матрица, обратная матрице  $\mathbf{A}$ .

В MATLAB это решение может быть найдено при выполнении выражений:

»  $\mathbf{X}=\text{inv}(\mathbf{A})*\mathbf{B}$

или

»  $\mathbf{X}=\mathbf{A}\backslash\mathbf{B}$ .

Пример. С использованием MATLAB найти решение системы линейных уравнений:

$$\begin{cases} x_1 + x_2 = 2; \\ x_1 + 2x_2 = 5. \end{cases}$$

Решение:

»  $\mathbf{A} = [1 \ 1; 1 \ 2];$

Задание матрицы A

»  $\mathbf{B} = [3; 5];$

Задание вектор-столбца B

»  $\mathbf{X}=\mathbf{A}\backslash\mathbf{B}$

Решение матричного уравнения

X =

Ответ системы

1  
2

#### 1.1.4. Операторы : (двоеточие) и . (точка)

Оператор : (двоеточие) используется для создания упорядоченной последовательности чисел с равноотстоящими значениями. Формат записи этого оператора:

**[Начальное\_значение]:[Шаг]:[Конечное\_значение]**

Если шаг не задан, то он принимается равным 1 или -1 в зависимости от того, больше конечное значение начального значения или меньше. При его использовании создаётся вектор:

»  $\mathbf{k} = 2:2:10$

Задаётся последовательность  $\mathbf{k}$  с шагом 2

k =

Ответ системы

2 4 6 8 10

»  $\mathbf{m} = 0:2$

Задаётся последовательность  $\mathbf{m}$  с шагом 1

m =

Ответ системы

0 1 2

»  $\mathbf{s}=\text{sin}(\mathbf{m})$

Использование последовательности  $\mathbf{m}$

s =

Ответ системы

0 0.8415 0.9093

При необходимости осуществления **почленного** умножения (или деления) элементов одного массива на элементы другого массива (той же размерности) используются соответственно операторы  $(.*)$  и  $(./)$ . Например:

» <b>m+s</b>	Алгебраическое сложение векторов <b>m, s</b>
ans =	Ответ системы
0 1.8415 2.9093	
» <b>m.*s</b>	Почленное умножение элементов <b>m, s</b>
j =	Ответ системы
0 0.8415 1.8186	

Для осуществления операции возведения в степень элементов массива используется операция " .^ ".

### 1.1.5. Построение графиков

Одной из самых привлекательных черт системы MATLAB является визуализация вычислений. Для построения графиков функций может использоваться функция **plot**.

Эта функция имеет различные формы записи в зависимости от набора входных аргументов. Если **y** – вектор, функция **plot(y)** производит построение кусочно-линейного графика элементов **y** в зависимости от индекса элементов **y**. Если функция **plot(x,y)** имеет два векторных аргумента одинаковой размерности, то будет построен график **y** в зависимости от **x**.

Функция **plot** может иметь несколько пар аргументов **x – y**. В этом случае она изображает несколько пар функций **y(x)** кривыми разного цвета, используя предопределённый (но изменяемый пользователем) список цветов. Например, **plot(x1, y1, x2, y2, x3, y3)** – выстраиваются три графика функции: **y1(x1), y2(x2), y3(x3)**.

Также имеется возможность определять цвет, стиль линии и маркеры с помощью ещё одного аргумента функции **plot**:

**plot(x,y,'color\_style\_marker')**

где **color\_style\_marker** – это символьная строка (ограниченная одиночными кавычками), характеризующая цвет, стиль линии и тип маркера:

- символы определения цвета: 'c', 'm', 'y', 'r', 'g', 'b', 'w', 'k'. Они соответствуют цветам cyan, magenta, yellow, red, green, blue, white, black;
- обозначения стиля линии: '-' сплошная, '--' пунктирная, ':' штриховая, '-.' штрихпунктирная;
- типы маркера: '+', 'o', '\*', 'x'.

Например, выражение: **plot(x, y, 'r--o')** вычерчивает красную пунктирную кривую и устанавливает маркеры «кружок» в каждой точке данных.

Любая из трёх перечисленных составляющих **color\_style\_marker** может быть пропущена. Если определяется тип маркера, а стиль линии опущен, система MATLAB строит только маркеры для этой кривой.

На график дополнительно можно нанести сетку из координатных линий, информацию о кривой (заголовок графика), а также обозначить координатные оси.

Нанесение координатной сетки на график осуществляется с помощью функции **grid**, к которой следует обратиться сразу после обращения к функции **plot**:

```
» plot(x, y), grid
```

Заголовок графика выводится с помощью процедуры **title**. Если после обращения к процедуре **plot** вызвать **title** таким образом:

```
title(' <текст> '),
```

то над графиком появится текст, записанный между апострофами в скобках. При этом следует помнить, что текст всегда должен помещаться в апострофы.

Аналогично можно вывести объяснения к графику, которые размещаются вдоль горизонтальной оси (функция **xlabel**) и вдоль вертикальной оси (функция **ylabel**).

Например, совокупность операторов

```
» x = -4*pi : pi/100 : 4*pi;
```

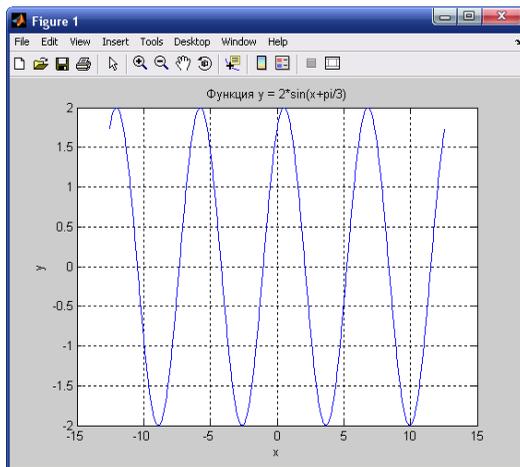
```
» y = 2* sin(x+pi/3);
```

```
» plot(x,y), grid;
```

```
» title('Функция y = 2*sin(x+pi/3)');
```

```
» xlabel('x'); ylabel('y');
```

приведёт к оформлению поля фигуры в виде:



## 1.2. ПРОГРАММИРОВАНИЕ В СРЕДЕ MATLAB

Работа в режиме калькулятора в среде MATLAB, несмотря на довольно значительные возможности, во многих отношениях неудобна. Невозможно повторить предшествующие вычисления и действия при новых

значениях исходных данных без повторного набора предшествующих операторов. Нельзя возвратиться назад и повторить некоторые действия, или по некоторому условию перейти к выполнению другой последовательности операторов.

Поэтому сложные алгоритмы вычислений, сопровождаемые переходами по определённым условиям с часто повторяемыми однотипными действиями, рациональнее оформлять в виде программ.

Создание программы в среде MATLAB осуществляется с помощью либо собственного встроенного, либо стороннего текстового редактора (например, Notepad среды Windows).

Программы на языке MATLAB имеют две разновидности – так называемые Script-файлы (файлы-сценарии, или управляющие программы) и файлы-функции (процедуры). Обе разновидности должны иметь расширение имени файла '.m'. С помощью Script-файлов оформляют основные программы, управляющие от начала до конца организацией всего вычислительного процесса, и отдельные части основных программ (они могут быть записаны в виде отдельных Script-файлов). Как файл-функции оформляются отдельные процедуры и функции (т.е. такие части программы, которые рассчитаны на неоднократное использование Script-файлами или другими процедурами при измененных значениях исходных параметров и не могут быть выполнены без предварительного задания значений переменных, которые называют входными) [1].

В дальнейшем под M-файлом будем понимать любой файл (файл-функцию или Script-файл), записанный на языке системы MATLAB.

Основные особенности записи текста программы (M-файла) [1]:

- Обычно каждый оператор записывается в отдельной строке текста программы. Окончание ввода строки осуществляется нажатием клавиши <Enter>.

- Можно размещать несколько операторов в одной строке, которые отделяются друг от друга символом ';' или ', '. Можно длинный оператор записывать в несколько строк. При этом предыдущая строка оператора должна заканчиваться тремя точками '... '.

- Если очередной оператор не заканчивается символом ';', результат его действия при выполнении программы будет выведен в командное окно. Чтобы предотвратить вывод на экран результатов действия оператора программы, запись этого оператора в тексте программы должна заканчиваться символом '; '. Если выражение завершается знаком ';', то MATLAB не выводит результирующее значение на экран. Обычно формат оператора с ';' используется при работе со средой в интерактивном режиме, а без ';' – при написании программы.

- Строка программы, начинающаяся с символа '%', не выполняется. Эта строка воспринимается системой MATLAB как комментарий.

➤ Строки комментария, предшествующие первому выполняемому оператору программы, т.е. такому, который не является комментарием, воспринимаются системой MATLAB как описание программы.

➤ В программах на языке MATLAB отсутствует символ окончания текста программы.

➤ В языке MATLAB переменные не описываются и не объявляются. Любое новое имя, появляющееся в тексте программы при ее выполнении, воспринимается системой MATLAB как имя матрицы. Размер этой матрицы устанавливается при предварительном вводе значений ее элементов либо определяется действиями по установлению значений ее элементов, описанными в предшествующих операторах или процедуре.

➤ В языке MATLAB невозможно использование матрицы или переменной, в которой предварительно не введены или не вычислены значения ее элементов (при выполнении программы MATLAB появится сообщение об ошибке – "Переменная не определена").

### 1.2.1. Создание простейших файл-функций

Файл-функция (процедура) должна начинаться со строки заголовка:

**function** [**<OutVar>**] = **<имя\_процедуры>**(**<InVar>**),

где **<OutVar>**, **<InVar>** – соответственно, перечни выходных и входных величин файл-функции. Если перечень выходных величин (**OutVar**) содержит только один объект (в общем случае – матрицу), то файл-функция представляет собой обычную функцию (одной или нескольких переменных). Первая строка в этом случае имеет вид:

**function** **<имя переменной>** = **<имя процедуры>**(**< InVar >**).

Если же в результате выполнения файл-функции должны быть определены (вычислены) несколько объектов (матриц), такая файл-функция представляет собой уже более сложный объект. Общий вид первой строки в этом случае становится таким:

**function** [**y1, y2, ... , y**] = **<имя процедуры>**(**< InVar >**),

т.е. перечень выходных величин **y1, y2, ... , y** должен быть представлен как вектор-строка с элементами **y1, y2, ... , y** (все они могут быть матрицами).

В простейшем случае функции одной переменной заголовок приобретет вид:

**function** **y** = **func**(**x**),

где **func** – имя функции (M-файла).

*Пример:* составить m-файл для вычисления функции

$$y = \ln(x + y) + \frac{x}{x + y} .$$

Процесс создания m-файла выглядит следующим образом:

1. Вызвать меню File командного окна MATLAB и выбрать в нём сначала команду New, а затем команду M-file.

2. В появившемся окне текстового редактора набрать текст:

```
function y = func(x,z)
```

```
y=log(x+z)+x./(x+z);
```

3. Сохранить этот текст в файле под именем func.m. Необходимый M-файл создан.

Теперь можно пользоваться этой функцией при расчётах. Например:

```
» y = func(1, 1)
```

```
y =
```

```
1.1931
```

```
» x1=0:1:5;
```

```
» x2=1:1:6;
```

```
» y=func(x1,x2)
```

```
y =
```

```
0 1.4319 2.0094 2.3745 2.6417 2.8524
```

При создании файл-функций необходимо помнить некоторые особенности:

- Во избежание вывода на экран промежуточных результатов, необходимо в тексте процедуры все вычислительные операторы завершать символом " ; ".

- С точки зрения области видимости переменных переменные в файл-функции являются *локальными*, поэтому имена переменных, используемые в файл-функции, могут не совпадать с именами соответствующих переменных при обращении к этой файл-функции.

- С целью использования в файл-функции глобальной переменной (некоторой переменной рабочего пространства (Workspace)) необходимо в диалоговом режиме (или в Script-файле) и файл-функции объявить эту переменную с атрибутом **global**. Если в одной строке объявляются несколько переменных как глобальные, они должны отделяться *пробелами*.

- Возможность использования файл-функции как для отдельных чисел, так и для векторов и матриц обусловлена применением в записи соответствующего M-файла вместо обычных знаков арифметических действий их аналогов с предшествующей точкой ' .' ('.\*', './', '.^' и др.).

### 1.2.2. Создание Script-файлов

При создании *Script-файлов* учитываются следующие особенности [1]:

- Script-файлы являются независимо (самостоятельно) исполняемыми блоками операторов и команд;

- все используемые переменные образуют так называемое рабочее пространство (Workspace), которое является общим для всех исполняемых

Script-файлов; из этого следует, что при выполнении нескольких Script-файлов имена переменных в них должны быть согласованы, так как одно имя означает в каждом из них один и тот же объект вычислений;

- в них отсутствует заголовок, т.е. первая строка определённого вида и назначения (в отличие от файл-функций, которые начинаются с заголовка **function**);

- обращение к ним не требует указания никаких имен переменных: все переменные формируются в результате выполнения программы либо сформированы ранее и существуют в рабочем пространстве.

### 1.2.3. Ввод и вывод информации в диалоговом режиме

Для обеспечения взаимодействия с пользователем в процессе выполнения m-файла используются такие команды:

**disp, sprintf, fprintf, input, menu, keyboard, pause.**

Команда **disp** осуществляет вывод значений указанной переменной или указанного текста в командное окно. Обращение к ней имеет вид:

**disp (<переменная или текст в апострофах>)**

Например:

**disp(x1)** – в командное окно выводится значение переменной *x1*.

**disp('value')** – в командное окно выводится текст *value*.

Чтобы вывести значения нескольких переменных в одну строку (например, при создании таблиц данных), нужно создать единый объект, который содержал бы все эти значения. Это можно сделать, объединив соответствующие переменные в вектор, пользуясь операцией создания вектора-строки:

**x = [x1 x2 ... x].**

Тогда вывод значений нескольких переменных в одну строку будет иметь вид:

**disp ([x1 x2 ... x]).**

Например:

» x1=-3.14; x2=-2.5; x3=5.6; x4=-9.33;

» disp([x1 x2 x3 x4])

-3.1400 -2.5000 5.6000 -9.3300

Для одновременного вывода символьной и цифровой информации в командное окно удобно использовать функцию **sprintf**, например:

» disp(sprintf('Параметр1 = %g Параметр2 = %g',x1,x2))

Параметр1 = -3.14 параметр2 = -2.5

Аналогично работает функция **fprintf**, где в качестве первого параметра для вывода информации в командное окно используется **1**, например:

» fprintf(1,'Параметр1 = %g Параметр2 = %g',x1,x2)

Параметр1 = -3.14 параметр2 = -2.5

Ввод информации с клавиатуры в диалоговом режиме можно осуществить с помощью функции **input**:

**x = input('<поясняющий\_текст>')**

При выполнении этой функции программа ожидает ввода информации с клавиатуры. По окончании ввода, которое определяется нажатием клавиши <Enter>, введённая информация запоминается в программе под именем "x", и выполнение программы продолжается.

Команда **pause** временно прекращает выполнение программы до тех пор, пока пользователь не нажмёт любую клавишу клавиатуры.

#### 1.2.4. Ветвление и циклы

В языке MATLAB для организации ветвления используются команды **if** и **switch**.

Общая форма записи оператора **if** имеет вид

```
if <условие 1>  
    <группа операторов>;  
[elseif <условие 2>]  
    [<группа операторов>; ]  
else  
    <группа операторов>;  
end
```

*Пример.* Разработать m-файл-функцию, осуществляющую определение знака числа.

$$\text{sgn}(x) = \begin{cases} 1, & x > 0; \\ 0, & x = 0; \\ -1, & x < 0. \end{cases}$$

Функция будет располагаться в m-файле **signum.m**.

```
function y = signum(x)  
if x > 0  
    y = 1;  
elseif x == 0  
    y = 0;  
else  
    y = -1;  
end
```

Команда **switch** позволяет осуществлять ветвление по нескольким условиям (направлениям) так же просто, как и с двумя, причём условия рассматриваются на равенство. Ниже представлен простой пример с разделением на три условия формирования параметра ввода.

```
function y = count(x)  
switch x
```

```

case 1
    y = 'one';
case 2
    y = 'two';
otherwise
    y = 'many';
end

```

Здесь выражение **switch** вычисляет параметр ввода **x**, а затем выполнение программы в файле переходит туда, где выражение **case** имеет то же самое значение, что и параметр ввода. Таким образом, если параметр ввода **x** имеет значение, равное 1, тогда параметр вывода **y** определяется в виде строки 'one', если **x** равен 2, тогда **y** определяется в виде строки 'two'. После выполнения всех команд, следующих за выбранным условием, программа MATLAB встречает либо другое выражение **case**, либо выражение **otherwise**, что приводит к переходу выполнения программы к выражению **end**.

В языке MATLAB для организации циклов используются команды **for** и **while**.

Организация цикла с помощью команды **for** выглядит следующим образом:

```

for n=n_0: [step:] n_k
    [тело цикла]
end

```

Цикл начинается с выражения **for** и заканчивается выражением **end**. Величины **n\_0** – начальное значение переменной цикла, **step** – шаг, **n\_k** – конечное значение переменной цикла.

*Пример.* Требуется вычислить 10!.

Фрагмент кода для определения факториала представлен ниже. Он может быть выполнен непосредственно в командном режиме

```

f=1;
for n=2:1:10
    f=f*n;
end
f

```

f=

3628800

Организация цикла с помощью команды **while** выглядит следующим образом:

```

while <условие>
    [тело цикла]
end

```

Цикл **while** продолжается до тех пор, пока <условие> истинно.

*Пример:* вычислить ряд  $\sum \frac{1}{n^2}$  с точностью  $10^{-9}$ .

```
n = 1;
oldsum = -1;
newsum = 0;
while (newsum-oldsum)>1e-9
    oldsum = newsum;
    newsum = newsum + 1/(n^2);
n = n + 1;
end
newsum

newsum =
1.6449
```

Фрагмент кода для вычисления ряда представлен ниже. Он может быть выполнен непосредственно в командном режиме.

Иногда бывает необходимо, чтобы программа MATLAB преждевременно вышла из цикла, например, при возникновении определённого условия. Для этой цели используется команда **break**.

## 2. КРАТКОЕ ОПИСАНИЕ РАБОТЫ В SIMULINK

### 2.1. СОЗДАНИЕ МОДЕЛЕЙ С SIMULINK

SIMULINK – это интерактивная система для анализа линейных и нелинейных динамических систем. Она может работать с линейными, нелинейными, непрерывными, дискретными, многомерными системами.

Перед построением модели необходимо предварительно загрузить систему MATLAB и запустить пакет SIMULINK.

Процесс построения модели SIMULINK включает в себя компоновку модели и задание необходимых параметров. Компоновка заключается в выборе из библиотек SIMULINK необходимых блоков, их размещение в открывшемся окне и соединение между собой. Далее для каждого блока устанавливаются соответствующие параметры, отвечающие требованиям моделируемой системы. Для их изменения надо дважды щёлкнуть на блоке и изменить нужные значения в диалоговом окне.

Блоки имеют названия, которые можно изменять, щёлкнув по нему левой клавишей мыши и отредактировав текст.

Блоки можно поворачивать. Для того, чтобы повернуть блок на 90 градусов, надо выделить его и нажать клавиши **Ctrl+R**. Комбинация **Ctrl+I** позволяет выполнить зеркальное отражение входов и выходов.

Блоки соединяются линиями связи, по которым распространяются сигналы. Для того, чтобы соединить блоки, надо щёлкнуть левой клавишей мыши по источнику сигнала и затем, при нажатой клавише **Ctrl**, по блоку-приёмнику. Можно также протянуть мышкой линию связи между нужными выходом и входом [2].

При создании ответвления от существующей соединительной линии, т.е. для соединения входного порта какого-либо блока с существующей линией, необходимо при выполнении соединения воспользоваться клавишей **Ctrl**. Также для создания точки разветвления в соединительной линии можно подвести курсор к предполагаемому узлу и, нажав правую клавишу мыши, протянуть линию.

Модель можно скопировать в буфер обмена в виде растрового рисунка. Для этого в окне модели надо выбрать в верхнем меню пункт **Edit / Copy model to clipboard**. Предварительно лучше уменьшить размеры окна до минимальных, чтобы не было белых полей.

Блок-диаграммы **SIMULINK** могут быть объединены в составные блоки, что позволяет использовать иерархическое представление структуры модели. Создание составного блока возможно двумя путями:

1. Перенести из библиотеки блоков блок **Subsystem** (составной блок) и, раскрыв его окно, сформировать его структуру в этом отдельном окне.

2. Выстроить блок-диаграмму в основном окне создаваемой модели, выделить её часть (предназначенную для объединения в составной блок) и, перейдя в меню, выбрать пункт **Edit/Create Subsystem**. В результате в основном окне автоматически образуется составной блок, в котором будет заключена та часть блок-диаграммы, что была выделена.

Для того, чтобы запустить моделирование, надо щёлкнуть левую клавишу мыши по кнопке  на панели инструментов. Эта же кнопка позволяет остановить моделирование при необходимости. Запуск модели на расчёт можно осуществить также через пункт меню **Simulation/Start**.

Предварительно перед запуском разработанной модели необходимо:

1. Установить параметры работы модели (интервал времени работы модели, метод интегрирования, погрешность и др.). Установка этих параметров осуществляется через пункт меню **Simulation/Parameters...** Самые важные параметры – это время моделирования (**Stop time**) и метод численного интегрирования уравнений (**Solver options**).

2. Сохранить разработанную модель (пункт **File/Save**).

**Библиотеки блоков SIMULINK.** Доступные блоки **SIMULINK** отображаются в окне **Simulink Library Browser**. Все блоки по функциональному назначению разделены по библиотекам. Среди них:

- *Continuous* – библиотека непрерывных элементов (интегратор, дифференциатор, линейная система обыкновенных дифференциальных уравнений и др.);

- *Discontinuities* – нелинейные элементы (реле, зона нечувствительности, люфт и др.);
- *Discrete* – библиотека дискретных элементов (интегратор с дискретным временем, дискретный фильтр и др.);
- *Math operations* – математические функции;
- *Logic and Bit operations* – логические и битовые операции;
- *Sinks* – средства отображения (временная диаграмма, вывод результатов в файл, остановка выполнения модели и т.д.);
- *Sources* – источники сигналов (генератор импульсных/синусоидальных сигналов, генератор случайных чисел, генератор пилообразных сигналов, часы и т.д.).

### Непрерывные линейные системы (Continuous)

$$\frac{1}{s+1}$$

**Transfer Fcn** – передаточная функция, в параметрах задаются числитель (**Numerator**) и знаменатель (**Denominator**) в виде полиномов.

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

**State Space** – модель в пространстве состояний, в параметрах задаётся четвёрка матриц, определяющих модель, и начальные условия для вектора состояния (**Initial conditions**).

$$\frac{(s-1)}{s(s+1)}$$

**Zero-Pole** – модель в форме «нули-полюса», в параметрах задаются массивы нулей (**Zeros**), полюсов (**Poles**), а также коэффициент усиления (**Gain**).

$$\frac{1}{s}$$

**Integrator** – интегратор с возможностью установки начальных условий (**Initial condition**), а также пределов насыщения (**Lower saturation limit** и **Upper saturation limit**). Когда сигнал выхода выходит за границы, определяемые этими пределами, интегрирование прекращается.



**Transport Delay** – блок фиксированной задержки сигнала.

### Нелинейные элементы (Discontinuities)



**Relay** – блок реализует релейную характеристику с гистерезисом.



**Saturation** – блок выполняет ограничение величины сигнала.



**Backlash** – блок моделирует нелинейность типа «люфт».



**Rate Limiter** – блок обеспечивает ограничение скорости изменения сигнала (первой производной).



**Dead Zone** – блок реализует нелинейную зависимость типа «зона нечувствительности (мертвая зона)».

### Основные источники сигналов (Sources)



**Constant** – сигнал постоянной величины.



**Step** – ступенчатый сигнал, меняется время скачка (**Step Time**), начальное (**Initial Value**) и конечное значения (**Final Value**).



**Ramp** – линейно возрастающий сигнал с заданным наклоном (**Slope**). Можно задать также время начала изменения сигнала (**Start Time**) и начальное значение (**Initial Value**).



**Pulse Generator** – генератор прямоугольных импульсов, задаются амплитуда (**Amplitude**), период (**Period**), ширина (**Pulse Width** в процентах от периода), фаза (**Phase Delay**).



**Repeating Sequence** – последовательность импульсов, их форма задаётся в виде пар чисел (время; величина сигнала).



**Sine Wave** – синусоидальный сигнал, задаётся амплитуда (**Amplitude**), частота (**Frequency**), фаза (**Phase**) и среднее значение (**Bias**).



**Signal Builder** – построитель сигналов, позволяющий задавать форму сигнала, перетаскивая мышью опорные точки.



**Random Number** – случайные числа с нормальным (гауссовым) распределением. Можно задать среднее значение (**Mean Value**), дисперсию (**Variance**), период изменения сигнала (**Sample Time**).



**Uniform Random Number** – случайные числа с равномерным распределением в заданном интервале от **Minimum** до **Maximum**.



**Band Limited White Noise** – случайный сигнал, ограниченный по полосе белый шум (имеющий равномерный спектр до некоторой частоты). Блок используется как источник белого шума для моделей непрерывных систем. Задаётся интенсивность (**Noise Power**) и интервал дискретизации (**Sample Time**), в течение которого удерживается постоянное значение сигнала. Чем меньше интервал, тем точнее моделирование, однако больше вычислительные затраты.

## Основные устройства вывода (Sinks)



**Display** – цифровой дисплей, показывает изменение входного сигнала в цифровом виде.



**Scope** – осциллограф, показывает изменение сигнала в виде графика, позволяет передавать данные в рабочую область MATLAB для последующей обработки и оформления.

## Другие часто используемые блоки

### Math Operations



**Gain** – усилитель, задаётся коэффициент усиления (**Gain**).



**Sum** – сумматор, используется для сложения и вычитания входов. Параметр **List of signs** задаёт количество входов, их знаки («+» для сложения и «-» для вычитания). Промежутки между входами (обозначаются знаком |).



**Trigonometric Function** – тригонометрическая функция.

### Signal Routing



**Manual Switch** – ручной переключатель, позволяет двойным щелчком переключать выход на один из двух входных сигналов.



**Mux** – мультиплексор, объединяет несколько сигналов в один «жгут» (векторный сигнал), в параметрах задаётся число входов (**Number of Inputs**).



**Demux** – демультиплексор, позволяет «разбить» векторный сигнал на несколько скалярных, в параметрах задаётся число выходов (**Number of Outputs**).

Лабораторная работа 1

## ИССЛЕДОВАНИЕ ЛИНЕЙНОЙ СТАЦИОНАРНОЙ ДИНАМИЧЕСКОЙ СИСТЕМЫ В СРЕДЕ MATLAB

*Цель работы:* освоение методов анализа одномерной линейной непрерывной системы с помощью среды MATLAB.

*Задачи работы:*

- ввести модель системы в виде передаточной функции;
- построить эквивалентные модели в пространстве состояний и в форме «нули-полюса-коэффициент усиления»;
- научиться строить импульсную и переходную характеристики, частотные характеристики;

- научиться использовать окно **LTI Viewer** для построения различных характеристик;
- научиться строить процессы на выходе линейной системы при произвольном входном сигнале.

*Содержание отчёта:* исследуемая передаточная функция, копия протокола работы (вводимые команды и результаты вычислений) в среде MATLAB (копируется через буфер обмена из рабочего окна среды MATLAB), графики динамических характеристик.

## КРАТКИЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

**Модели линейных систем.** Для описания линейных стационарных систем могут применяться несколько способов [2]:

- дифференциальные уравнения;
- модели в пространстве состояний;
- передаточные функции;
- модели вида «нули-полюса-коэффициенты передачи»;

Первые два способа описывают поведение системы во временной области и отражают внутренние связи между сигналами. Передаточные функции и модели вида «нули-полюса» относятся к *частотным* способам описания, так как непосредственно связаны с частотными характеристиками системы и отражают только вход-выходные свойства.

Частотные методы позволяют применять для анализа и синтеза алгебраические методы, что часто упрощает расчёты. С другой стороны, для автоматических вычислений более пригодны методы, основанные на моделях в пространстве состояний, поскольку они используют вычислительно устойчивые алгоритмы линейной алгебры.

Исходные уравнения динамики объектов, которые строятся на основе законов физики, в большинстве случаев имеют вид нелинейных дифференциальных уравнений. Для приближённого анализа и синтеза обычно проводят их линеаризацию в окрестности установившегося режима и получают линейные дифференциальные уравнения.

Линеаризованная модель объекта может быть описана дифференциальным уравнением вида:

$$\begin{aligned} a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_1 \frac{dy}{dt} + a_0 y = \\ = b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} y}{dt^{m-1}} + \dots + b_1 \frac{du}{dt} + b_0 u \end{aligned}$$

при нулевых начальных условиях. Причём  $m \leq n$ .

Модель в пространстве состояний связана с записью дифференциальных уравнений в стандартной форме Коши (в виде системы уравнений первого порядка):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u};$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u},$$

где  $\mathbf{x}$  – вектор переменных состояния размера  $n \times 1$ ;  $\mathbf{u}$  – вектор входных сигналов (вектор управления) размера  $m \times 1$  и  $\mathbf{y}$  – вектор выходных сигналов размера  $p \times 1$ . Кроме того,  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  и  $\mathbf{D}$  – постоянные матрицы. Согласно правилам матричных вычислений, матрица  $\mathbf{A}$  должна быть квадратной размера  $n \times n$ , матрица  $\mathbf{B}$  имеет размер  $n \times m$ , матрица  $\mathbf{C}$  –  $p \times n$  и матрица  $\mathbf{D}$  –  $p \times m$ . Для систем с одним входом и одним выходом матрица  $\mathbf{D}$  – скалярная величина.

*Передаточная функция*  $W(s)$  линейной стационарной системы от комплексной переменной  $s$  определяется как отношение преобразования Лапласа выхода к преобразованию Лапласа входа при нулевых начальных условиях:

$$W(s) = \frac{Y(s)}{U(s)}.$$

Для объекта, описываемого представленным выше дифференциальным уравнением, передаточная функция имеет вид:

$$W(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}.$$

По передаточной функции можно построить модель в форме «нули-полюса». Нулями называются корни числителя, полюсами – корни знаменателя.

**Динамические характеристики линейных систем.** Динамические свойства систем характеризуют реакции на входные воздействия специального вида. Динамические характеристики систем подразделяются на временные и частотные.

**Временные характеристики.** *Импульсной характеристикой* (весовой функцией)  $w(t)$  называется реакция системы на единичный бесконечный импульс (дельта-функцию или функцию Дирака) при нулевых начальных условиях. Дельта-функция  $\delta(t)$  определяется равенствами

$$\delta(t) = \begin{cases} 0, & t \neq 0, \\ \infty, & t = 0, \end{cases} \quad \int_{-\infty}^{\infty} \delta(t) dt = 1.$$

Это *обобщённая функция* – математический объект, представляющий собой идеальный сигнал, никакое реальное устройство не способно его воспроизвести. Дельта-функцию можно рассматривать как предел прямоугольного импульса единичной площади с центром в точке  $t = 0$  при стремлении ширины импульса к нулю.

*Переходной характеристикой* (переходной функцией)  $h(t)$  называется реакция системы (при нулевых начальных условиях) на единичный ступенчатый сигнал (единичный скачок):

$$1(t) = \begin{cases} 0, & t < 0; \\ 1, & t \geq 0. \end{cases}$$

Импульсная и переходная функции связаны выражениями

$$w(t) = \frac{dh(t)}{dt}; \quad h(t) = \int_0^t w(\tau) d\tau.$$

Для систем без интеграторов переходная характеристика стремится к постоянному значению. Переходная характеристика системы с дифференцирующим звеном (числитель передаточной функции имеет нуль в точке  $s = 0$ ) стремится к нулю. Если система содержит интегрирующие звенья, переходная характеристика асимптотически стремится к прямой, параболе и т.д., в зависимости от количества интеграторов.

По переходной характеристике можно найти важнейшие показатели качества системы – перерегулирование (*overshoot*) и время переходного процесса (*settling time*).

*Перерегулирование* определяется как

$$\sigma = \frac{h_{\max} - h_{\infty}}{h_{\infty}} \times 100\%,$$

где  $h_{\max}$  – максимальное значение функции  $h(t)$ , а  $h_{\infty} = \lim_{t \rightarrow \infty} h(t)$  – установившееся значение выхода.

*Время переходного процесса* – это время, после которого сигнал выхода отличается от установившегося значения не более, чем на заданную малую величину (в среде MATLAB по умолчанию используется точность 2%).

**Частотные характеристики.** Благодаря широкому применению при исследовании устойчивости динамических систем и проектировании регуляторов получили распространение частотные характеристики.

При подаче на вход линейной системы гармонического (синусоидального) сигнала  $u(t) = \sin(\omega t)$  с частотой  $\omega$  (она измеряется в радианах в секунду), на выходе будет также гармонический сигнал той же частоты, но другой амплитуды и фазы  $y(t) = A \sin(\omega t + \varphi)$ , где  $A$  – амплитуда и  $\varphi$  – сдвиг фазы.

Для построения частотной характеристики надо использовать подстановку  $s = j\omega$  в передаточной функции  $W(s)$ . Выражение  $W(j\omega)$  называется *частотной передаточной функцией* или *амплитудно-фазовой частотной характеристикой системы* (АФЧХ).

Зависимость модуля величины  $W(j\omega)$  от частоты называется *амплитудной частотной характеристикой* (АЧХ), а зависимость аргумента комплексного числа (фазы)  $W(j\omega)$  от частоты – *фазовой частотной характеристикой* (ФЧХ):

$$A(\omega) = |W(j\omega)|, \quad \varphi(\omega) = \arg W(j\omega) = \arctg \frac{\operatorname{Im} W(j\omega)}{\operatorname{Re} W(j\omega)}.$$

АЧХ показывает, насколько усиливается амплитуда сигналов разных частот после прохождения через систему, а ФЧХ характеризует сдвиг фазы сигнала.

На практике широкое применение получила диаграмма Бode (логарифмическая амплитудная характеристика – ЛАХ), которая определяется как  $L(\omega) = 20 \lg A(\omega)$ . Измеряется ЛАХ в децибелах и строится как функция от  $\lg(\omega)$ .

**Пакет CONTROL System Toolbox.** В состав системы MATLAB входит пакет прикладных программ (ППП) CONTROL System Toolbox, который предназначен для работы ЛТИ-моделями (Linear Time Invariant Models – линейные инвариантные во времени системы) систем управления. В русскоязычной литературе за такими системами закрепилось название – линейные стационарные системы.

В состав этого пакета входит более ста различных функций, в том числе функции создания и преобразования моделей, представленных в виде уравнений состояния, передаточных функций, функций для построения частотных и временных характеристик системы и др.

Для построения передаточной функции используется функция **tf**. Она имеет вид:

$$\text{tf}(B, A),$$

где  $B$  – вектор коэффициентов числителя передаточной функции, записанных по убыванию степеней;  $A$  – вектор коэффициентов знаменателя передаточной функции.

**Пример:** необходимо сформировать передаточную функцию

$$F(s) = \frac{2s + 4}{s^3 + 0,8s^2 + 2,06s + 2,26}.$$

В данном случае векторы коэффициентов числителя и знаменателя передаточной функции имеют вид:  $B=[2 \ 4]$ ,  $A=[1 \ 0.8 \ 2.06 \ 2.26]$ .

Процедура образования передаточной функции в диалоговом режиме имеет вид:

- »  $B=[2 \ 4]$ ;
- »  $A=[1 \ 0.8 \ 2.06 \ 2.26]$ ;
- »  $f=\text{tf}(B,A)$

После нажатия клавиши <ENTER> в памяти создаётся объект класса **tf**, описывающий передаточную функцию, и на экране появится передаточная функция в виде:

Transfer function:

$$2 s + 4$$

-----  
 $s^3 + 0.8 s^2 + 2.06 s + 2.26$

Определение передаточной функции  $F(s)$  можно также осуществить без предварительного построения числителя и знаменателя:

» `f=tf([2 4], [1 0.8 2.06 2.26])`

По передаточной функции можно легко построить модель в форме «нули-полюса-коэффициент передачи»:

» `f_zpk = zpk(f)`

Zero/pole/gain:

$$2 (s+2)$$

-----  
 $(s+1) (s^2 - 0.2s + 2.26)$

Нулями называются корни числителя, полюсами – корни знаменателя. Эта функция имеет один нуль в точке  $s = -2$  и три полюса в точках  $s = -1$  и  $s = -0,1 \pm 1,5i$ . Паре комплексных полюсов соответствует квадратный трёхчлен  $s^2 - 0,2s + 2,26$ .

Для определения нулей и полюсов передаточной функции  $F(s)$  используются соответственно функции **zero** и **pole**:

» `zero(f)`

ans =

$$-2$$

» `pole(f)`

ans =

$$0.1000 + 1.5000i$$

$$0.1000 - 1.5000i$$

$$-1.0000$$

Для преобразования передаточной функции  $F(s)$  в модель в пространстве состояний используется команда

» `f_ss = ss(f)`

a =

	x1	x2	x3
x1	-0.8	-1.03	-1.13
x2	2	0	0
x3	0	1	0

```

b =
      u1
  x1   2
  x2   0
  x3   0
c =
      x1      x2      x3
  y1   0      0.5    1
d =
      u1
  y1   0

```

Continuous-time model.

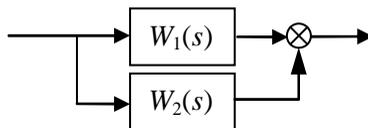
Это означает, что матрицы модели имеют вид

$$A = \begin{bmatrix} -0,8 & -1,03 & -1,13 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}, C = [0 \ 0,5 \ 1], D = 0.$$

На практике достаточно часто реальные системы автоматического управления (САУ) представляются структурно в виде соединённых между собой отдельных блоков (динамических звеньев), уравнения которых обычно достаточно просты.

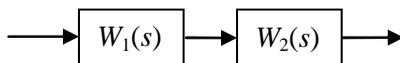
В MATLAB предусмотрена возможность программно строить схему САУ путём предварительного ввода моделей звеньев, составляющих САУ, и последующего соединения этих звеньев в единую структуру. К процедурам, осуществляющим расчёт характеристик соединений отдельных звеньев, относятся:

- **plus (minus)** или '+' ('-') – выполняет параллельное соединение указанных при обращении звеньев, т.е. определяет характеристики модели системы из параллельно соединённых звеньев, например  $W_1$  и  $W_2$ :



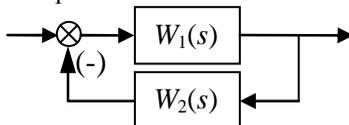
- **parallel** – осуществляет ту же процедуру параллельного соединения звеньев; в отличие от предыдущей процедуры может использоваться для многомерных систем и осуществления параллельного соединения лишь по некоторым входам и выходам;

- **mtimes** или '\*' – осуществляет последовательное соединение звеньев, имена которых указаны (применяется для одномерных систем):



• **series** – последовательное соединение (может использоваться для многомерных систем);

• **feedback** – соединение с обратной связью, т.е. такое соединение двух звеньев, когда второе указанное звено составляет цепь отрицательной обратной связи для первого звена:



*Пример:* Разработать модель замкнутой системы автоматического управления, в которой объект управления представляется в виде соединения трёх апериодических звеньев 1 порядка с передаточными функциями

$W_1 = \frac{2}{3s+1}$ ,  $W_2 = \frac{1}{2s+1}$ ,  $W_3 = \frac{1}{4s+1}$ , а в обратную связь включён регулятор с передаточной функцией  $R = 0,5 + \frac{0,2}{s}$ .

*Решение:*

```

»w1=tf([2],[3 1])           %формирование звена w1
»w2=tf([1],[2 1])         %формирование звена w2
»w3=tf([1],[4 1])         %формирование звена w2
»w=w1*w2*w3                %формирование объекта
»R=tf([0.5],[1])+tf([0.2],[1 0]) %формирование регулятора
»W_zs= feedback(w,R)       %формирование замкнутой системы
»step(W_zs)                % построение отклика системы

```

Пакет CONTROL System Toolbox предоставляет возможности по построению временных и частотных характеристик систем с использованием следующих функций (табл. 1):

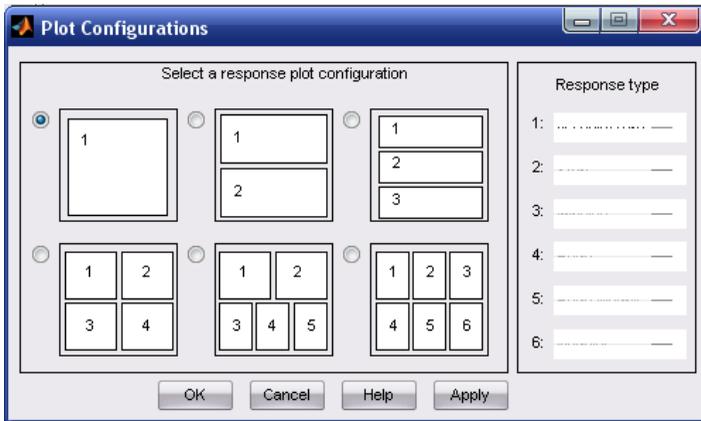
Таблица 1

Синтаксис	Описание
step(<LTI-объект>)	Построение графика переходной функции
impulse(<LTI-объект>)	Построение графика импульсной переходной функции
bode(<LTI-объект>)	Построение логарифмических частотных характеристик (диаграммы Боде)
nyquist(<LTI-объект>)	Построение частотного годографа Найквиста

Другим вариантом получения графиков динамических характеристик системы является использование графического окна – LTI Viewer. Вызов его осуществляется командой **ltiview**, которой, в качестве параметра, можно указать имя переменной, содержащей LTI-объект, например:  $\gg$  ltiview(f)

Загрузку исследуемого объекта можно осуществить также непосредственно из окна LTI Viewer, вызвав меню **File**, а затем команду **Import...**

Используя команду **Plot Configuration...**, из меню **Edit** можно выбрать конфигурацию области отображения графиков (количество и расположение окон построения графиков), а также виды выстраиваемых характеристик системы (Step, Impulse, Bode и др.) – раздел Response type (рис. 1).



**Рис. 1. Окно Plot Configuration**

Чтобы построить частотную характеристику в MATLAB можно использовать функцию **freqresp(f,w)**, где **f** – модель линейной системы, заданная в виде передаточной функции, в пространстве состояний или в форме «нули-полуса-коэффициент усиления», **w** – вектор частот.

Предварительно до использования этой функции необходимо создать массив частот в нужном диапазоне. Для этой цели могут быть задействованы функции **linspace** и **logspace**. Первая функция осуществляет равномерное распределение точек по линейной шкале, вторая – по логарифмической.

Процедура построения АЧХ системы на участке частот [0, 10] в диалоговом режиме имеет вид:

- $\gg$  w= linspace (0, 10, 100);
- $\gg$  freq = freqresp(f, w);
- $\gg$  freq = freq(:);
- $\gg$  plot (w, abs(freq));

Для вычисления фазы (в градусах) и построения её на графике могут использоваться команды:

- » `phi = angle(freq)*180/pi;`
- » `plot (w, phi);`

## ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Запустить среду MATLAB и в командном режиме очистить рабочее пространство и окно:

- » `clear all`
- » `clc`

2. Введите передаточную функцию как объект **tf** (коэффициенты полиномов передаточной функции вводятся в соответствии с вариантом (табл. 2), определяемым преподавателем):

- » `A = [a4 a3 a2 a1 a0]`
- » `B = [b2 b1 b0]`
- » `f = tf(B,A)`

3. Найдите нули, полюса передаточной функции и коэффициент усиления:

- » `z = zero(f)`
- » `p = pole(f)`

4. Постройте модель исходной системы в форме «нули-полюса»:

- » `f_zpk = zpk(f)`
- » `k = dcgain(f)`

5. Постройте модель системы в пространстве состояния:

- » `f_ss = ss(f)`

6. В командном режиме постройте переходную характеристику:

- » `step(f)`

7. В командном режиме постройте импульсную характеристику:

- » `impulse(f)`

8. Постройте логарифмические частотные характеристики (диаграммы Боде):

- » `bode(f)`

9. Постройте амплитудно-фазовую частотную характеристику (частотного годографа Найквиста):

- » `nyquist(f)`

10. Запустите графическое окно LTI Viewer:

- » `ltiview`

11. Загрузите объект **f**. Для этого в меню File необходимо выбрать пункт Import..., а далее выбрать объект **f**.

12. Постройте вышеперечисленные динамические характеристики, используя интерфейс LTI Viewer. По окончании работы с LTI Viewer закройте все окна за исключением командного окна MATLAB.

13. Постройте сигнал, имитирующий прямоугольные импульсы единичной амплитуды с периодом 10 секунд:

» `[u,t] = gensig('square',10);`

14. Выполните моделирование и постройте на графике сигнал выхода системы **f** при данном входном сигнале **u(t)**:

» `lsim(f, u, t)`

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие Вы знаете способы описания линейных стационарных динамических систем?

2. Задано дифференциальное уравнение системы:

$$\frac{d^2 y}{dt^2} + 2 \frac{dy}{dt} + y = 2u.$$

Представьте описание системы в виде передаточной функции; модели вида «нули-полюса»; уравнений состояния.

3. Представьте исследуемую вами систему в виде последовательно соединения типовых звеньев.

4. Постройте динамические характеристики типовых звеньев.

5. Постройте динамические характеристики типовых регуляторов (П, И, ПИ, ПИД, ПД).

## 2. Таблица вариантов

№	Вид передаточной функции	Коэффициенты полиномов							
		$b_2$	$b_1$	$b_0$	$a_4$	$a_3$	$a_2$	$a_1$	$a_0$
1	$W(s) = \frac{b_2 s^2 + b_1 s + b_0}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0}$	1	2	2	1	3	4	5	3
2		1	3	1	2	2	4	2	1
3		1	2	1	1	3	5	4	3
4		1	0	1	1	2	5	4	3
5		1	2	3	1	3	4	5	2
6	$W(s) = \frac{b_2 s^2 + b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0}$	1	2	1	-	2	5	4	3
7		2	1	1	-	2	4	5	1
8		1	3	2	-	1	5	2	3
9		2	1	2	-	1	5	5	2
10		1	3	3	-	1	5	4	2
11	$W(s) = \frac{b_2 s^2 + b_1 s}{a_3 s^3 + a_2 s^2 + a_1 s + a_0}$	1	1	-	-	2	5	4	3
12		1	3	-	-	2	4	5	1
13		3	1	-	-	1	5	2	3
14		1	4	-	-	1	5	5	2
15		1	2	-	-	1	5	4	2
16	$W(s) = \frac{b_2 s^2 + b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s}$	1	1	1	-	1	1	5	-
17		2	0	1	-	2	3	2	-
18		0	2	1	-	1	3	3	-
19		0	3	2	-	2	2	1	-
20		1	0	1	-	2	1	2	-

## МОДЕЛИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ В ПАКЕТЕ SIMULINK

*Цель работы:* освоение методов моделирования линейных систем в пакете SIMULINK

*Задачи работы:*

- научиться строить и редактировать модели систем управления в пакете SIMULINK;
- научиться изменять параметры блоков;
- научиться строить переходные процессы;
- оценить влияние настроечных параметров ПИД-регулятора на качественные показатели процесса регулирования в одноконтурной АСР.

*Содержание отчёта:* краткое описание исследуемой системы, схемы ПИД-регулятора и одноконтурной АСР из пакета SIMULINK, графики переходных процессов в одноконтурной АСР при изменении задания и отработке возмущения с различными типами регуляторов и соответствующие им настройки регуляторов, анализ влияния настроечных параметров на процесс регулирования.

### ОПИСАНИЕ МОДЕЛИРУЕМОЙ СИСТЕМЫ

В работе требуется провести исследование одноконтурной АСР с ПИД-регулятором. Её структурная схема показана на рис. 2.

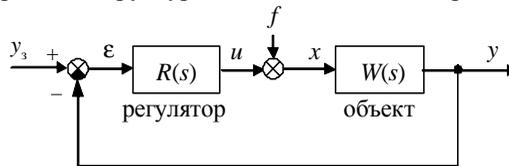


Рис. 2. Структурная схема одноконтурной АСР

Передаточная функция промышленных объектов во многих случаях с достаточной точностью может быть представлена в виде:

$$W(s) = \frac{K_{об} e^{-\tau s}}{(Ts + 1)(\sigma s + 1)},$$

где  $T, \sigma$  – соответственно, большая и меньшая постоянные времени объекта управления;  $K_{об}$  – коэффициент усиления объекта управления;  $\tau$  – время чистого запаздывания.

Передаточная функция ПИД-регулятора

$$W_p(s) = K_p + \frac{K_i}{s} + K_d s, \quad (1)$$

где  $K_p, K_i, K_d$  – соответственно, настроечные коэффициенты пропорциональной, интегральной и дифференциальной составляющей.

### ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Для запуска пакета SIMULINK щёлкните по кнопке  в командном окне MATLAB или введите команду **simulink** в командной строке.

2. Создайте новую модель с помощью верхнего меню открывшегося окна *Simulink Library Browser*. Для этого в меню **File** выберите пункт **New**, а затем **Model**, или воспользуйтесь сочетанием клавиш **Cntl-N**.

3. Перетащите блок **Gain** (усилитель) из окна *Simulink Library Browser* (группа **Math Operations**) в окно модели и дважды продублируйте его. Отредактируйте описание блоков –  $K_p, K_i, K_d$ .

4. Перетащите блоки **Integrator** (интегратор) и дифференциатор из окна *Simulink Library Browser* (группа **Continuous**) в окно модели.

5. Перетащите блок Sum (сумматор) из *Simulink Library Browser* (группа **Math Operations**) в окно модели и перейдите в окно редактирования его свойств (дважды нажав на левую клавишу мыши на изображении блока). Для свойства Icon shape установите rectangular (прямоугольный вид сумматора), а для свойства List of signs задайте +++ (три плюса), говорящие о том, что будет использовано три суммирующих входа.

6. Из *Simulink Library Browser*, соответственно из групп **Sources** и **Sinks** перетащить блоки **In1** и **Out1**.

7. Произведите соединение блоков в соответствии со схемой, представленной на рис. 3. Схема реализует ПИД-регулятор (1).

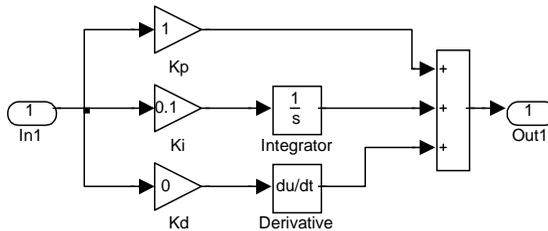


Рис. 3. Структура ПИД-регулятора

8. Создайте из нарисованной схемы, реализующей ПИД-регулятор, подсистему (Subsystem). Для этого выделите все элементы схемы, нажмите на выделенной области правую клавишу мыши и в появившемся окне выберите пункт **Create Subsystem**.

9. Отредактируйте название образовавшегося блока, назвав его PID.

10. Перетащите 2 блока **Transfer Fcn** (передаточная функция) и **Transport Delay** (Запаздывание) из окна *Simulink Library Browser* (группа

*Continuous*) в окно модели. Введите числитель и знаменатель передаточной функции в соответствии с номером варианта (табл. 3), а также время запаздывания Time Delay для блока **Transport Delay**.

Transfer Fcn1	Transfer Fcn2	Transport Delay
Numerator [K] Denominator [T 1]	Numerator [1] Denominator [ $\sigma$ 1]	Time Delay $\tau$

11. Перетащите 2 блока Sum (сумматор) из *Simulink Library Browser* в окно модели. Для организации отрицательной обратной связи отредактируйте свойство List of signs (задайте + -) одного из блоков.

12. Перетащите в окно модели блок **Scope** (осциллограф) из группы **Sinks** и установите его в правой части окна.

13. Перетащите в окно модели 2 блока **Step** из группы **Sources** и установите его слева от сумматора. Дайте название первому блоку Zадание, второму – Возмущение.

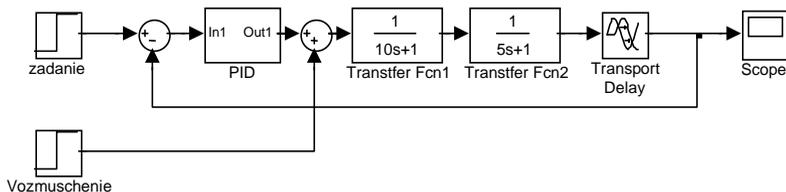
14. Установите в обоих блоках время подачи сигнала равное 0.

15. Произведите соединение блоков в соответствии со схемой, представленной на рис. 4. Схема реализует одноконтурную АСР.

16. Сохраните модель в своей папке под именем **lab2.mdl**

17. Установите время моделирования 100 секунд.

18. В блоке с именем Zадание (тип блока Step) установите величину подаваемого сигнала равную 1, а в блоке Возмущение – 0. При этом при выполнении моделирования будет имитироваться ситуация отработки изменения задания на 1. При отработке возмущения необходимо наоборот установить 0 для блока Zадание и 1 – для блока Возмущение.



**Рис. 4. Реализация одноконтурной АСР в пакете SIMULINK**

19. Выполните моделирование, нажав левой клавишей мыши на кнопке . Вид переходного процесса в одноконтурной АСР при изменении задания можно наблюдать в окне Scope.

20. Подберите настроечные параметры регулятора, обеспечивающие устойчивость замкнутой системы регулирования. Настройки регуля-

торов ( $K_p$ ,  $K_i$ ,  $K_d$ ) устанавливаются при редактировании блоков Gain в блоке PID.

21. Постройте график переходного процесса при обработке возмущения.

22. Проведите исследование одноконтурной АСР с различными типами регуляторов (П, ПИ, ПИД, ПД) с целью оценки влияния настроечных параметров регулятора на прямые показатели качества процесса регулирования (статическая и динамическая ошибки регулирования, величина перерегулирования, время переходного процесса), а также на устойчивость системы.

### 3. Таблица вариантов

№	$K_{об}$	$T$	$\sigma$	$\tau$	№	$K_{об}$	$T$	$\sigma$	$\tau$
1	1,5	10	5	2	11	1,5	13	2	3
2	2,0	15	4	3	12	2,0	12	3	4
3	3,0	17	3	4	13	3,0	17	4	5
4	3,5	18	4	2	14	3,5	14	4	3
5	3,2	12	5	3	15	3,2	11	5	2
6	0,5	13	7	4	16	0,5	16	3	3
7	1,2	14	6	3	17	1,2	15	4	2
8	2,5	11	3	2	18	2,5	14	5	4
9	1,7	18	3	3	19	1,7	16	4	5
10	2,1	12	4	4	20	2,1	17	3	3

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как запустить пакет SIMULINK?
2. Что такое Simulink Library Browser?
3. Какое расширение имеют файлы – модели SIMULINK?
4. Как создать новую модель?
5. Как создать подсистему (Subsystem)?
6. Как сделать, чтобы один и тот же сигнал поступал на несколько блоков?
7. Как скопировать (продублировать) блок в окне модели?
8. Как скопировать изображение модели в документ *Microsoft Word*?
9. Как изменить знаки арифметических действий в сумматоре?
10. Как ввести параметры блока Transfer Fcn (передаточная функция)?
11. Как изменить время моделирования?
12. Почему при использовании П- или ПД-регулятора система не компенсирует постоянное возмущение?

13. Какими свойствами должен обладать регулятор для того, чтобы постоянное возмущение полностью компенсировалось?

14. С помощью каких показателей качества оценивают переходный процесс в АСР?

### Лабораторная работа 3

## ПОСТРОЕНИЕ МАТЕМАТИЧЕСКОГО ОПИСАНИЯ ОБЪЕКТА УПРАВЛЕНИЯ ЭКСПЕРИМЕНТАЛЬНЫМИ МЕТОДАМИ

*Цели работы:* освоение методики построения математического описания объекта регулирования по результатам экспериментально полученных статических и динамических характеристик объекта.

*Задачи работы:*

- освоение методики аппроксимации экспериментальной статической характеристики объекта регулирования;
- освоение одной из методик аппроксимации экспериментальной динамической характеристики объекта регулирования передаточной функцией;
- закрепление навыков по созданию и использованию m-файлов;
- знакомство с методикой решения оптимизационных задач.

*Содержание отчёта:* исходные данные, результаты аппроксимации статической характеристики (аппроксимирующий полином, графики), результаты аппроксимации динамической характеристики (передаточная функция, графики).

### КРАТКИЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Математические модели объектов управления, процедура их построения зависят от их целевого назначения, свойств объекта, а также функций и задач системы управления.

При автоматизации производственных процессов наиболее правильные решения могут быть приняты на основании данных о статических и динамических свойствах регулируемого объекта. Для получения этих данных определяют соответственно статические и динамические характеристики регулируемых объектов.

Статические характеристики АСР или отдельных её звеньев представляют собой функциональную зависимость выходных величин от входных при статическом (стационарном, установившемся) режиме работы, т.е. при не изменяющихся во времени значениях входных и выходных величин.

Статические характеристики позволяют определить значения регулируемых параметров, положение регулирующих органов, расходы ве-

ществ или энергии и другие данные для любого установившегося состояния исследуемого объекта. Эти характеристики используют при расчётах АСР, расчётах оптимальных режимов, при оценке линейности объектов, при технологических и экономических расчётах.

Динамические характеристики представляют собой зависимости между изменениями входных и выходных величин в динамическом режиме (во времени). Они могут быть представлены в виде дифференциальных уравнений, передаточных функций, временных и частотных характеристик. Имея динамическую характеристику, представленную в каком-либо виде, нетрудно преобразовать её и представить в любом другом (из перечисленных выше) виде.

Динамические характеристики дают информацию об инерционных свойствах регулируемых объектов (систем, элементов систем) и таким образом являются исходными данными при синтезе автоматических систем регулирования. Они позволяют выполнить эту работу в полном объёме и завершить её расчётом параметров настройки на данном регулируемом объекте с целью получения желаемых форм графиков процессов регулирования.

Статические и динамические характеристики могут быть получены аналитическим, экспериментальным или экспериментально-аналитическим методом.

При построении систем автоматической стабилизации отдельных технологических параметров достаточно располагать динамическими и статическими характеристиками объекта управления в узком диапазоне изменения входных и выходных координат. Для получения этих характеристик применяются многочисленные экспериментальные методы. Эти методы базируются на предположении о линейности и сосредоточенности параметров объекта, неизменности во времени его статических и динамических характеристик. Принятие этих допущений позволяет достаточно просто описать изменение наблюдаемых выходных координат объекта линейными дифференциальными уравнениями с постоянными коэффициентами.

При построении экспериментальных математических моделей используют пассивный или активный эксперименты.

При активных методах на входе исследуемого объекта создаются искусственные испытательные воздействия (ступенчатые, импульсные, гармонические и др.). Пассивными методами называют такие, при которых не допускаются искусственные возмущения на входе исследуемого объекта, а в качестве входных испытательных воздействий используются естественные воздействия, возникающие в процессе нормальной эксплуатации.

**Аппроксимация статических характеристик.** При экспериментальном исследовании статических характеристик некоторого однонального объекта управления получено  $m$  значений входной  $x_i$  ( $i = \overline{1, m}$ ) и выходной  $y_i$  координат. Требуется найти функциональную зависимость  $\hat{y} = f(x, \mathbf{a})$ , наилучшим образом описывающую экспериментальные данные, где  $\mathbf{a}$  – вектор параметров функции. Для этой цели достаточно часто используется метод наименьших квадратов. Согласно этому методу вводится функция, характеризующая степень близости экспериментальных и расчётных данных:

$$S = \sum_i (y_i - \hat{y}_i)^2 .$$

При использовании метода наименьших квадратов задаются видом функции и вектором параметров этой функции. Часто в качестве аппроксимирующей функции  $\hat{y} = f(x, \mathbf{a})$  используют степенную функцию вида:

$$\hat{y} = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots$$

Задача поиска аппроксимирующей функции (вектора параметров) ставится и решается как задача безусловной многомерной оптимизации: требуется найти вектор параметров  $\mathbf{a}$ , обеспечивающий минимум функции  $S$ .

$$S = \sum_i (y_i - \hat{y}_i)^2 \rightarrow \min_{\mathbf{a}} .$$

При использовании пакета MATLAB полиномиальная аппроксимация данных измерений, которые сформированы как некоторый вектор  $\mathbf{Y}$ , при некоторых значениях аргумента, которые образуют вектор  $\mathbf{X}$  такой же длины, что и вектор  $\mathbf{Y}$ , осуществляется процедурой **polyfit(X, Y, n)**. Здесь  $n$  – порядок аппроксимирующего полинома. Результатом действия этой процедуры является вектор длиной  $(n + 1)$  из коэффициентов аппроксимирующего полинома.

**Аппроксимация динамических характеристик.** В инженерной практике при настройке АСР математическое описание объекта регулирования достаточно часто представляется в виде передаточных функций. Передаточная функция объекта регулирования может быть определена в результате аппроксимации экспериментальной динамической характеристики, полученной в результате проведения активного эксперимента. В качестве испытательного воздействия чаще всего используется ступенчатое воздействие.

Алгоритм определения аппроксимирующей передаточной функции объекта регулирования имеет вид:

- При подаче на вход объекта ступенчатого воздействия на выходе снимается переходная характеристика  $h(t)$ .

- По виду кривой определяется структура передаточной функции. Для большого числа реальных объектов регулирования модель объекта может быть представлена в виде:

$$W(s) = \frac{ke^{-s\tau}}{(T_1s+1)(T_2s+1)}. \quad (2)$$

- Аппроксимация переходной характеристики объекта передаточной функцией. Аппроксимация заключается в подборе таких параметров передаточной функции, которые обеспечивают максимальную близость экспериментальной кривой разгона  $h(t)$  и кривой разгона, соответствующей передаточной функции (2) –  $h_p(t)$ .

Настройные параметры  $k, T_1, T_2, \tau$  передаточной функции (2) могут быть получены при минимизации функции

$$I = \int (h(t) - h_p(t))^2 dt \rightarrow \min_{k, T_1, T_2, \tau}. \quad (3)$$

## ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

### Аппроксимация статической характеристики

1. Запустите среду MATLAB. В командном режиме очистите рабочее пространство и окно:

» clear all

» clc

2. Сформируйте массивы экспериментальных точек – векторы X и Y (в соответствии с вариантом (табл.4), определяемым преподавателем).

» X = [x1 x2 x3 x4 x5 x6 x7 x8 x9 x10];

» Y = [y1 y2 y3 y4 y5 y6 y7 y8 y9 y10];

3. Произведите аппроксимацию экспериментальных точек полиномом первой степени (линейной функцией) и второй степени и определите векторы коэффициентов полиномов – соответственно P1 и P2.

» P1 = polyfit(X,Y,1)

» P2 = polyfit(X,Y,2)

4. Используя функцию polyval, постройте аппроксимирующие полиномы.

» F1 = polyval(P1,X)

»  $F2 = \text{polyval}(P2, X)$

5. На одном рисунке постройте экспериментальные точки, а также аппроксимирующие полиномы первой и второй степени

» `plot(X, Y, 'o', X, F1, '-.', X, F2, '-')`

### Аппроксимация динамической характеристики

Исходными данными для аппроксимации являются данные об экспериментальной переходной характеристике, представленной в табличной форме, например:

T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
h	0	0	0	0.07	0.23	0.43	0.64	0.84	1.03	1.19	1.33	1.45	1.55	1.64	1.71	1.77

Вариант исходных данных определяется преподавателем (табл. 5).

4. Создайте Script-файл для решения задачи (3). Для этого вызовите меню File командного окна MATLAB и выберите в нём сначала команду New, а затем команду M-file.

5. В появившемся окне текстового редактора наберите текст:

```
% t – Массив моментов времени
% h – массив значений экспериментальной переходной характеристики
% hp – массив значений рассчитанной переходной характеристики
% a – вектор параметров передаточной функции объекта
% a(1)=k, a(2)=t1, a(3)=t2, a(4)=tau
global t h hp
t=0:1:15;
h=[0 0 0 0.07 0.23 0.43 0.64 0.84 1.03 1.19 1.33 1.45 1.55 1.64 1.71 1.77];
[a,fval] = fminunc(@func,[1 1 2 1])
plot(t,h,'o',t,hp,'-')
```

Процедура **fminunc** осуществляет поиск минимума функции нескольких переменных. Имя минимизируемой функции задаётся в первом аргументе процедуры **fminunc**, а во втором – передаётся начальная точка для поиска.

6. Сохраните этот текст в файле под именем `din_app.m`.

7. Создайте файл-функцию, высчитывающую минимизируемую функцию (3). Текст файл-функции имеет вид:

```
function f=func(a)
global t h hp
y=series(tf([a(1)],[a(2) 1]),tf([1],[a(3) 1]));
y.inputd = a(4);
[hp,t]=step(y,t);
f=sum((h-hp).^2);
```

В этой функции строки  
`y=series(tf([a(1)],[a(2) 1]),tf([1],[a(3) 1]));`  
`y.inputd = a(4);`

определяют передаточную функцию объекта в соответствии с (2). Здесь  $a(1)$ ,  $a(2)$ ,  $a(3)$ ,  $a(4)$  соответственно  $k, T_1, T_2, \tau$ .

Строка `[hp,t]=step(y,t);` строит отклик на ступенчатое воздействие.

Последняя строка определяет сумму квадратов невязок между экспериментальной и рассчитанной переходными характеристиками.

8. Сохраните введённый текст в файле под именем `func.m`.

9. В командном режиме запустите на исполнение Script-файл:

» `din_app`

10. При завершении выполнения этого Script-файла в командном окне MATLAB будет отображён результат решения задачи минимизации функции (3) в виде:

`a =`

2.0382 2.4265 4.8308 2.0181

`fval =`

3.9104e-005

Также на графике (рис. 5) будут представлены экспериментально снятые точки переходной характеристики и рассчитанная аппроксимирующая переходная характеристика.

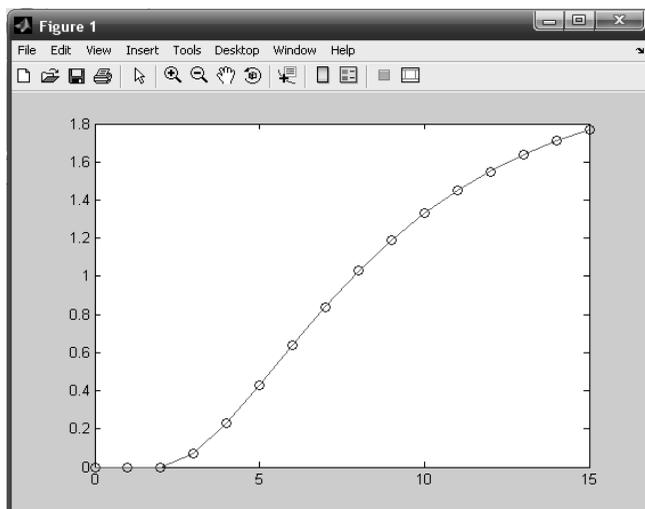


Рис. 5. Результаты аппроксимации переходной характеристики

#### 4. Таблица вариантов экспериментальных статических характеристик

1	$x_i$	0	1	2	3	4	5	6	7	8	9
	$y_i$	3.0	5.1	7.4	9.9	12.6	15.5	18.6	21.9	25.4	29.1
2	$x_i$	0	1	2	3	4	5	6	7	8	9
	$y_i$	6.0	7.9	9.6	11.1	12.4	13.5	14.4	15.1	15.6	15.9
3	$x_i$	0	1	2	3	4	5	6	7	8	9
	$y_i$	4.0	6.8	9.2	11.2	12.8	14.0	14.8	15.2	15.2	14.8
4	$x_i$	0	1	2	3	4	5	6	7	8	9
	$y_i$	5.0	10.1	14.4	17.9	20.6	22.5	23.6	23.9	23.4	22.1
5	$x_i$	0	1	2	3	4	5	6	7	8	9
	$y_i$	35.0	34.3	33.0	31.1	28.6	25.5	21.8	17.5	12.6	7.1
6	$x_i$	0	2	4	6	8	10	12	14	16	18
	$y_i$	3.0	5.1	7.4	9.9	12.6	15.5	18.6	21.9	25.4	29.1
7	$x_i$	0	2	4	6	8	10	12	14	16	18
	$y_i$	6.0	7.9	9.6	11.1	12.4	13.5	14.4	15.1	15.6	15.9
8	$x_i$	0	2	4	6	8	10	12	14	16	18
	$y_i$	4.0	6.8	9.2	11.2	12.8	14.0	14.8	15.2	15.2	14.8
9	$x_i$	0	2	4	6	8	10	12	14	16	18
	$y_i$	5.0	10.1	14.4	17.9	20.6	22.5	23.6	23.9	23.4	22.1
10	$x_i$	0	2	4	6	8	10	12	14	16	18
	$y_i$	35.0	34.3	33.0	31.1	28.6	25.5	21.8	17.5	12.6	7.1
11	$x_i$	1	2	3	4	5	6	7	8	9	10
	$y_i$	3.0	5.1	7.4	9.9	12.6	15.5	18.6	21.9	25.4	29.1
12	$x_i$	1	2	3	4	5	6	7	8	9	10
	$y_i$	6.0	7.9	9.6	11.1	12.4	13.5	14.4	15.1	15.6	15.9
13	$x_i$	1	2	3	4	5	6	7	8	9	10
	$y_i$	4.0	6.8	9.2	11.2	12.8	14.0	14.8	15.2	15.2	14.8
14	$x_i$	1	2	3	4	5	6	7	8	9	10
	$y_i$	5.0	10.1	14.4	17.9	20.6	22.5	23.6	23.9	23.4	22.1
15	$x_i$	1	2	3	4	5	6	7	8	9	10
	$y_i$	35.0	34.3	33.0	31.1	28.6	25.5	21.8	17.5	12.6	7.1

5. Таблица вариантов экспериментальных динамических характеристик

1	T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	h	0	0	0	0,07	0,23	0,43	0,64	0,84	1,03	1,19	1,33	1,45	1,55	1,64	1,71	1,77
2	T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	h	0	0	0	0,09	0,29	0,53	0,77	0,99	1,19	1,35	1,49	1,61	1,69	1,76	1,82	1,86
3	T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	h	0	0	0	0,13	0,43	0,79	1,15	1,49	1,78	2,03	2,23	2,40	2,54	2,64	3,72	2,79
4	T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	h	0	0	0,14	0,46	0,84	1,19	1,52	1,81	2,05	2,24	2,40	2,52	2,63	2,71	2,79	2,82
5	T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	h	0	0	0,098	0,31	0,56	0,80	1,02	1,21	1,36	1,49	1,60	1,69	1,75	1,81	1,85	1,88
6	T	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
	h	0	0	0	0,07	0,23	0,43	0,64	0,84	1,03	1,19	1,33	1,45	1,55	1,64	1,71	1,77
7	T	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
	h	0	0	0	0,09	0,29	0,53	0,77	0,99	1,19	1,35	1,49	1,61	1,69	1,76	1,82	1,86
8	T	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
	h	0	0	0	0,13	0,43	0,79	1,15	1,49	1,78	2,03	2,23	2,40	2,54	2,64	3,72	2,79
9	T	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
	h	0	0	0,14	0,46	0,84	1,19	1,52	1,81	2,05	2,24	2,40	2,52	2,63	2,71	2,79	2,82
10	T	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
	h	0	0	0,098	0,31	0,56	0,80	1,02	1,21	1,36	1,49	1,60	1,69	1,75	1,81	1,85	1,88
11	T	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5
	h	0	0	0	0,07	0,23	0,43	0,64	0,84	1,03	1,19	1,33	1,45	1,55	1,64	1,71	1,77
12	T	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5
	h	0	0	0	0,09	0,29	0,53	0,77	0,99	1,19	1,35	1,49	1,61	1,69	1,76	1,82	1,86
13	T	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5
	h	0	0	0	0,13	0,43	0,79	1,15	1,49	1,78	2,03	2,23	2,40	2,54	2,64	3,72	2,79
14	T	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5
	h	0	0	0,14	0,46	0,84	1,19	1,52	1,81	2,05	2,24	2,40	2,52	2,63	2,71	2,79	2,82
15	T	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5
	h	0	0	0,098	0,31	0,56	0,80	1,02	1,21	1,36	1,49	1,60	1,69	1,75	1,81	1,85	1,88

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что позволяют определить статические характеристики объекта?
2. Как осуществить линеаризацию статической характеристики?
3. Какие виды динамических характеристик вы знаете?
4. Какая взаимосвязь между импульсной и переходной характеристиками?
5. Какие методы аппроксимации динамических характеристик вы знаете?
6. Чем отличается файл-функция от Script-файла в пакете MATLAB?
7. Какая функция может быть использована для проведения аппроксимации полиномом  $n$ -й степени?
8. Что осуществляет функция `polyval` в пакете MATLAB?
9. Какая функция в пакете MATLAB может быть использована для поиска минимума функции нескольких переменных?

### Лабораторная работа 4

## ПОСТРОЕНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ АНАЛИТИЧЕСКИМ МЕТОДОМ (МОДЕЛИРОВАНИЕ ТЕПЛООБМЕННОЙ АППАРАТУРЫ)

*Цель работы:* ознакомление с методами моделирования процесса теплообмена через стенку и с расчётом теплообменных аппаратов на ЭВМ.

*Задания работы:*

1. Ознакомиться с принципами составления тепловых балансов потоков и разделяющей стенки в теплообменных и реакционных аппаратах химической технологии при теплообмене через стенку.
2. Отметить особенности математических описаний при разных способах организации теплообмена.
3. Выполнить вариант задания на ЭВМ.

*Содержание отчёта:*

1. Вариант задания.
2. Математическая модель теплообменного аппарата.
3. Все расчётные формулы, используемые в работе.
4. Графики рассчитанных зависимостей для теплоносителей.
5. Тексты программ.

## КРАТКИЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Процесс передачи тепла через стенку весьма распространён в химической технологии и значительно влияет на протекание химических реакции во всех типах реакторов. Процесс передачи тепла в теплообменной

аппаратуре является основным и служит для сообщения технологическому потоку нужной температуры.

Выбирая различные способы оформления реакторов, можно влиять на интенсивность теплообмена между основным (реакционным) потоком и потоком хладагента или окружающей средой. При полном отсутствии теплообмена через стенку получают адиабатический реактор. Реакторы, имеющие теплообмен с внешней средой, относятся к политропическим.

При рассмотрении процесса передачи тепла от одного теплоносителя к другому через стенку можно выделить несколько элементарных этапов: переход тепла от горячего теплоносителя к более холодной стенке, поглощение тепла материалом стенки и её нагрев, распределение тепла по объёму стенки, переход тепла от стенки к холодному теплоносителю.

Если процесс теплообмена протекает стационарно, то температура в каждой точке материала (теплоносителей и стенки) не изменяется во времени. Применение модели с сосредоточенными параметрами (т.е. когда пространственные координаты не входят в математическое описание) приводит к алгебраическим соотношениям между температурами в системе. Если, наоборот, температуры меняются во времени, математическое описание получается в виде системы обыкновенных дифференциальных уравнений (аргументом является время).

Зависимость температур от геометрических координат обуславливает математическое описание статики в виде обыкновенных дифференциальных уравнений (если пространственная координата одна) или дифференциальных уравнений в частных производных. Независимыми переменными при этом являются пространственные координаты. Динамическая модель при наличии пространственно-распределённых эффектов описывается уравнениями в частных производных, причём одной из независимых переменных является время.

Интенсивность перехода тепла от одного теплоносителя (например, горячего потока жидкости или газа) к другому (стенке) зависит от разности температур между ними, а также от теплового сопротивления. В расчётные уравнения, однако, обычно включают не сопротивление, а обратную величину  $\alpha$  – коэффициент теплоотдачи,  $q$  – тепловой поток (ккал/ч или Вт) через поверхность площадью  $1 \text{ м}^2$  при разности температур (температурном напоре)  $1 \text{ }^\circ\text{C}$ .

Полный тепловой поток  $q$  определяется произведением коэффициента теплоотдачи  $\alpha$ , площади поверхности  $F$  и температурного напора  $\Delta T$ :

$$q = \alpha F \Delta T . \quad (4)$$

Уравнение (4) применимо как к нагреванию стенки от горячей жидкости, так и, наоборот, к нагреванию холодной жидкости горячей стенкой; при этом  $\Delta T$  будет иметь разные знаки.

Если пренебречь распространением тепла в стенке, то теплопередачу от горячего потока жидкости к холодному, находящемуся по другую сторону стенки, можно представить как процесс преодоления тепловым потоком двух последовательных сопротивлений теплоотдачи – от горячего потока к стенке и от нагретой стенки к холодному потоку.

Используя вместо сопротивлений коэффициенты теплоотдачи ( $\alpha_1$  и  $\alpha_2$ ), получаем выражение, определяющее коэффициент теплопередачи ( $K$ ):

$$\frac{1}{K} = \frac{1}{\alpha_1} + \frac{1}{\alpha_2} . \quad (5)$$

В практических расчётах часто используют коэффициент теплопередачи как характеристику интенсивности теплообмена между потоками:

$$q = KF\Delta T . \quad (6)$$

В тех случаях, когда коэффициенты теплоотдачи учитываются по-разному, принимают усреднённую температуру стенки, разделяющей потоки. Иными словами, считают, что теплопроводность материала стенки настолько велика, что перепад температуры отсутствует.

Коэффициенты теплоотдачи зависят от многих параметров, но наиболее сильно – от скорости потока, характера набегающего жидкости на стенку, плотности и теплопроводности жидкости. При выполнении точных расчётов зависимость коэффициента теплоотдачи от параметров потока следует учитывать. Однако для большинства инженерных расчётов теплообменной аппаратуры и реакторов достаточны упрощённые представления.

Для вывода уравнений математического описания процесса теплообмена через стенку следует рассмотреть тепловой баланс каждой среды, имеющей запас тепла. Он складывается из прихода и расхода тепла, которые определяют накопление тепла в объёме; накопление является временным процессом: накопление = приход – расход. В статике ввиду равенства прихода и расхода тепла накопление равно нулю.

Накопление связано с изменением температуры:

$$\rho c_p V \Delta T$$

или для элементарного объёма

$$\rho c_p S dT dl ,$$

где  $\rho$  – плотность;  $c_p$  – удельная теплоёмкость;  $V$  – объём;  $S$  – сечение потока;  $dl$  – элементарный участок потока.

Приход и расход тепла могут определяться теплоотдачей (теплопередачей), а в случае проточной системы с распределёнными параметрами – притоком и уносом тепла с конвективным потоком.

Количество тепла, поступающее в аппарат с конвективным потоком, определяется как

$$G\rho c_p T\tau$$

или для элементарного объёма за элементарное время  $d\tau$

$$G\rho c_p T d\tau,$$

где  $G$  – объёмный расход потока.

Количество тепла, уходящее из рассматриваемого объёма с конвективным потоком, определяется следующим выражением:

$$G\rho c_p (T + \Delta T)\tau,$$

или для элементарного объёма

$$G\rho c_p (T + dT)d\tau.$$

Приход тепла, определяемый теплопередачей:

$$K \frac{F}{V} (T_{\text{вн}} - T)V\tau,$$

или для элементарного объёма за элементарное время

$$K \frac{FS}{V} (T_{\text{вн}} - T)dld\tau,$$

где  $T_{\text{вн}}$  – температура внешнего теплоносителя.

С учётом полученных соотношений накопление тепла в системе составит:

$$\rho c_p V \Delta T = G\rho c_p T\tau - G\rho c_p (T + \Delta T)\tau + KF(T_{\text{вн}} - T)\tau,$$

или в элементарном объёме за элементарное время

$$\rho c_p S dT dl = G\rho c_p T d\tau - G\rho c_p (T + dT)d\tau + K \frac{FS}{V} (T_{\text{вн}} - T)dld\tau.$$

Проведя несложные преобразования, получим уравнение теплового баланса, описывающее динамику теплообменников, во всём объёме которых происходит полное (идеальное) смешение частиц потока:

$$\rho c_p V \frac{dT}{d\tau} = G\rho c_p (T^0 - T) + KF(T_{\text{вн}} - T), \quad (7)$$

где  $T^0$ ,  $T$  – температура потока на входе и в зоне идеального смешения.

Соответственно для трубчатых теплообменников, работающих по принципу вытеснения, уравнение динамики будет выглядеть следующим образом:

$$\rho c_p \frac{\partial T}{\partial \tau} = -\frac{G\rho c_p}{S} \frac{\partial T}{\partial l} + \frac{KF(T_{\text{вн}} - T)}{V}. \quad (8)$$

Ввиду того, что в статическом режиме накопление тепла в системе равно нулю, модель статики теплообменников смешения будет иметь вид:

$$Gr_{c_p}(T^0 - T) + KF(T_{\text{вн}} - T) = 0, \quad (9)$$

статика трубчатых теплообменников описывается уравнением

$$\frac{\partial T}{\partial l} = \frac{KF(T_{\text{вн}} - T)}{u\rho c_p V}, \quad (10)$$

где  $u$  – скорость движения потока.

**Пример 1.** Теплообменник представляет собой тонкостенный змеевик, по которому в режиме идеального вытеснения движется охлаждаемый поток жидкости. Змеевик погружён в воду, непрерывно протекающую через сосуд, так что температура охлаждающей воды  $T_{\text{вн}}$  практически постоянна и равна  $10^\circ\text{C}$  во всём объёме.

Требуется определить температуру на выходе потока, идущего по змеевику со скоростью  $u = 4$  м/с, если температура его на входе равна  $95^\circ\text{C}$ , длина трубки змеевика  $L = 2$  м, его сечение  $S = 10^{-4}$  м<sup>2</sup>, коэффициент теплопередачи  $K = 1,16 \cdot 10^4$  Вт/(м<sup>2</sup> · °C). Теплоёмкость охлаждаемой жидкости  $c_p = 2,93 \cdot 10^3$  Дж/(кг · °C), её плотность  $\rho = 900$  кг/м<sup>3</sup>. Параметры считать не зависящими от температуры; изменение объёма не учитывать. Режим работы считать стационарным.

Температура охлаждаемого потока  $T$  подчиняется дифференциальному уравнению (10):

$$\frac{\partial T}{\partial l} = \frac{2K\pi r(T_{\text{вн}} - T)}{u\rho c_p S}, \quad (11)$$

где  $l$  – длина;  $r$  – радиус змеевика;  $uS$  – объёмный расход потока.

Начальное условие для уравнения (11)  $T(0) = 95^\circ\text{C}$ .

Вычислим коэффициент уравнения:

$$\beta = \frac{2K\sqrt{\pi}}{u\rho c_p \sqrt{S}} = \frac{2 \cdot 1,16 \cdot 10^4 \cdot \sqrt{3,14}}{4 \cdot 900 \cdot 2,93 \cdot 10^3 \sqrt{10^{-4}}} = 0,39.$$

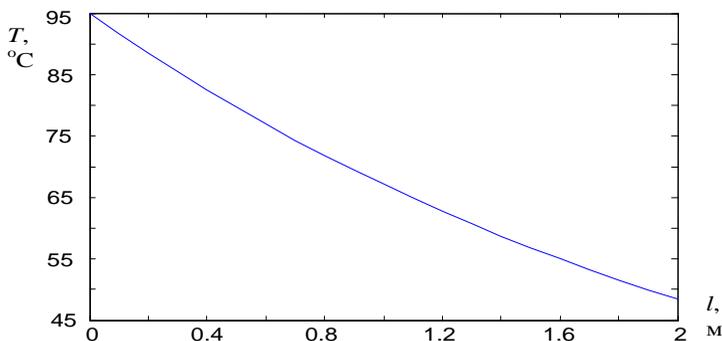
Тогда

$$\frac{\partial T}{\partial l} = 0,39(T_{\text{вн}} - T). \quad (12)$$

Уравнение (12) подлежит решению в пределах изменения независимой переменной  $l$  от 0 до 2 м.

Решение представлено на рис. 6. Температура на выходе потока составила  $T(L) = 47^\circ\text{C}$ .

Для численного интегрирования уравнения (12) в MATLAB необходимо создать два m-файла. В файле **func1\_T.m** описывается функция для вычисления правой части дифференциального уравнения (12). В файле **Теп1.m** осуществляются задание исходных данных для решения задачи и начального условия, а также решение дифференциального уравнения с использованием функции Matlab **ode45**.



**Рис. 6. Профиль температуры по длине теплообменника**

Функция **ode45** в данном случае имеет следующий синтаксис:

$$[l, T] = \text{ode45}(@\text{func1\_T}, [0 \ L], T_0),$$

где @func1\_T – дескриптор ODE-функции правой части дифференциального уравнения; [l, T] – матрица решений T, где каждая строка соответствует длине, возвращенной в векторе-столбце l; [0 L] – вектор, определяющий интервал интегрирования; T<sub>0</sub> – вектор начальных условий.

Файл **func1\_T.m**

```
function dT = func1_T(l,T) % Функция правой части
                        % дифференциального уравнения
Tvn=10;                % Температура внешнего теплоносителя
dT = 0.39*(Tvn-T);    % Уравнение правой части
                        % дифференциального уравнения
```

Файл **Тер11.m**

```
L=2;                  % Длина трубки змеевика
T0=95;                % Начальное условие диф. уравнения
[l,T] = ode45(@func1_T,[0 L],T0); % Функция решения
                        % дифференциального уравнения
plot(l,T);            % Построение зависимости T(l)
```

**Пример 2.** Жидкость охлаждается в теплообменнике типа «труба в трубе». Охлаждаемая жидкость и хладагент движутся параллельно (прямотоком). Требуется определить температуры потоков на выходе теплообменника, если начальная температура охлаждаемой жидкости равна 170 °С, а хладагента 15 °С. Плотность охлаждаемой жидкости и хладагента  $\rho = 900 \text{ кг/м}^3$ . Диаметры труб теплообменника: внутренней  $D_1 = 0,1 \text{ м}$ , наружной (для хладагента)  $D_2 = 0,3 \text{ м}$ . Длина теплообменника  $L = 1 \text{ м}$ . Теплоёмкость жидкости и хладагента  $c_p = 3,35 \cdot 10^3 \text{ Дж/(кг} \cdot \text{°C)}$ . Объёмный расход охлаждаемой жидкости  $G_1 = 2,28 \cdot 10^{-4} \text{ м}^3/\text{с}$ , хладагента  $G_2 = 5,75 \cdot 10^{-4} \text{ м}^3/\text{с}$ ,

коэффициент теплопередачи  $K = 4900 \text{ Вт}/(\text{м}^2 \cdot ^\circ\text{С})$ . Температурный профиль по длине для каждого из потоков определяется решением системы дифференциальных уравнений:

$$\frac{\partial T_1}{\partial l} = \frac{K\pi D_1(T_2 - T_1)}{G_1\rho_1 c_{p1}}, \quad (13)$$

$$\frac{\partial T_2}{\partial l} = \frac{K\pi D_1(T_1 - T_2)}{G_2\rho_2 c_{p2}}, \quad (14)$$

где  $T_1$  и  $T_2$  – температуры охлаждаемой и охлаждающей жидкости.

Начальные условия:  $T_1(0) = 170 \text{ }^\circ\text{С}$ ;  $T_2(0) = 15 \text{ }^\circ\text{С}$ .

После подстановки в уравнения (13) и (14) численных значений параметров получаем следующую систему:

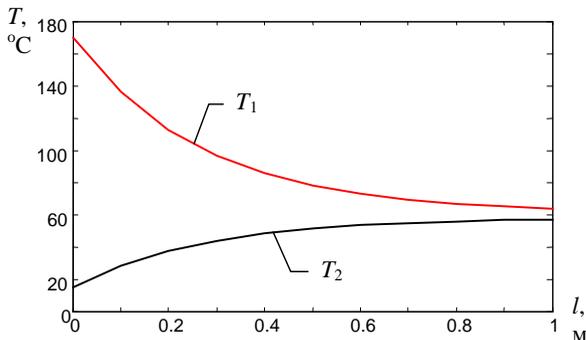
$$\frac{\partial T_1}{\partial l} = 2,239(T_2 - T_1), \quad (15)$$

$$\frac{\partial T_2}{\partial l} = 0,888(T_1 - T_2). \quad (16)$$

Графики решения системы уравнений математического описания статики теплообменника представлены на рис. 7. На нём изображены температурные профили вдоль теплообменника для обоих теплоносителей.

Можно видеть, что движущая сила процесса сильно меняется по длине, поэтому эффективность использования различных участков теплообменника не одинакова. Температуры теплоносителей на выходе теплообменника равны:  $T_1(L) = 62,9 \text{ }^\circ\text{С}$ ,  $T_2(L) = 56,5 \text{ }^\circ\text{С}$ .

Численное интегрирование уравнений (15) и (16) аналогично примеру 1. Создаются два m-файла: **func2\_T.m** – функция для вычисления правых частей дифференциальных уравнений, **Temp1.m** – задание исходных данных, начальных условий и решение дифференциальных уравнений с использованием функции MATLAB **ode45**.



**Рис. 7. Изменение температур теплоносителей по длине прямооточного теплообменника**

### Файл **func2\_T.m**

```
function dT = func2_T(l,T) % Функция правых частей
                                % дифференциальных уравнений
dT = zeros(2,1);             % Создание вектора температур
dT(1)=2.239*(T(2)-T(1));    % Уравнение правой части 1-го
                                % диф. уравнения
dT(2)=0.888*(T(1)-T(2));    % Уравнение правой части 2-го
                                % диф. уравнения
```

### Файл **Терп2.m**

```
L=1;                          % Длина теплообменника
T1_0=170;                     % Начальное условие для 1-го диф. уравнения
T2_0=15;                      % Начальное условие для 2-го диф. уравнения
[l,T] = ode45(@func2_T,[0 L],[T1_0 T2_0]); % Функция решения
                                % дифференциальных уравнений
plot(l,T(:,1),'r',l,T(:,2),'k'); % Построение зависимостей
                                % T1(l) и T2(l)
```

**Пример 3.** Смоделировать статический режим теплообменника типа «труба в трубе», используя данные, приведённые в примере 2, для случая противотока. Принять полную длину теплообменника  $L = 2,5$  м.

Тепловые процессы в противоточном теплообменнике подчиняются тем же закономерностям, что и в прямоточном. Поэтому математическое описание теплообменника записывается аналогично, однако формально однотипные уравнения для обоих теплоносителей имеют аргументы различного знака:

$$\frac{\partial T_1}{\partial l} = \frac{K\pi D_1(T_2 - T_1)}{G_1 \rho_1 c_{p1}}, \quad (17)$$

$$\frac{\partial T_2}{\partial(-l)} = \frac{K\pi D_1(T_1 - T_2)}{G_2 \rho_2 c_{p2}}. \quad (18)$$

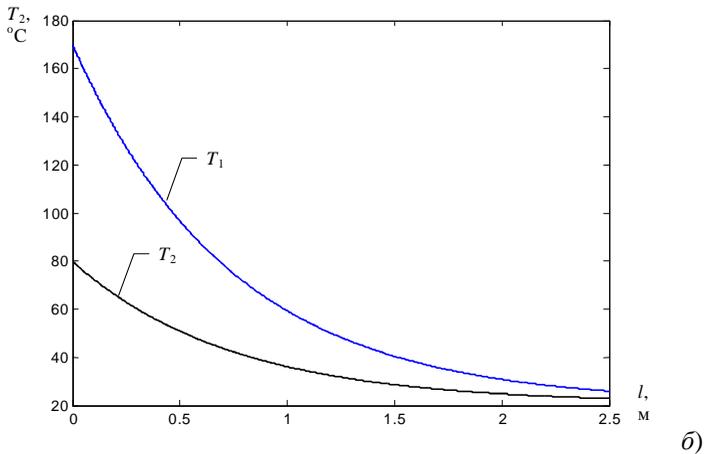
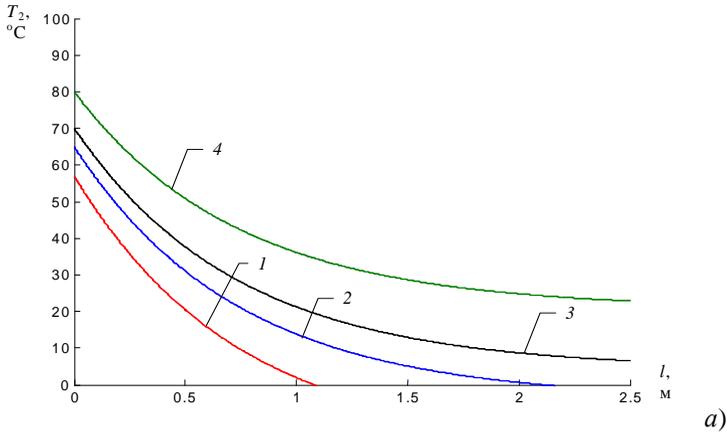
Существенным различием, отражающим иную организацию потоков теплоносителей, является принципиально другое задание условий решения уравнений (17) и (18) по сравнению с заданием при решении уравнений (13) и (14).

Совместное интегрирование уравнений (17) и (18) возможно лишь в одном направлении: либо при  $l$ , меняющемся от 0 до  $L$ , либо в обратном – от  $L$  до 0. При этом в любом случае оговорено лишь одно начальное условие, второе остаётся неизвестным. Известно лишь, к какому значению в конце решения должна подойти вторая переменная.

Для решения задачи воспользуемся последним обстоятельством и попытаемся отыскать неизвестное начальное условие  $T_2(0)$  с таким расчётом, чтобы условие, заданное для конца решения (граничное условие), было выполнено, т.е.  $T_2(L) = 15$  °С. Такие задачи при малом числе условий, подлежащих определению, обычно решают методом проб и ошибок.

Задачей поиска начального условия  $T_2(0)$  является выполнение граничного условия  $T_2(L)$  при интегрировании системы уравнений (17) и (18).

Рисунок 8, а иллюстрирует процесс поиска неизвестного начального условия  $T_2(0)$ . Кривая 1 отражает профиль температуры  $T_2$ , полученный в предположении, что хладагент нагреется до  $57^\circ\text{C}$  (значение взято произвольно). Видно, что это предположение неудачно, так как не получено ожидаемое значение  $T_2(L) = 15^\circ\text{C}$ .



**Рис. 8. Результаты моделирования на ЭВМ противоточного теплообменника:**  
 а – серия кривых решения системы уравнений (17) и (18), полученная в процессе определения недостающего начального условия;  
 б – изменение температур теплоносителей по длине теплообменника

Приняв  $T_2(0) = 65$  °С, получаем кривую 2. Она также неудовлетворительна, так как вместо  $T_2(L) = 15$  °С получаем лишь 4 °С. Кривые 3 и 4 также не соответствуют искомому начальному условию: в первом случае входная температура  $T_2(L)$  получилась заниженной, во втором – завышенной. По-видимому, искомое начальное условие лежит между двумя последними проверенными значениями (70 и 80 °С).

Тщательно исследуя намеченный диапазон задания начального условия, находим  $T_2(0) = 75$  °С.

На рисунке 8, б показан результат решения задачи. Выходная температура охлаждаемого потока  $T_1(L)$  достигает 18 °С.

Численное интегрирование уравнений (17) и (18) аналогично примеру 2.

#### Файл **func3\_T.m**

```
function dT = func3_T(l,T) % Функция правых частей
                        % дифференциальных уравнений
global b1 b2;          % Описание глобальных переменных
dT = zeros(2,1);      % Создание вектора выходных координат
dT(1)=b1*(T(2)-T(1)); % Уравнение правой части 1-го
                        % диф. уравнения
dT(2)=-b2*(T(1)-T(2)); % Уравнение правой части 2-го
                        % диф. уравнения
```

#### Файл **Тепл3.m**

```
global b1 b2;          % Описание глобальных переменных
L=2.5;                % Длина теплообменника
T1_0=170;             % Начальное условие для 1-го диф. уравнения
T2_0=80;              % Начальное условие для 2-го диф. Уравнения
ro=900;               % Плотность охлаждаемой жидкости и хладагента
D1=0.1;               % Диаметры внутренней трубы теплообменника
D2=0.3;               % Диаметры наружной трубы теплообменника
cp=3.35e3;            % Теплоемкость жидкости и хладагента
G1=2.28e-4;           % Объемный расход охлаждаемой жидкости
G2=5.75e-4;           % Объемный расход хладагента
K=4900;               % Коэффициент теплопередачи
                        % Расчет коэффициентов модели
b1=K*pi*D1/(ro*cp*G1);
b2=K*pi*D1/(ro*cp*G2);
[l,T] = ode45(@func3_T,[0 L],[T1_0 T2_0]); % Функция решения
                        % дифференциальных уравнений
plot(l,T(:,1),'b',l,T(:,2),'k');          % Построение зависимостей
                        % T1(l) и T2(l)
```

**Пример 4.** Смоделировать переходный режим теплообменника типа «смешение – смешение». Теплообменник представляет собой двухкамер-

ную ёмкость. В первую камеру ёмкости поступает охлаждаемая жидкость, во вторую – хладагент.

Для обеспечения однородного распределения температуры по объёму в камерах установлены мешалки. Плотность охлаждаемой жидкости  $850 \text{ кг/м}^3$ , а хладагента  $920 \text{ кг/м}^3$ . Объёмы камер равны и составляют  $2,5 \text{ м}^3$  каждая. Объёмный расход теплоносителя  $4,12 \cdot 10^{-3} \text{ м}^3/\text{с}$ , хладагента  $5,43 \cdot 10^{-3} \text{ м}^3/\text{с}$ .

Теплоёмкости жидкости и хладагента соответственно  $3,75 \cdot 10^3 \text{ Дж}/(\text{кг} \cdot ^\circ\text{C})$  и  $3,14 \cdot 10^3 \text{ Дж}/(\text{кг} \cdot ^\circ\text{C})$ .

Поверхность теплообмена составляет  $4 \text{ м}^2$ , а коэффициент теплопередачи равен  $K = 4360 \text{ Вт}/(\text{м}^2 \cdot ^\circ\text{C})$ . Температура охлаждаемой жидкости на входе меняется скачкообразно от  $115$  до  $200 \text{ }^\circ\text{C}$ , а температура хладагента от  $10$  до  $15 \text{ }^\circ\text{C}$ .

Изменение температуры по времени для каждого потока определяется решением системы:

$$\rho_1 c_{p1} V_1 \frac{dT_1}{dt} = G_1 \rho_1 c_{p1} (T_1^0 - T_1) + KF(T_2 - T_1), \quad (19)$$

$$\rho_2 c_{p2} V_2 \frac{dT_2}{dt} = G_2 \rho_2 c_{p2} (T_2^0 - T_2) + KF(T_1 - T_2), \quad (20)$$

где  $T_1^0$  и  $T_2^0$  – температуры жидкости и хладагента на входе в теплообменник.

Система уравнений (19) и (20) решается при начальных условиях:  $T_1(0) = T_1^*$ ;  $T_2(0) = T_2^*$ .

Для определения  $T_1^*$  и  $T_2^*$ , соответствующих номинальному статическому режиму, составляются уравнения теплового баланса стационарного режима работы теплообменника:

$$G_1 \rho_1 c_{p1} (T_1^0 - T_1) + KF(T_2 - T_1) = 0,$$

$$G_2 \rho_2 c_{p2} (T_2^0 - T_2) + KF(T_1 - T_2) = 0.$$

После подстановки численных значений получаем следующую систему:

$$13132,5 \cdot (115 - T_1) + 17440 \cdot (T_2 - T_1) = 0,$$

$$15686,2 \cdot (10 - T_1) + 17440 \cdot (T_1 - T_2) = 0,$$

которая может быть решена численно или аналитически относительно любых двух переменных, входящих в эти уравнения.

Температуры теплоносителей на выходе теплообменника в установившемся состоянии составили:  $T_1^* = 74,46 \text{ }^\circ\text{C}$ ,  $T_2^* = 43,93 \text{ }^\circ\text{C}$ .

Таким образом, получим следующую систему:

$$\frac{dT_1}{d\tau} = \frac{G_1}{V_1} (T_1^0 - T_1) + \frac{KF}{\rho_1 c_{p1} V_1} (T_2 - T_1), \quad (21)$$

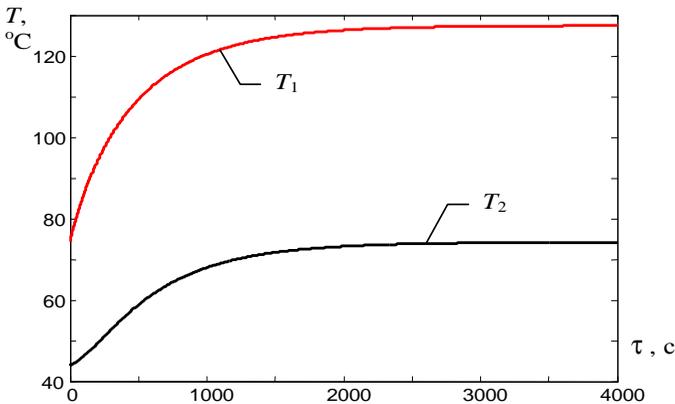
$$\frac{dT_2}{d\tau} = \frac{G_2}{V_2} (T_2^0 - T_2) + \frac{KF}{\rho_2 c_{p2} V_2} (T_1 - T_2). \quad (22)$$

Начальные условия:  $T_1(0) = 74,46 \text{ }^\circ\text{C}$ ;  $T_2(0) = 43,93 \text{ }^\circ\text{C}$ .

Графики решения системы уравнений математического описания динамики теплообменника представлены на рис. 9. На нём изображены изменения температур во времени для обоих теплоносителей на выходе теплообменника.

Температуры теплоносителей на выходе теплообменника установились через 2000 с и составили  $T_1 = 126,6 \text{ }^\circ\text{C}$ ;  $T_2 = 71,4 \text{ }^\circ\text{C}$ .

Численное интегрирование уравнений (21) и (22) аналогично предыдущим примерам 1 – 3. Отличие состоит в том, что интегрирование уравнений с помощью функции ode45 производится по времени.



**Рис. 9. Изменение температур теплоносителей во времени теплообменника типа «смешение – смешение»**

#### Файл **func4\_T.m**

```
function dT = func4_T(time,T) % Функция правых частей
% диф. уравнений
global a1 b1 a2 b2 T1_vx T2_vx; % Описание глобальных
% переменных
dT = zeros(2,1); % Создание вектора температур
dT(1)=a1*(T1_vx-T(1))+b1*(T(2)-T(1)); % Уравнение правой
% части 1-го диф. уравнения
```

$dT(2)=a2*(T2\_vx-T(2))+b2*(T(1)-T(2));$  % Уравнение правой  
% части 2-го диф. уравнения

#### Файл **Тепл4.m**

global a1 b1 a2 b2 T1\_vx T2\_vx; % Описание глобальных  
% переменных

tk=4000; % Время интегрирования  
T1\_0=74.46; % Начальное условие для 1-го диф. уравнения  
T2\_0=43.93; % Начальное условие для 2-го диф. уравнения  
T1\_vx=200; % Температура охлаждаемой жидкости на входе  
T2\_vx=10; % Температура хладагента на входе  
ro1=850; % Плотность охлаждаемой жидкости  
ro2=920; % Плотность охлаждаемого хладагента  
cp1=3.75e3; % Теплоёмкость охлаждаемой жидкости  
cp2=3.14e3; % Теплоёмкость хладагента  
G1=4.12e-3; % Объёмный расход охлаждаемой жидкости  
G2=5.43e-3; % Объёмный расход хладагента  
K=4360; % Коэффициент теплопередачи  
F=4; % Площадь поверхности теплообмена  
V=2.5; % Объём камер  
% Расчёт коэффициентов модели

a1=G1/V;

a2=G2/V;

b1=K\*F/(ro1\*cp1\*V);

b2=K\*F/(ro2\*cp2\*V);

[time,T] = ode45(@func4\_T,[0 tk],[T1\_0 T2\_0]); % Функция  
% решения дифференциальных уравнений  
plot(time,T(:,1),'b',time,T(:,2),'k'); % Построение  
% зависимостей T1(time) и T2(time)

### ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Составить математическое описание теплообменного аппарата (варианты теплообменных аппаратов и исходные данные приведены в табл. 6, 7, 8).

2. В зависимости от варианта смоделировать на ЭВМ статический и (или) динамический режимы теплообменника и определить температурные зависимости для всех теплоносителей.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Чем определяется интенсивность перехода тепла от одного теплоносителя к другому?

2. В чём отличие объектов идеального смешения и идеального вытеснения?

3. Что представляет собой уравнение теплового баланса (в статике и динамике) для объектов идеального смешения?

4. Что представляет собой уравнение теплового баланса (в статике и динамике) для объектов идеального вытеснения?

5. Какие функции MATLAB используются для численного интегрирования обыкновенных дифференциальных уравнений?

6. Какие дополнительные параметры содержит функция ode45 и другие аналогичные?

### 6. Варианты теплообменных аппаратов

№	Схема теплообменника	№	Схема теплообменника
1		3	
2		4	

### 7. Исходные данные

№	№ схемы	Режим	$V_1$ , м	$V_2$ , м	$F$ , м <sup>2</sup>	$\rho_1, \rho_2$ , кг/м <sup>3</sup>
1	1	Динамический	1,1	0,8	4,4	800
2	1	— " —	1,5	1,3	3,2	612
3	1	— " —	2,5	2,2	5,5	200
4	1	— " —	1,5	2,8	2,4	710
5	1	— " —	2,1	1,4	4,5	100
6	1	— " —	0,8	0,5	3,8	300

Продолжение табл. 7

№	№ схемы	Режим	$V_1, \text{ м}$	$V_2, \text{ м}$	$F, \text{ м}^2$	$\rho_1, \rho_2, \text{ кг/м}^3$
7	1	Динамический	1,2	0,9	6,5	520
8	2	— " —	5,2	—	2,5	800
9	2	— " —	6,5	—	4,5	612
10	2	— " —	4,6	—	3,6	200
11	2	— " —	3,2	—	2,2	710
12	2	— " —	7,2	—	5,1	100
13	2	— " —	2,5	—	2,1	300

№	$c_{p1}, c_{p2},$ кДж/(кг·°С)	$G_1, G_2,$ м <sup>3</sup> /с	$T_1^{\text{BX}}, \text{ °С}$	$T_2^{\text{BX}}, \text{ °С}$	$K, \text{ Вт/(м}^2 \cdot \text{°С)}$
1	4120	$4,5 \cdot 10^{-3}$	250	10	4360
2	5220	$3,18 \cdot 10^{-3}$	160	15	3650
3	7080	$5,23 \cdot 10^{-3}$	180	18	5650
4	4050	$3,91 \cdot 10^{-3}$	150	10	2020
5	9140	$6,88 \cdot 10^{-3}$	120	10	1110
6	6380	$4,66 \cdot 10^{-3}$	430	15	3470
7	4970	$7,88 \cdot 10^{-3}$	260	12	5720
8	4120	$4,51 \cdot 10^{-3}$	340	12	4590
9	5220	$3,18 \cdot 10^{-3}$	410	15	3210
10	7080	$5,23 \cdot 10^{-3}$	270	10	6320
11	4050	$3,91 \cdot 10^{-3}$	560	24	9230
12	9140	$6,88 \cdot 10^{-3}$	350	8	12010
13	6380	$4,66 \cdot 10^{-3}$	120	11	3260

### 8. Исходные данные

№	№ схемы	Режим	$D_1, \text{ м}$	$\rho_1, \text{ кг/м}^3$	$\rho_2, \text{ кг/м}^3$	$c_{p1},$ кДж/(кг·°С)	$c_{p2},$ кДж/(кг·°С)
14	3	Статический	0,01	900	750	4350	5140
15	3	— " —	0,02	600	450	3270	5230
16	3	— " —	0,03	200	480	1020	3120
17	3	— " —	0,015	515	200	3270	4780
18	3	— " —	0,025	700	960	3920	4770
19	3	— " —	0,035	800	300	2560	2680

№	№ схемы	Режим	$D_1$ , м	$\rho_1$ , кг/м <sup>3</sup>	$\rho_2$ , кг/м <sup>3</sup>	$c_{p1}$ , кДж/(кг·°С)	$c_{p2}$ , кДж/(кг·°С)
20	4	Статический	0,01	900	750	4350	5140
21	4	стат.	0,02	600	450	3270	5230
22	4	стат.	0,03	200	480	1020	3120
23	4	стат.	0,015	515	200	3270	4780
24	4	стат.	0,025	700	960	3920	4770
25	4	стат.	0,035	800	300	2560	2680

№	$G_1$ , м <sup>3</sup> /с	$G_2$ , м <sup>3</sup> /с	$L$ , м	$T_1(0)$ , °С	$T_2(0)$ , °С	$T_2(L)$ , °С	$K$ , Вт/(м <sup>2</sup> ·°С)
14	$2,08 \cdot 10^{-5}$	$4,12 \cdot 10^{-5}$	1	212	15	–	4817
15	$3,44 \cdot 10^{-5}$	$5,21 \cdot 10^{-5}$	1,5	193	25	–	2022
16	$2,92 \cdot 10^{-4}$	$2,45 \cdot 10^{-4}$	2	620	15	–	1460С
17	$5,22 \cdot 10^{-4}$	$2,74 \cdot 10^{-3}$	3,2	500	10	–	34600
18	$5,48 \cdot 10^{-4}$	$6,18 \cdot 10^{-4}$	1,8	450	18	–	41600
19	$3,12 \cdot 10^{-5}$	$4,05 \cdot 10^{-5}$	2,5	670	25	–	10350
20	$2,08 \cdot 10^{-5}$	$4,12 \cdot 10^{-5}$	1	212	–	15	4817
21	$3,44 \cdot 10^{-5}$	$5,21 \cdot 10^{-5}$	1,5	193	–	25	2022
22	$2,92 \cdot 10^{-4}$	$2,45 \cdot 10^{-4}$	2	620	–	15	14600
23	$5,22 \cdot 10^{-4}$	$2,74 \cdot 10^{-3}$	3,2	500	–	10	34600
24	$5,48 \cdot 10^{-4}$	$6,18 \cdot 10^{-4}$	1,8	450	–	18	41600
25	$3,12 \cdot 10^{-5}$	$4,05 \cdot 10^{-5}$	2,5	670	–	25	10350

Лабораторная работа 5

**ПОСТРОЕНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ  
АНАЛИТИЧЕСКИМ МЕТОДОМ  
(МОДЕЛИРОВАНИЕ ХИМИЧЕСКИХ РЕАКЦИЙ)**

*Цель работы:* освоить методику математического моделирования химических реакций на ЭВМ.

*Задания работы:*

1. Ознакомиться с основными принципами моделирования химических реакций.

2. Выполнить вариант задания.

*Содержание отчёта:*

1. Вариант задания.
2. Математическая модель химической реакции.
3. Все расчётные формулы, используемые в работе.
4. Графики рассчитанных зависимостей для компонентов реакции.
5. Тексты программ.

## КРАТКИЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Скорость химической реакции есть изменение (уменьшение или увеличение) числа молей реагентов в результате химического взаимодействия в единицу времени на единицу объёма (для гомогенных реакций) или на единицу поверхности для гетерогенных процессов.

В соответствии с этим определением скорость гомогенных химических реакции

$$v_r = \frac{1}{V} \frac{dN}{d\tau}, \quad (23)$$

где  $V$  – объём реактора;  $dN/d\tau$  – изменение числа молей реагентов в единицу времени.

Учитывая, что число молей  $N = CV$ , где  $C$  – концентрация реагента, в общем случае получим:

$$v_r = \frac{1}{V} \frac{d(CV)}{d\tau} \quad (24)$$

или

$$v_r = \frac{1}{V} \frac{VdC + CdV}{d\tau}. \quad (25)$$

Для реакций, идущих при постоянном объёме в замкнутой системе, второе слагаемое уравнения (25) равно нулю и уравнение приводится к виду:

$$v_r = \pm \frac{dC}{d\tau}. \quad (26)$$

Знак (+) указывает, что в реакции накапливается вещество, знак (–) – что концентрация вещества снижается.

Для гетерогенных химических реакций скорость процесса можно представить следующим соотношением:

$$v_r = \frac{1}{S} \frac{dN}{d\tau}. \quad (27)$$

В дальнейшем будем полагать, что в системе осуществляется гомогенный химический процесс.

Скорость химической реакции является функцией концентраций реагирующих веществ, а также температуры и иногда давления в системе:

$$v_r = f(\bar{C}, P, T), \quad (28)$$

где  $\bar{C}$  – вектор концентраций реагирующих веществ;  $P$  и  $T$  – соответственно давление и температура.

Химические реакции можно условно разделить на простые и сложные. Химическая кинетика определяет простые реакции как такие, которые по существу проходят в одну стадию; простая реакция содержит один акт химического превращения. Скорость простой реакции зависит от концентраций только исходных веществ. В случае сложных реакций их скорость зависит от концентраций исходных, а также промежуточных (образующихся) веществ, а механизм учитывает несколько путей химического превращения (обратимые, последовательные, параллельные, разветвлённые, последовательно-параллельные реакции).

Уравнения кинетики простых и сложных химических реакций имеют некоторые различия, поэтому будем рассматривать их отдельно.

### Кинетика простых реакций

Если в простой реакции участвуют  $n$  веществ, то материальный баланс её выражается стехиометрическим уравнением:

$$\alpha_1 A_1 + \alpha_2 A_2 + \dots + \alpha_n A_n = 0, \quad (29)$$

где  $\alpha_i$  – стехиометрические коэффициенты (отрицательные для исходных веществ и положительные для образующихся веществ);  $A_i$  – участвующие в реакции вещества.

Коэффициент  $\alpha_i$  показывает, сколько молекул вещества  $A_i$  участвует в реакции.

Определим скорость простой химической реакции как скорость образования любого вещества, участвующего в реакции, делённую на стехиометрический коэффициент указанного компонента. При этом для исходных реагирующих компонентов стехиометрические коэффициенты принимаются со знаком минус, а для продуктов химического превращения – со знаком плюс. Так, для химической реакции (29) скорость определяется как

$$v_r = \frac{v_{rA_1}}{\alpha_1} = \frac{v_{rA_2}}{\alpha_2} = \dots = \frac{v_{rA_n}}{\alpha_n}. \quad (30)$$

Во многих формально-кинетических расчётах используются значения скорости реакции, выраженные через конкретные вещества. Для этих значений из формулы (30) получаем:

$$v_{ri} = \alpha_i v_r. \quad (31)$$

При использовании формулы (31) следует учитывать, что  $\alpha_i < 0$  для исходных веществ и  $\alpha_i > 0$  для продуктов реакции.

Скорость простой реакции, а также стадий сложных реакций, согласно закону действующих масс, пропорциональна наличной концентрации реагирующих веществ в некоторых степенях. Так, для реакции



соответствующая зависимость будет выражаться кинетическим уравнением:

$$v_r = k C_A^{n_1} C_B^{n_2}. \quad (33)$$

Здесь  $n_1$  – порядок реакции по веществу  $A$ ,  $n_2$  – порядок по веществу  $B$ , сумму  $n_1 + n_2$  называют общим, или суммарным порядком. Коэффициент пропорциональности  $k$  именуют константой скорости реакции.

Согласно уравнению Аррениуса константа скорости химической реакции зависит от температуры:

$$k = k_0 e^{-\frac{E}{RT}}, \quad (34)$$

где  $E$  – энергия активации;  $R$  – универсальная газовая постоянная;  $k_0$  – предэкспоненциальный множитель.

Порядок реакции или стадии будем обозначать цифрой над (или под) стрелкой, указывающей направление реакции; здесь же будем приводить обозначение константы скорости. Так, запись

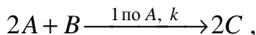


обозначает реакцию, имеющую 1-й порядок по веществу  $A$  и  $B$  и константу скорости  $k$ .

На практике встречаются реакции самых разнообразных порядков: целочисленных и дробных, нулевого, изредка и отрицательных.

**Пример 5.** Скорость реакции по веществам.

Рассмотрим реакцию



имеющую первый порядок по реагенту  $A$  и константу скорости  $k$ . Для этой реакции, в соответствии с формулой (31), имеем

$$v_{rA} = -kC_A, \quad v_{rB} = -kC_A, \quad v_{rC} = 2kC_A.$$

## Кинетика сложных реакций

Сложная реакция состоит из ряда стадий, каждая из них имеет свою скорость, и нельзя достаточно точно определить скорость всей сложной реакции в целом.

В то же время любое из участвующих в реакции веществ образуется или расходуется с определённой скоростью, причём знание этой скорости обязательно при расчёте процесса.

Скорость сложной реакции по любому веществу  $i$  равна алгебраической сумме скоростей всех стадий по этому веществу (с учётом стехиометрических коэффициентов):

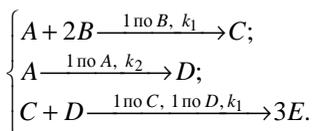
$$v_{ri} = \sum_{j=1}^m v_{rj1} = \sum_{j=1}^m \alpha_{ij} v_{rj1} \quad (36)$$

Здесь  $j$  – номер стадии;  $m$  – число стадий.

Скорость стадии по веществу определяется формулой (31). Если вещество в стадии не участвует, его стехиометрический коэффициент в этой стадии равен нулю.

**Пример 6.** Скорости сложной реакции по веществам.

Запишем кинетические уравнения для веществ, участвующих в реакции:



Вещество  $A$  участвует в 1-й и 2-й стадиях со стехиометрическими коэффициентами  $-1$  и  $-1$ . Поэтому

$$v_{rA} = -v_{r1} - v_{r2} = -k_1 C_B - k_2 C_A.$$

Соответственно:

$$\begin{aligned} v_{rB} &= -2k_1 C_B, \quad v_{rC} = k_1 C_B - k_3 C_C C_D, \\ v_{rD} &= k_2 C_A - k_3 C_C C_D, \quad v_{rE} = 3k_3 C_C C_D. \end{aligned}$$

### Выбор ключевых компонентов реакции

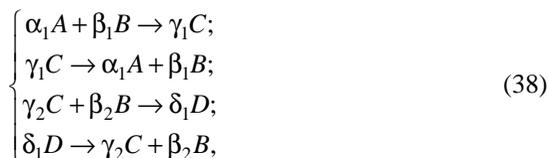
Если химическая реакция (простая или сложная) протекает в замкнутой системе при постоянном объёме, то ход её описывается системой дифференциальных уравнений. Действительно, совместное рассмотрение уравнений (23), (33) и (36) приводит к формуле:

$$\frac{dC_i}{dt} = \sum_{j=1}^m \alpha_{ij} v_{rj1}, \quad (37)$$

причём число таких уравнений равно числу веществ, участвующих в реакции (реагентов и продуктов). При этом для сложных химических реакций число уравнений может стать значительным, что существенно затруднит возможность использования математической модели при исследовании такого процесса.

Число уравнений математической модели, описывающих изменение количеств компонентов химической реакции с течением процесса, можно сократить до некоторого минимума путём написания указанных уравнений только для ключевых компонентов реакции. Поведение остальных компонентов реакции, не вошедших в число ключевых, при этом описывается простыми стехиометрическими соотношениями, представленными через количества ключевых компонентов.

Для иллюстрации этих правил воспользуемся конкретным примером следующей химической реакции:



где  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  – стехиометрические коэффициенты.

Кинетика процесса полностью описывается, если заданы скорости  $v_{r1}$ ,  $v_{r2}$ ,  $v_{r3}$  и  $v_{r4}$  всех стадий. В этом случае скорость образования любого компонента, участвующего в сложной реакции, можно записать через скорости стадий, в которых участвует данный компонент, и соответствующие стехиометрические коэффициенты с учётом правила знаков:

$$\begin{aligned} v_{rA} &= -\alpha_1 v_{r1} + \alpha_1 v_{r2}, \\ v_{rB} &= -\beta_1 v_{r1} + \beta_1 v_{r2} - \beta_2 v_{r3} + \beta_2 v_{r4}, \\ v_{rC} &= \gamma_1 v_{r1} - \gamma_1 v_{r2} - \gamma_2 v_{r3} + \gamma_2 v_{r4}, \\ v_{rD} &= -\delta_1 v_{r3} + \delta_1 v_{r4}. \end{aligned} \quad (39)$$

Составим матрицу стехиометрических коэффициентов для системы уравнений (5.17):

$$\begin{pmatrix} -\alpha_1 & \alpha_1 & 0 & 0 \\ -\beta_1 & \beta_1 & -\beta_2 & \beta_2 \\ \gamma_1 & -\gamma_1 & -\gamma_2 & \gamma_2 \\ 0 & 0 & \delta_1 & -\delta_1 \end{pmatrix}, \quad (40)$$

где коэффициент, стоящий на пересечении  $i$ -й строки и  $k$ -го столбца, представляет собой стехиометрический коэффициент  $i$ -го компонента в  $k$ -й стадии реакции, взятый с учётом знака.

Задача выбора ключевых компонентов сводится к нахождению ранга матрицы стехиометрических коэффициентов (40). Под рангом матрицы понимается порядок наименьшего определителя, который можно построить из матрицы. Ранг матрицы и, следовательно, число ключевых компонентов можно определить известными методами матричного исчисления.

Для случая матрицы (40) можно поступить следующим образом. Будем искать ранг матрицы, последовательно увеличивая порядок определителей. Очевидно, что определитель первого порядка, т.е. содержащий только один элемент, можно построить легко, для чего необходимо взять любой элемент матрицы, отличный от нуля.

Теперь переходим к отысканию определителя второго порядка, отличного от нуля. Берём первый определитель, стоящий в левом верхнем углу матрицы (40):

$$\begin{vmatrix} -\alpha_1 & \alpha_1 \\ -\beta_1 & \beta_1 \end{vmatrix} = -\alpha_1\beta_1 + \alpha_1\beta_1 = 0. \quad (41)$$

Он оказывается равным нулю. Нетрудно проверить, что и любой определитель, составленный из строк первых двух столбцов матрицы, также равен нулю, что в данном случае указывает на линейную зависимость между первым и вторым столбцами матрицы (40). Действительно, второй столбец может быть получен из первого простым умножением на  $-1$ .

Аналогично обстоит дело с определителями второго порядка, построенными из строк третьего и четвёртого столбцов.

Остаётся рассмотреть определители второго порядка, которые могут быть построены из строк второго и третьего столбцов. Легко убедиться, что для этих столбцов любая комбинация двух строк даёт определители, отличные от нуля, например определители:

$$\begin{vmatrix} \alpha_1 & 0 \\ \beta_1 & \beta_2 \end{vmatrix} = \alpha_1\beta_2 \neq 0, \quad \begin{vmatrix} \alpha_1 & 0 \\ 0 & \delta_1 \end{vmatrix} = \alpha_1\delta_1 \neq 0. \quad (42)$$

Итак, получено, что ранг матрицы (40) равен 2 и, следовательно, в качестве ключевых компонента можно использовать два компонента.

Пусть в качестве ключевых компонентов выбраны  $A$  и  $D$ ; тогда, как нетрудно проверить, скорости образования компонентов  $B$  и  $C$  могут быть выражены через скорости образования  $v_{rA}$  и  $v_{rD}$ :

$$v_{rB} = \frac{\beta_1}{\alpha_1} v_{rA} - \frac{\beta_2}{\delta_1} v_{rD}, \quad v_{rC} = \frac{\gamma_1}{\alpha_1} v_{rA} - \frac{\gamma_2}{\delta_1} v_{rD}. \quad (43)$$

Если же в качестве ключевых компонентов выбрать  $A$  и  $B$ , то для  $v_{rC}$  и  $v_{rD}$  соответственно получим:

$$v_{rC} = -\frac{\beta_1\gamma_2 + \beta_2\gamma_1}{\alpha_1\beta_2}v_{rA} + \frac{\gamma_2}{\beta_2}v_{rB}, \quad v_{rC} = \frac{\beta_1\delta_1}{\alpha_1\beta_2}v_{rA} - \frac{\delta_1}{\beta_2}v_{rB}. \quad (44)$$

Аналогично можно найти соотношения и для любой другой выбранной пары ключевых компонентов:  $A$  и  $C$ ;  $B$  и  $C$ ;  $B$  и  $D$ ;  $C$  и  $D$ .

Уравнения, описывающие изменение концентраций компонентов со временем, имеют в данном случае вид:

$$\begin{cases} \frac{dC_A}{d\tau} = v_{rA}, & \frac{dC_B}{d\tau} = v_{rB}, \\ \frac{dC_C}{d\tau} = v_{rC}, & \frac{dC_D}{d\tau} = v_{rD}. \end{cases} \quad (45)$$

С учётом соотношений (43) второе и третье уравнения системы (45) могут быть заменены выражениями:

$$\frac{dC_B}{d\tau} = \frac{\beta_1}{\alpha_1} \frac{dC_A}{d\tau} - \frac{\beta_2}{\delta_1} \frac{dC_D}{d\tau}, \quad \frac{dC_C}{d\tau} = -\frac{\gamma_1}{\alpha_1} \frac{dC_A}{d\tau} - \frac{\gamma_2}{\delta_1} \frac{dC_D}{d\tau}, \quad (46)$$

после интегрирования которых получим:

$$C_B - C_{B0} = \frac{\beta_1}{\alpha_1}(C_A - C_{A0}) - \frac{\beta_2}{\delta_1}(C_D - C_{D0}),$$

$$C_C - C_{C0} = -\frac{\gamma_1}{\alpha_1}(C_A - C_{A0}) - \frac{\gamma_2}{\delta_1}(C_D - C_{D0}),$$

где  $C_{A0}$ ,  $C_{B0}$ ,  $C_{C0}$  и  $C_{D0}$  – начальные концентрации компонентов.

Отсюда окончательно находим соотношения:

$$C_B = C_{B0} + \frac{\beta_1}{\alpha_1}(C_A - C_{A0}) - \frac{\beta_2}{\delta_1}(C_D - C_{D0}),$$

$$C_C = C_{C0} - \frac{\gamma_1}{\alpha_1}(C_A - C_{A0}) - \frac{\gamma_2}{\delta_1}(C_D - C_{D0}),$$

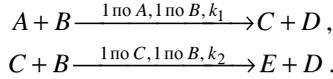
которые могут использоваться в системе уравнений математического описания химической реакции вместо второго и третьего дифференциального уравнения системы (45).

**Пример 7.** Моделирование химической реакции омыления этилсукцената едким натром.

Для упрощения записи введём обозначения для веществ, участвующих в реакции:



Тогда схему реакции можно записать так:



Реакция протекает в жидкой фазе, т.е. практически изохорно. При  $0^\circ\text{C}$   $k_1 = 2,24 \text{ м}^3/(\text{кмоль}\cdot\text{с})$ ;  $k_2 = 0,33 \text{ м}^3/(\text{кмоль}\cdot\text{с})$ .

Вследствие изохорности реакции можно применить уравнение (26):

$$v_{ri} = \frac{dC_i}{d\tau}.$$

По схеме скорости реакции равны

$$\begin{cases} v_{rA} = -k_1 C_A C_B, \\ v_{rB} = -k_1 C_A C_B - k_2 C_C C_B, \\ v_{rC} = k_1 C_A C_B - k_2 C_C C_B, \\ v_{rD} = k_1 C_A C_B + k_2 C_C C_B, \\ v_{rE} = k_2 C_C C_B. \end{cases}$$

Нетрудно убедиться, что ранг матрицы стехиометрических коэффициентов равен 2, вследствие чего выберем в качестве ключевых любые два компонента (например, компоненты  $A$  и  $E$ ). Тогда скорости остальных компонентов можно выразить следующими соотношениями:

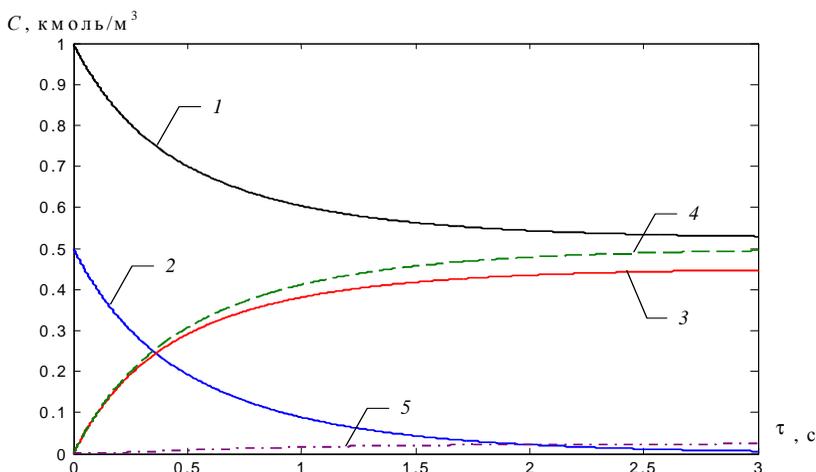
$$\begin{aligned} v_{rB} &= v_{rA} - v_{rE}, \\ v_{rC} &= -v_{rA} - v_{rE}, \\ v_{rD} &= -v_{rA} + v_{rE}. \end{aligned}$$

С учётом последних соотношений математическое описание реакции будет выглядеть следующим образом:

$$\begin{cases} \frac{dC_A}{d\tau} = -k_1 C_A C_B, \\ \frac{dC_E}{d\tau} = k_2 C_C C_B, \\ C_B = C_{B0} + (C_A - C_{A0}) - (C_E - C_{E0}), \\ C_C = C_{C0} - (C_A - C_{A0}) - (C_E - C_{E0}), \\ C_D = C_{D0} - (C_A - C_{A0}) + (C_E - C_{E0}). \end{cases} \quad (47)$$

На рисунке 10 приведены графические результаты моделирования – полученные кинетические кривые для начальных условий:  $C_{A0} = 1 \text{ кмоль/м}^3$ ;  $C_{B0} = 0,5 \text{ кмоль/м}^3$ .

Необходимо отметить, что без применения вычислительной техники расчёт очень громоздок; использование ЭВМ позволяет весьма просто и быстро рассчитать кинетические кривые для любых начальных концентраций и констант скоростей.



**Рис. 10. Графические результаты моделирования реакции омыления**

Программа на Matlab, решающая систему уравнений (5.25) представлена ниже.

```

dt=0.001;           % Шаг интегрирования по времени
time=[0:dt:3];     % Массив времени
n=length(time);    % Размерность массива времени
% Создание массивов концентраций
Ca=zeros(1,n);
Cb=zeros(1,n);
Cc=zeros(1,n);
Cd=zeros(1,n);
Ce=zeros(1,n);
% Начальные условия
Ca(1)=1;
Cb(1)=0.5;
Cc(1)=0;
Cd(1)=0;
Ce(1)=0;
% Константы скоростей реакции
k1=2.24; k2=0.33;
% Интегрирование дифференциальных уравнений методом Эйлера
for i=2:n
    Ca(i)=Ca(i-1)+dt*(-k1*Ca(i-1)*Cb(i-1));
    Ce(i)=Ce(i-1)+dt*(k2*Cc(i-1)*Cb(i-1));
    Cb(i)=Cb(1)+(Ca(i-1)-Ca(1))-(Ce(i-1)-Ce(1));
    Cc(i)=Cc(1)-(Ca(i-1)-Ca(1))-(Ce(i-1)-Ce(1));

```

Cd(i)=Cd(1)-(Ca(i-1)-Ca(1))+(Ce(i-1)-Ce(1));

end

% Построение кинетических кривых

plot(time,Ca,'k', time,Cb,'b', time,Cc,'r', time,Cd,'g', time,Ce,'m');

### ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Выразить уравнения скоростей реакции по каждому веществу, участвующему в процессе (варианты контрольных заданий и исходные данные приведены в табл. 9, 10).

2. Провести выбор ключевых компонентов реакции.

3. Составить математическое описание химической реакции с учётом пункта 2.

4. Определить кинетические зависимости для веществ, участвующих в процессе.

### 9. Варианты контрольных заданий

№	Описание реакции	№	Описание реакции
1	$A + B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} C + D$ $C + D \xrightarrow{1 \text{ по } C, 1 \text{ по } D, k_2} A + B$ $A + C \xrightarrow{1 \text{ по } A, 1 \text{ по } C, k_3} E$	7	$A + 2B \xrightarrow{2 \text{ по } B, k_1} 2C + 2D$ $2C + 2D \xrightarrow{1 \text{ по } C, 1 \text{ по } D, k_2} A + 2B$ $2B + C \xrightarrow{2 \text{ по } B, k_3} E$
2	$2A + B \xrightarrow{2 \text{ по } A, k_1} 2C + D$ $A + 2C \xrightarrow{1 \text{ по } A, 1 \text{ по } C, k_2} 2D + E$ $B \xrightarrow{1 \text{ по } B, k_3} E$	8	$A + B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} C + D$ $D + A \xrightarrow{1 \text{ по } D, 1 \text{ по } A, k_2} C + E$ $D \xrightarrow{1 \text{ по } D, k_3} A + B$
3	$2A + B \xrightarrow{2 \text{ по } A, k_1} 2C + D$ $2C + D \xrightarrow{2 \text{ по } C, k_2} 2A + B$ $A + 2C \xrightarrow{2 \text{ по } C, k_3} 2E$	9	$A + 2B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} 2C + 2D$ $2C + 2D \xrightarrow{1 \text{ по } C, 1 \text{ по } D, k_2} A + 2B$ $2B + C \xrightarrow{2 \text{ по } B, k_3} E$
4	$A + 3B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} 2C + D$ $2B \xrightarrow{2 \text{ по } B, k_2} D + 2E$ $B + D \xrightarrow{1 \text{ по } B, 1 \text{ по } D, k_3} 2E$	10	$2A + 2B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} 2C + D$ $2C + B \xrightarrow{2 \text{ по } C, k_2} E$ $E \xrightarrow{1 \text{ по } E, k_3} 2C + B$
5	$A + 2B \xrightarrow{2 \text{ по } B, k_1} C + 2D$ $C + 2D \xrightarrow{2 \text{ по } D, k_2} A + 2B$ $2A + C \xrightarrow{2 \text{ по } A, k_3} 2E$	11	$2A + B \xrightarrow{2 \text{ по } A, k_1} C + 2D$ $2A + D \xrightarrow{2 \text{ по } A, k_2} E$ $E \xrightarrow{2 \text{ по } E, k_3} 2A + D$
6	$3A + B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} 2C + D$ $2A \xrightarrow{2 \text{ по } A, k_2} D + 2E$ $A + D \xrightarrow{1 \text{ по } A, 1 \text{ по } D, k_3} E$	12	$A + B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} C + D$ $C + D \xrightarrow{2 \text{ по } C, k_2} A + B$ $A + C \xrightarrow{1 \text{ по } A, 1 \text{ по } C, k_3} E$

№	Описание реакции	№	Описание реакции
13	$3A + B \xrightarrow{2 \text{ по } A, k_1} C + 2D$ $3A + D \xrightarrow{2 \text{ по } A, k_2} E$ $E \xrightarrow{2 \text{ по } E, k_3} 3A + D$	20	$2A + B \xrightarrow{2 \text{ по } A, k_1} C + D$ $C + 2B \xrightarrow{2 \text{ по } B, k_2} D + E$ $C \xrightarrow{2 \text{ по } C, k_3} 3E$
14	$2A + B \xrightarrow{2 \text{ по } A, k_1} 2C + D$ $2C + D \xrightarrow{2 \text{ по } C, k_2} 2A + B$ $A + 2C \xrightarrow{1 \text{ по } A, 1 \text{ по } C, k_3} 2E$	21	$A + 2B \xrightarrow{2 \text{ по } B, k_1} C + 2D$ $2C + 2B \xrightarrow{2 \text{ по } B, k_2} 2D + E$ $C \xrightarrow{2 \text{ по } C, k_3} 2E$
15	$A + B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} C + D$ $C + 2B \xrightarrow{2 \text{ по } B, k_2} 2E$ $2E \xrightarrow{2 \text{ по } E, k_3} C + 2B$	22	$A + 2B \xrightarrow{2 \text{ по } B, k_1} C + 2D$ $2C + B \xrightarrow{2 \text{ по } C, k_2} 2D + E$ $C \xrightarrow{2 \text{ по } C, k_3} 2E$
16	$2A + B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} C + D$ $C + 2D \xrightarrow{2 \text{ по } D, k_2} A + 2B$ $2A + C \xrightarrow{2 \text{ по } A, k_3} 2E$	23	$2A + 2B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} C + 2D$ $A + 2C \xrightarrow{2 \text{ по } C, k_2} 2D + E$ $B \xrightarrow{1 \text{ по } B, k_3} E$
17	$2A + B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} C + D$ $C + 2B \xrightarrow{2 \text{ по } B, k_2} 2E$ $2E \xrightarrow{2 \text{ по } E, k_3} C + 2B$	24	$2A + 2B \xrightarrow{1 \text{ по } A, 1 \text{ по } B, k_1} 2C + D$ $A + 2C \xrightarrow{2 \text{ по } C, k_2} 2D + E$ $B \xrightarrow{1 \text{ по } B, k_3} E$
18	$A + 2B \xrightarrow{2 \text{ по } B, k_1} 2C + 2D$ $C + 2D \xrightarrow{1 \text{ по } C, 2 \text{ по } D, k_2} A + 2B$ $2B + C \xrightarrow{2 \text{ по } B, k_3} 2E$	25	$2A + B \xrightarrow{2 \text{ по } A, k_1} 2C + 2D$ $B + 2D \xrightarrow{2 \text{ по } D, k_2} C + 2E$ $D \xrightarrow{1 \text{ по } D, k_3} E$
19	$2A + B \xrightarrow{2 \text{ по } A, k_1} C + D$ $C + 2B \xrightarrow{2 \text{ по } B, k_2} D + E$ $2C \xrightarrow{2 \text{ по } C, k_3} 3E$	26	$2A + B \xrightarrow{2 \text{ по } A, k_1} 2C + D$ $B + 2D \xrightarrow{2 \text{ по } D, k_2} C + 2E$ $D \xrightarrow{1 \text{ по } D, k_3} E$

## 10. Исходные данные

№	$C_{A0}$ , кмоль/м <sup>3</sup>	$C_{B0}$ , моль/м <sup>3</sup>	$k_1$	$k_2$	$k_3$
1	2	3	4	5	6
1	2,0	3,0	1,12	0,85	1,74
2	5,0	2,5	1,12	0,85	1,74
3	3,5	2,5	2,65	0,11	0,43
4	3,2	1,6	1,16	1,05	0,72
5	2,4	0,8	1,26	1,00	1,07
6	1,6	2,1	2,25	1,12	0,87

№	$C_{A0}$ , кмоль/м <sup>3</sup>	$C_{B0}$ , моль/м <sup>3</sup>	$k_1$	$k_2$	$k_3$
7	0,8	1,4	0,25	0,12	0,87
8	2,0	1,0	1,28	0,25	0,82
9	0,8	0,5	2,14	1,18	0,54
10	3,8	2,5	2,27	0,54	1,47
11	1,6	0,9	2,47	0,87	2,45
12	2,5	1,5	0,23	2,94	1,16
13	1,8	2,0	0,85	1,74	1,15
14	2,0	1,5	0,75	1,65	1,31
15	1,6	1,4	1,25	1,51	2,08
16	2,4	1,2	2,62	0,97	1,28
17	2,5	1,6	2,41	2,50	1,66
18	4,0	2,0	0,37	3,21	1,44
19	2,1	2,9	1,15	0,75	1,64
20	4,5	2,5	1,22	0,68	1,51
21	3,4	2,6	2,59	0,16	0,37
22	3,0	2,1	1,09	1,12	0,62
23	3,5	2,6	2,22	0,49	1,52
24	1,5	1,0	2,54	0,82	2,55
25	2,6	1,5	0,21	3,04	1,21
26	2,0	2,0	0,91	1,67	1,21

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое скорость химической реакции?
2. Кинетика простых реакций.
3. Кинетика сложных реакций.
4. Каким образом можно сократить число уравнений математической модели, описывающих изменение количеств компонентов химической реакции с течением процесса?
5. Как определяется порядок химической реакции?
6. Объясните физический смысл уравнения Аррениуса.

Лабораторная работа 6

### МОДЕЛИРОВАНИЕ ПРОСТЫХ ГИДРАВЛИЧЕСКИХ СИСТЕМ

*Цель работы:* освоить методику математического моделирования гидравлических систем на ЭВМ.

### Задания работы:

1. Ознакомиться с основными принципами моделирования гидравлических систем.

2. Выполнить вариант задания.

### Содержание отчёта:

1. Вариант задания.

2. Математическая модель гидравлической системы.

3. Все расчётные формулы, используемые в работе.

4. Тексты программ и результаты вычислений.

## КРАТКИЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

К простым гидравлическим системам (рис. 11) относятся технологические схемы трубопроводов, для которых принимаются следующие допущения:

– во всех трубах протекает однофазный поток жидкости, температура которого одинакова на всех участках;

– все трубы располагаются на одном уровне, в системе нет обратных потоков (рециклов), не учитываются местные сопротивления и перепады давлений в трубах, т.е. рассматриваются так называемые короткие трубопроводы;

– системы включают только клапаны (вентили) с постоянными неизменяющимися коэффициентами пропускной способности и закрытые ёмкости (аккумуляторы), давление газа в которых подчиняется идеальным законам.

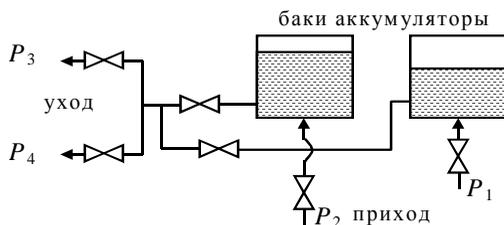


Рис. 11. Схематическое изображение гидравлической системы с двумя закрытыми емкостями (аккумуляторами)

**Пример 8.** Необходимо смоделировать гидравлическую систему, представленную на рис. 12.

Давления на входах и выходах системы считаются заданными. Также заданы значения коэффициентов пропускной способности. Определить расходы потоков в каждой ветви системы, а также давление в точках разветвления ( $P_{11}$ ,  $P_{12}$ ,  $P_{13}$ ).

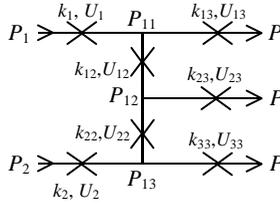


Рис. 12

Расход через сужающее устройство определяется уравнением:

$$U = k\sqrt{P_1 - P_2} ,$$

где  $P_1$  – давление до сужающего устройства;  $P_2$  – давление после сужающего устройства;  $k$  – коэффициент пропускной способности сужающего устройства.

$$U_1 = k_1\sqrt{P_1 - P_{11}} , \quad (48)$$

$$U_2 = k_2\sqrt{P_2 - P_{13}} , \quad (49)$$

$$U_{12} = k_{12}\sqrt{P_{11} - P_{12}} , \quad (50)$$

$$U_{22} = k_{22}\sqrt{P_{13} - P_{12}} , \quad (51)$$

$$U_{13} = k_{13}\sqrt{P_{11} - P} , \quad (52)$$

$$U_{23} = k_{23}\sqrt{P_{12} - P} , \quad (53)$$

$$U_{33} = k_{33}\sqrt{P_{13} - P} . \quad (54)$$

Необходимо добавить 3 уравнения, чтобы система уравнений имела решение (10 переменных – 10 уравнений).

$$U_1 - U_{12} - U_{13} = 0 , \quad (55)$$

$$U_{12} + U_{22} - U_{23} = 0 , \quad (56)$$

$$U_2 - U_{22} - U_{33} = 0 . \quad (57)$$

Для выбора алгоритма математической декомпозиции, который позволит определить искомые переменные, необходимо построить и проанализировать информационную матрицу системы уравнений математического описания (48) – (57).

Информационная матрица системы уравнений математического описания представляет собой квадратную матрицу, строки которой соответствуют номерам уравнений, а столбцы – обозначению определяемых переменных.

Составим информационную матрицу (табл. 11).

Каждая строка матрицы соответствует определённому уравнению. Каждой неизвестной переменной соответствует определённый столбец. Наличие переменной в том или ином уравнении обозначается символом на пересечении соответствующей строки с соответствующим столбцом.

Данные символы обозначают следующее:

[+] – начальное приближение;

<+> – в данном уравнении переменная определяется при известных значениях остальных переменных;

(+) – переменная косвенно определена другим уравнением.

### 11. Информационная матрица системы уравнений, описывающей стационарный режим гидравлической системы

Номер уравнения	$U_1$	$U_2$	$U_{12}$	$U_{22}$	$U_{13}$	$U_{23}$	$U_{33}$	$P_{11}$	$P_{12}$	$P_{13}$	Порядок решения
1	<+>							[+]			1
2		<+>								(+)	8
3			(+)					(+)	<+>		4
4				(+)					(+)	<+>	7
5					<+>			(+)			2
6						<+>			(+)		5
7							<+>			(+)	9
8	(+)		<+>	(+)							3
9			(+)	<+>		(+)					6
10		<+>		(+)			(+)				10

Если получаемое  $[U_2$  (в решении 8) –  $U_2$  (в решении 10)]  $> \epsilon$ , то методом половинного деления (или каким-либо другим числовым методом) выбрать новое начальное приближение (для данного примера  $P_{11}$ ).

Если  $U_2(8) = U_2(10)$  или  $U_2(8) - U_2(10) \leq \epsilon$ , то модель определена.

**Пример 9.** Смоделировать барботажный аппарат, схема которого определена на рис. 13.

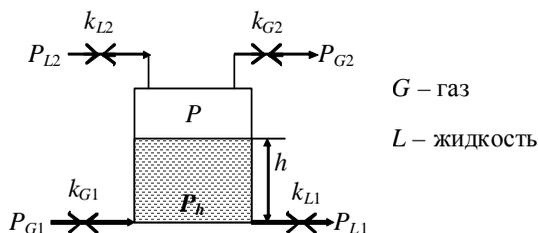


Рис. 13

Условия: давления на входах и выходах известны, также известны коэффициенты пропускной способности.

Определить расходы газа и жидкости через аппарат, давление в аппарате, давление на дне аппарата.

$$U_{G1} = k_{G1} \sqrt{P_{G1} - P_h}, \quad (58)$$

$$U_{G2} = k_{G2} \sqrt{P - P_{G2}}, \quad (59)$$

$$U_{L1} = k_{L1} \sqrt{P_h - P_{L1}}, \quad (60)$$

$$U_{L2} = k_{L2} \sqrt{P_{L2} - P}. \quad (61)$$

Неизвестные переменные:  $U_G, U_L, P, P_h, h, \rho$ .

$$P_h = P + \rho gh, \quad (62)$$

$$\rho = f(\rho_L, U_G, P), \quad (63)$$

где  $\rho_L$  – плотность жидкости (const);  $\rho$  – плотность газожидкостной эмульсии.

Составим информационную матрицу (табл. 12).

## 12. Информационная матрица системы уравнений, описывающей стационарный режим гидравлической системы

Номер уравнения	$U_G$	$U_L$	$P$	$P_h$	$h$	$\rho$	Порядок решения
1	<+>			[+]			1
2	(+)		<+>				2
3		<+>		(+)			3
4		<+>	(+)				6
5			(+)	(+)	<+>	(+)	5
6	(+)		(+)			<+>	4

$U_L(3) = U_L(4)$  или  $U_L(3) - U_L(4) < \varepsilon$  – условие окончания алгоритма.

### ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

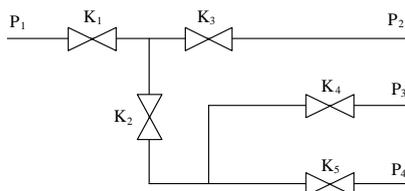
1. Определить уравнения расходов через сужающие устройства гидравлической системы, а также давления в точках разветвления системы (варианты контрольных заданий и исходные данные приведены ниже).

2. Закончить полученную систему уравнений для расходов до числа неизвестных давлений в точках разветвления уравнениями для узлов гидравлической системы.

3. Решить полученную систему уравнений на ЭВМ.

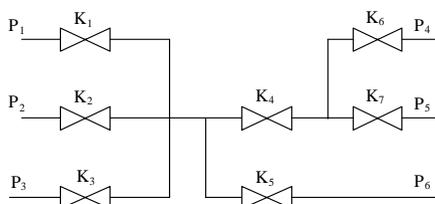
## ВАРИАНТЫ ЗАДАНИЙ

1.



$$K_1 = 0,75; K_2 = 2,2; K_3 = 1,9; K_4 = 0,3; P_1 = 4,7; P_2 = 2,8; P_3 = 1,9; P_4 = 0,5.$$

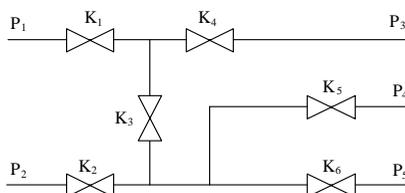
2.



$$K_1 = 0,6; K_2 = 1,2; K_3 = 2,5; K_4 = 3; K_5 = 0,8; K_6 = 0,8; K_7 = 1,5;$$

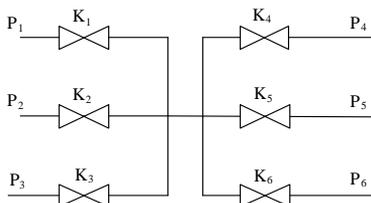
$$P_1 = 8; P_2 = 3; P_3 = 4,5; P_4 = 5; P_5 = 2; P_6 = 1.$$

3.



$$K_1 = 2,5; K_2 = 1,4; K_3 = 2; K_4 = 1,3; P_1 = 7; P_2 = 8; P_3 = 2,2; P_4 = 3,5; P_5 = 2.$$

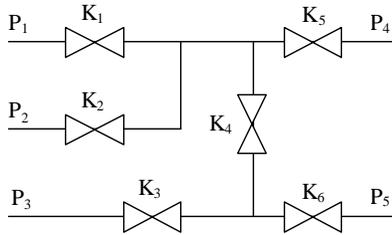
4.



$$K_1 = 1,6; K_2 = 0,8; K_3 = 2,5; K_4 = 2; K_5 = 1,3; K_6 = 1,4;$$

$$P_1 = 7; P_2 = 4; P_3 = 3,5; P_4 = 2; P_5 = 2,4; P_6 = 1,2.$$

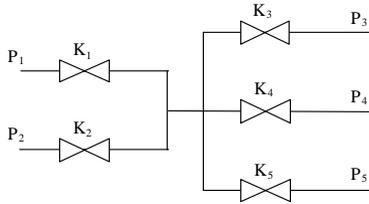
5.



$$K_1 = 2,9; K_2 = 1,8; K_3 = 1,5; K_4 = 2,5; K_5 = 4; K_6 = 3;$$

$$P_1 = 8; P_2 = 5; P_3 = 4; P_4 = 3; P_5 = 2.$$

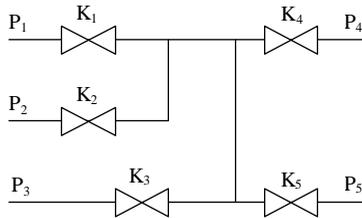
6.



$$K_1 = 2,2; K_2 = 1,6; K_3 = 0,5; K_4 = 1,1; K_5 = 0,3;$$

$$P_1 = 5; P_2 = 3; P_3 = 1,5; P_4 = 2; P_5 = 1,4.$$

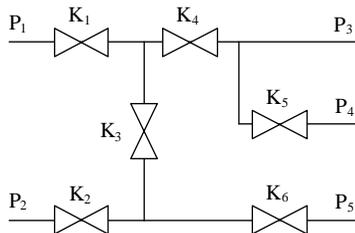
7.



$$K_1 = 2,3; K_2 = 1,2; K_3 = 1,4; K_4 = 1,5; K_5 = 3;$$

$$P_1 = 5; P_2 = 6; P_3 = 4; P_4 = 2; P_5 = 3.$$

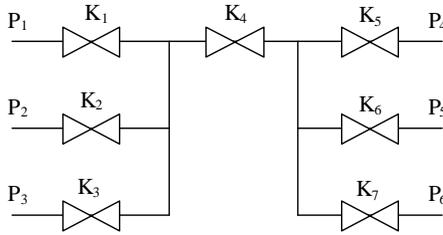
8.



$$K_{11} = 1; K_{12} = 1,5; K_3 = 0,6; K_4 = 1,5; K_5 = 0,5; K_6 = 0,8;$$

$$P_1 = 7; P_2 = 5; P_3 = 1,5; P_4 = 2; P_5 = 1,5.$$

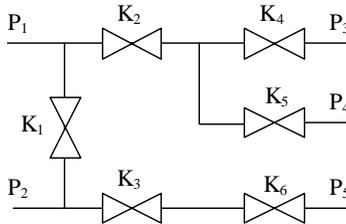
9.



$$K_1 = 0,7; K_2 = 0,4; K_3 = 0,8; K_4 = 1,5; K_5 = 2; K_6 = 1; K_7 = 2,1;$$

$$P_1 = 3; P_2 = 2,5; P_3 = 2; P_4 = 1; P_5 = 0,5; P_6 = 0,9.$$

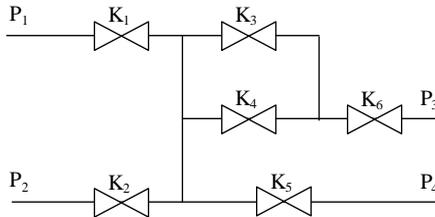
10.



$$K_1 = 0,9; K_2 = 2,4; K_3 = 1,8; K_4 = 1,5; K_5 = 2; K_6 = 1;$$

$$P_1 = 9; P_2 = 6; P_3 = 4; P_4 = 5; P_5 = 4.$$

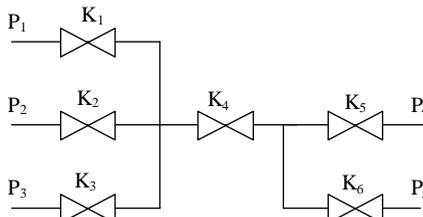
11.



$$K_1 = 1,4; K_2 = 2,3; K_3 = 0,4; K_4 = 1; K_5 = 3,8; K_6 = 2,1;$$

$$P_1 = 5; P_2 = 3; P_3 = 1; P_4 = 2.$$

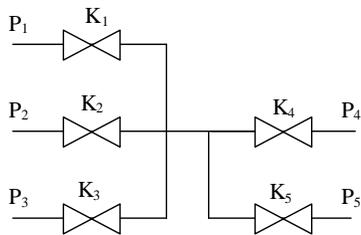
12.



$$K_1 = 2,7; K_2 = 1,4; K_3 = 1,8; K_4 = 2,5; K_5 = 1,8; K_6 = 0,8;$$

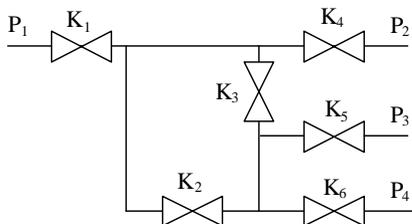
$$P_1 = 4; P_2 = 3,5; P_3 = 2; P_4 = 1; P_5 = 1,5.$$

13.



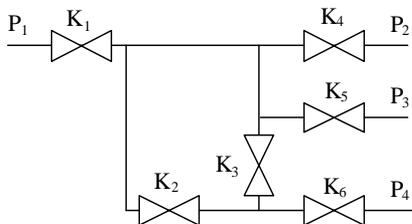
$K_1 = 2,2; K_2 = 2,4; K_3 = 1,2; K_4 = 0,8; K_5 = 2,8;$   
 $P_1 = 3; P_2 = 5,5; P_3 = 2; P_4 = 1; P_5 = 1,5.$

14.



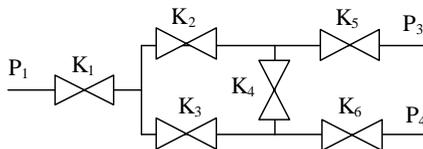
$K_1 = 0,8; K_2 = 1; K_3 = 1,2; K_4 = 1,5; K_5 = 2; K_6 = 1,4;$   
 $P_1 = 10; P_2 = 2; P_3 = 4; P_4 = 5.$

15.



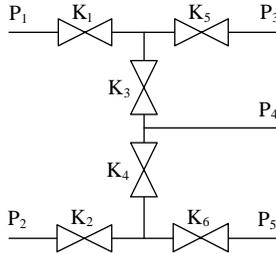
$K_1 = 1,8; K_2 = 1; K_3 = 2,2; K_4 = 0,5; K_5 = 2; K_6 = 1,4;$   
 $P_1 = 7; P_2 = 4; P_3 = 3; P_4 = 2.$

16.



$K_1 = 0,9; K_2 = 1,1; K_3 = 1,4; K_4 = 2,5; K_5 = 2; K_6 = 2,4;$   
 $P_1 = 7,8; P_2 = 3,6; P_3 = 4.$

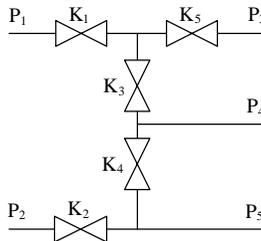
17.



$$K_1 = 2,6; K_2 = 2,1; K_3 = 0,4; K_4 = 1,5; K_5 = 1; K_6 = 2,4;$$

$$P_1 = 4; P_2 = 2,6; P_3 = 1; P_4 = 1,5; P_5 = 0,5.$$

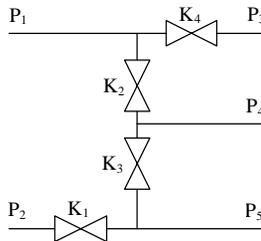
18.



$$K_1 = 1,6; K_2 = 2,1; K_3 = 3,4; K_4 = 1,5; K_5 = 1;$$

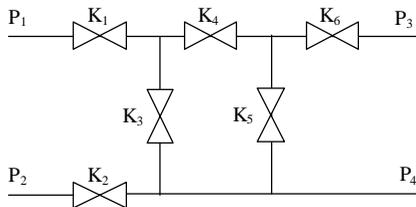
$$P_1 = 5; P_2 = 7,6; P_3 = 2; P_4 = 1,5; P_5 = 3,5.$$

19.



$$K_1 = 0,6; K_2 = 1,2; K_3 = 2,4; K_4 = 1,8; P_1 = 4; P_2 = 3; P_3 = 2; P_4 = 1; P_5 = 0,5.$$

20.



$$K_1 = 1,6; K_2 = 1,1; K_3 = 1,4; K_4 = 2,5; K_5 = 1; K_6 = 1,4; P_1 = 5; P_2 = 3,6; P_3 = 2; P_4 = 1,5.$$

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие основные допущения принимаются при компьютерном моделировании простой гидравлической системы?
2. Как описывается движение потока жидкости через клапан?
3. Как и с какой целью строится информационная матрица системы уравнений математического описания?
4. С какой целью проводится анализ параметрической чувствительности модели?
5. Как провести анализ параметрической чувствительности компьютерной модели?

## Лабораторная работа 7

### МОДЕЛИРОВАНИЕ ПРОЦЕССА ПОЛУЧЕНИЯ ПАРА

*Цель работы:* освоить методику математического моделирования сложных технологических процессов на ЭВМ.

*Задания работы:*

1. Ознакомиться с основными принципами построения математических моделей на основе балансных уравнений.
2. Выполнить вариант задания.

*Содержание отчёта:*

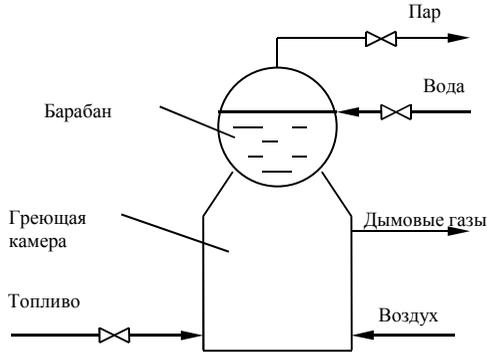
1. Вариант задания.
2. Математическая модель процесса.
4. Результаты анализа полученных статических и динамических характеристик объекта.
5. Результаты определения значений настроек ПИ-регулятора АСР.
6. Результаты анализа влияния значений настроек регулятора на качество переходных процессов в АСР.
4. Тексты программ.

### КРАТКИЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Тепловая энергия пара широко используется для отопления, вентиляции и горячего водоснабжения жилых и промышленных помещений.

Для выбора режимов работы, построения системы автоматического регулирования процессами, протекающими при производстве пара, необходимо провести анализ входных и выходных потоков технологического процесса и получить математическое описание объекта управления.

На рисунке 14 представлена упрощённая схема парового котла.

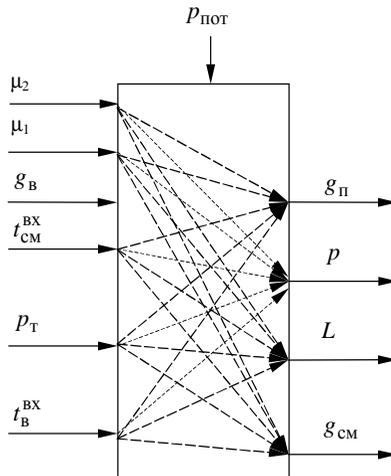


**Рис. 14. Упрощённая схема парового котла**

Проведём анализ процесса производства пара как объекта управления.

Основным качественным выходным параметром процесса производства пара является давление генерируемого пара  $p$ .

Входными воздействиями (регулирующими и возмущающими) для данного процесса являются проводимость клапана на линии подачи топлива  $\mu_2$ , проводимость клапана на линии отбора пара  $\mu_1$  (или расход потребляемого пара  $g_{\text{п}}$ ), температуры подаваемой топливной смеси  $t_{\text{см}}^{\text{вх}}$  и питающей воды  $t_{\text{в}}^{\text{вх}}$ , давление в линии подачи топлива  $p_{\text{т}}$  и давление в линии потребления пара  $p_{\text{пот}}$  (рис. 15).



**Рис. 15. Связь между входными и выходными параметрами**

В качестве регулирующего воздействия может быть выбрана проводимость клапана на линии подачи топлива в топку, а остальные воздействия являются возмущающими.

Основным возмущающим воздействием на процесс производства пара является колебание давления в линии потребителя пара.

На основе проведённого анализа разрабатывается математическое описание процесса.

При разработке математического описания приняты следующие допущения.

1. Объём воды и парового пространства считается объектом с сосредоточенными параметрами и принимается идеальное перемешивание в объёме.

2. Тепловой ёмкостью поверхности нагрева пренебрегаем.

3. Паровая среда представляется как идеальный газ.

4. При температуре меньше температуры кипения парообразования не происходит.

5. Потеря тепла через ограждения в окружающую среду не происходит.

6. Температуры воды и пара равны.

7. Удельные теплоёмкости воды, пара, газозооушной смеси постоянны.

8. Давление в камере сгорания постоянно.

Составим балансные уравнения для процесса производства пара.

Уравнение материального баланса для воды в барабане котла имеет вид:

$$\frac{dG_B}{d\tau} = g_B - g_{II}^H, \quad (64)$$

где  $G_B$  – масса воды в барабане котла, кг;  $g_B, g_{II}^H$  – расходы входящей питающей воды и испаряющегося пара, соответственно, кг/с.

Уравнение энергетического баланса для воды в барабане котла имеет вид:

$$c_B \frac{d(G_B t_B)}{d\tau} = kF(t_K - t_B) + g_B c_B t_B^{BX} - g_{II}^H i_{II}, \quad (65)$$

где  $c_B$  – удельная теплоемкость воды, Дж/(кг·°C);  $t_B$  – температура воды в барабане котла, °C;  $t_B^{BX}$  – температура питающей воды, °C;  $t_K$  – температура в греющей камере (топке), °C;  $k$  – коэффициент теплопередачи через поверхность нагрева, Вт/(м<sup>2</sup>·°C);  $F$  – поверхность теплопередачи, м<sup>2</sup>;  $g_{II}^H$  – расход испаряющегося пара, кг/с;  $i_{II}$  – удельная энтальпия испаряющегося пара, Дж/кг.

Продифференцируем (65) как сложную функцию и подставим в выражение (64):

$$c_B \frac{d(G_B t_B)}{d\tau} = c_B \frac{dG_B}{d\tau} t_B + c_B G_B \frac{dt_B}{d\tau} = c_B (g_B - g_B^H) t_B + c_B G_B \frac{dt_B}{d\tau}.$$

Тогда выражение (65) запишется в виде:

$$c_B G_B \frac{dt_B}{d\tau} = kF (t_K - t_B) - g_B c_B (t_B - t_B^{BX}) - g_B^H (i_B - c_B t_B).$$

С учётом того, что удельная теплота парообразования  $\lambda$  равна

$$\lambda = i_B - c_B t_B,$$

уравнение энергетического баланса для воды в барабане запишется в виде:

$$c_B G_B \frac{dt_B}{d\tau} = kF (t_K - t_B) - g_B c_B (t_B - t_B^{BX}) - g_B^H \lambda. \quad (66)$$

Уравнение материального баланса для пара в котле имеет вид:

$$\frac{dG_{II}}{d\tau} = g_{II}^H - g_{II}, \quad (67)$$

где  $G_{II}$  – масса пара в паровом пространстве, кг;  $g_{II}$  – расход отбираемого пара, кг/с.

Согласно принятым допущениям пар подчиняется закону идеального газа Менделеева–Клапейрона:

$$pV_{II} = \frac{G_{II}}{M} R (t_B + 273), \quad (68)$$

где  $p$  – давление пара в котле, Па;  $V_{II}$  – объём парового пространства, м<sup>3</sup>;  $M$  – молярная масса воды;  $R$  – универсальная газовая постоянная.

Объём парового пространства связан с объёмом (или массой воды) в котле соотношением:

$$V_{II} = V_o - V_B = V_o - \frac{G_B}{\rho_B}, \quad (69)$$

где  $V_o$ ,  $V_B$  – общий объём котла и объём воды в котле, м<sup>3</sup>;  $\rho_B$  – плотность воды, кг/м<sup>3</sup>.

Температура кипения воды (и температура пара) является функцией давления:

$$t_B = f(p). \quad (70)$$

Расход пара через клапан может быть описан выражением:

$$g_{II} = \mu_1 \sqrt{p^2 - p_{\text{пот}}^2}, \quad (71)$$

где  $\mu_1$  – проводимость клапана.

Уравнение энергетического баланса для газозвдушной среды в камере сгорания (топке) записывается в виде:

$$c_k G_k \frac{dt_k}{d\tau} = -kF(t_k - t_b) - g_{cm} c_k (t_k - t_{cm}^{BX}) + g_{cm} r, \quad (72)$$

где  $c_k, G_k$  – удельная теплоёмкость и масса газовой среды в камере сгорания;  $g_{cm}$  – расход топлива;  $r$  – удельная теплота сгорания топлива.

Тепловая емкость камеры сгорания значительно ниже тепловой емкостью нагреваемой воды, поэтому динамикой изменения температуры в топке  $t_k$  можно пренебречь. В результате уравнение (9) запишется в виде:

$$-kF(t_k - t_b) - g_{cm} c_k (t_k - t_{cm}^{BX}) + g_{cm} r = 0. \quad (73)$$

Расход топлива также может быть описан зависимостью

$$g_{cm} = \mu_2 \sqrt{P_T^2 - P_K^2}, \quad (74)$$

где  $\mu_2$  – проводимость клапана на линии подачи топлива.

**Имитационные исследования объекта управления.** При автоматизации производственных процессов наиболее правильные решения могут быть приняты на основании данных о статических и динамических свойствах регулируемого объекта. Для получения этих данных определяют соответственно статические и динамические характеристики объекта управления, используя его математическое описание.

Для проведения анализа чувствительности выходных параметров процесса производства пара на изменение входных параметров, а также для выбора регулирующих воздействий проведём построение статических характеристик процесса.

Для этого в математическом описании процесса (64), (66) – (74) приравняем производные в дифференциальных уравнениях к нулю и получим следующую систему алгебраических уравнений:

$$g_b - g_{II}^H = 0, \quad (75)$$

$$kF(t_k - t_b) - g_b c_b (t_b - t_b^{BX}) - g_{II}^H \lambda = 0, \quad (76)$$

$$g_{II}^H - g_{II} = 0, \quad (77)$$

$$pV_{II} = \frac{G_{II}}{M} R (t_b + 273), \quad (78)$$

$$V_{II} = V_o - V_b = V_o - \frac{G_b}{\rho_b}; \quad (79)$$

$$t_b = f(p); \quad (80)$$

$$g_{II} = \mu_1 \sqrt{P^2 - P_{ПOT}^2}; \quad (81)$$

$$-kF(t_k - t_b) - g_{cm} c_k (t_k - t_{cm}^{BX}) + g_{cm} r = 0; \quad (82)$$

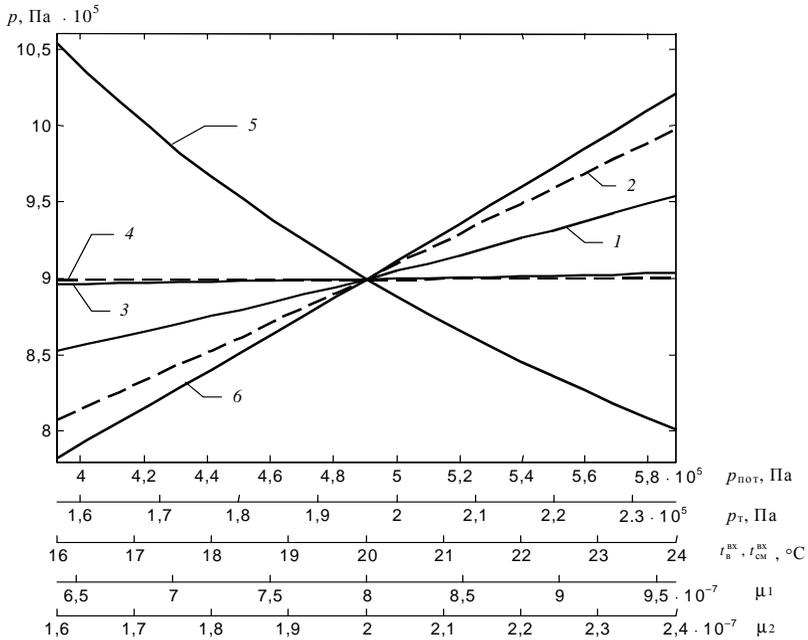
$$g_{cm} = \mu_2 \sqrt{p_T^2 - p_K^2}. \quad (83)$$

Для получения статических характеристик объекта управления входные воздействия необходимо менять в диапазоне  $\pm 20\%$  от своего значения в номинальном режиме.

Графики статических характеристик приведены на рис. 16.

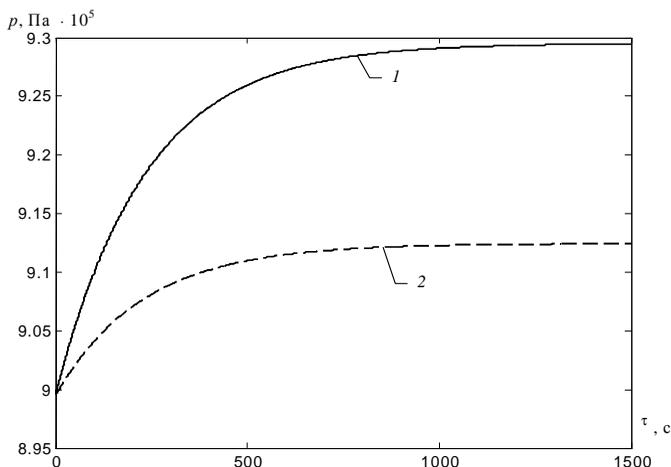
Из анализа статических характеристик можно сделать вывод, что давление обладает наибольшей чувствительностью по изменению проводимости клапанов на трубопроводах подвода топливной смеси и отбора пара из парового котла. Следовательно, проводимость клапанов можно использовать в качестве регулирующего воздействия. Давление в линии подачи топливной смеси можно считать постоянным. Основным возмущением является давление потребляемого пара, так как чувствительность давления пара в котле на изменение этого параметра наибольшая.

Анализ динамических характеристик котла проведём по переходным функциям, построенным для наиболее сильнодействующего возмущения  $p_{пот}$  и регулирующего воздействия  $\mu_2$  (рис. 17). При этом значения входных воздействий менялись на 5% от своего значения в номинальном режиме.



**Рис. 16. Статические характеристики парового котла:**

1 –  $p(p_{пот})$ ; 2 –  $p(p_T)$ ; 3 –  $p(t_B^{BX})$ ; 4 –  $p(t_{cm}^{BX})$ ; 5 –  $p(\mu_1)$ ; 6 –  $p(\mu_2)$

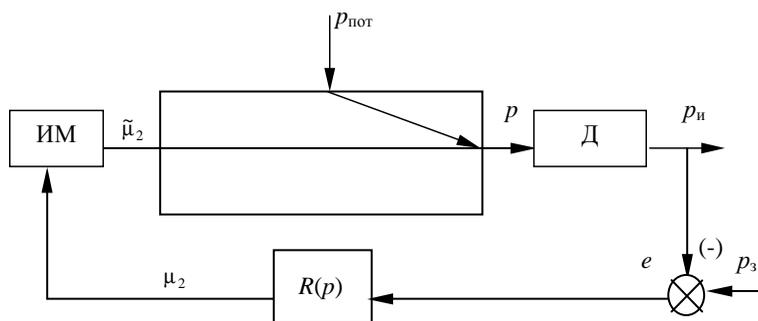


**Рис. 17. Переходные функции парового котла при изменении:**  
 $1 - \mu_2$ ;  $2 - p_{\text{пот}}$

**Моделирование одноконтурной системы регулирования давления в паровом котле.** Анализируя переходные функции можно сделать следующие выводы:

- объект обладает свойством самовыравнивания;
- время перехода из одного состояния в другое составляет  $\approx 1300$  с.

Структурная схема одноконтурной АСР давления пара представлена на рис. 18.



**Рис. 18. Структурная схема системы регулирования давления пара:**  
 $x$  – вход объекта;  $p$  – давление – выходная регулируемая величина;  
 $p_3$  – заданное значение температуры;  $p_n$  – измеренное значение температуры;  
 $e$  – ошибка регулирования

Помимо собственно объекта регулирования в систему регулирования входят чувствительный элемент (Д), исполнительный механизм с регулирующим органом (ИМ) и регулятор (R). Поэтому в математическое описание АСР входят уравнения, описывающие характеристики датчика, исполнительного механизма и регулятора.

Динамические характеристики датчика и исполнительного механизма представляются в виде дифференциальных уравнений:

$$T_D \frac{dp_{и}}{d\tau} + p_{и} = p ; \quad (84)$$

$$T_{ИМ} \frac{d\tilde{\mu}_2}{d\tau} + \tilde{\mu}_2 = \mu_2 , \quad (85)$$

где  $T_D, T_{ИМ}$  – постоянные времени датчика давления и исполнительного механизма, с;  $p_{и}$  – измеряемое давление, Па;  $\tilde{\mu}_2$  – реальная проводимость клапана на линии подачи топлива.

Постоянные времени используемого датчика давления  $T_D = 0,5$  мин, а исполнительного механизма  $T_{ИМ} = 1$  мин.

Расчёт одноконтурной АСР заключается в определении настроечных параметров функции  $R_1(p)$ . В практике автоматизации промышленных объектов широкое распространение получили ПИ-регуляторы. Для регулирования давления в паровом котле уравнение ПИ-регулятора имеет вид:

$$\mu_2 = \mu_2^{\text{НОМ}} + K_p \left( e + \frac{1}{T_{и}} \int e d\tau \right) , \quad (86)$$

где  $K_p, T_{и}$  – параметры настройки регулятора;  $e$  – ошибка регулирования.

Ошибка регулирования определяется как разница между заданным значением и текущим значением измеренного давления в паровом котле:

$$e = p_3 - p_{и} .$$

При заданном законе регулирования расчёт одноконтурной АСР сводится к определению оптимальных (с точки зрения какого-либо критерия качества) настроечных параметров регулятора с учётом выполнения ограничения на запас устойчивости системы.

Наиболее часто в качестве критерия качества используется интегральный квадратичный критерий при ограничении на степень затухания:

$$I = \int_0^{\tau_{\text{кон}}} e^2 d\tau \rightarrow \min_{K_p, T_{и}} ,$$

где  $\tau_{\text{кон}}$  – время регулирования.

На рисунках 19, 20 показаны графики переходных процессов в одно-контурной АСР при найденных настройках регулятора. При этом задание регулятору давления пара в парогенераторе изменялось с  $8,995 \cdot 10^5$  Па до  $9,995 \cdot 10^5$  Па. По каналу возмущения – давление пара в линии потребления изменялось с  $4,905 \cdot 10^5$  Па до  $6,867 \cdot 10^5$  Па.

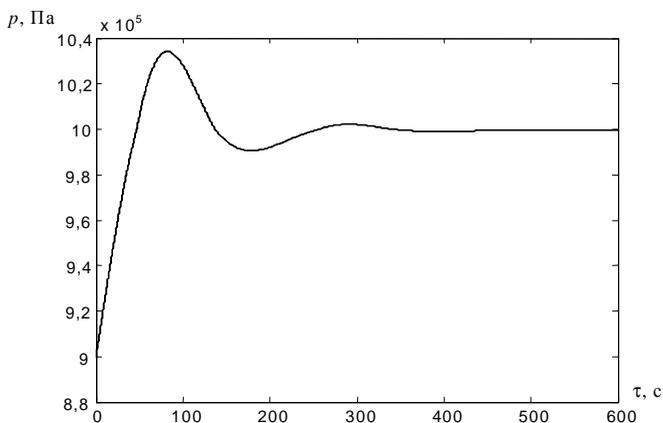


Рис. 19. Переходной процесс в АСР при изменении задания

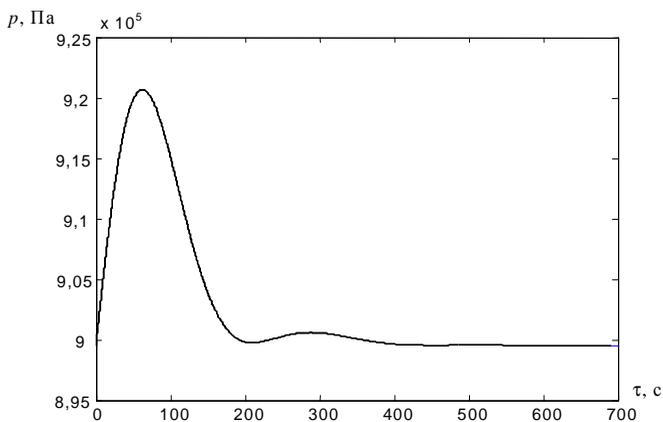


Рис. 20. Переходной процесс в АСР при действии возмущения

**Анализ качественных показатели переходных процессов в АСР.** Качество регулирования численно может быть охарактеризовано прямыми показателями качества, которые определяются непосредственно из графиков переходных процессов АСР.

В табл. 13 сведены прямые показатели качества процесса регулирования в АСР (с ПИ-регулятором).

### 13. Прямые показатели качества процесса регулирования

Внешнее воздействие	$t_{\text{рег}}, \text{с}$	$\xi$	$X_m$	$\psi$
Возмущение	420	–	0,1	0,86
Изменение задания	350	0,035	–	0,91

Программа построения переходных процессов в АСР давления пара в котле представлена ниже.

#### ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. В соответствии с заданным вариантом исходных данных процесса получения пара в паровом котле (варианты заданий выдаются преподавателем индивидуально) необходимо исследовать статические и динамические характеристики процесса.

2. Определить значения параметров регулятора и произвести построение переходных процессов в АСР.

3. Исследовать влияние значения параметров настройки регулятора на качество переходных процессов. Для этого значения настроек регулятора необходимо менять относительно найденных значений в большую и меньшую сторону.

#### Программа для расчёта уравнений математической модели и построения переходных процессов в АСР

```

clc % очистка командного окна
dt=0.05; % шаг интегрирования по времени
tk=1000; % конечное время интегрирования д.у.
t=[0:dt:tk]; % вектор времени
n=length(t); % размерность вектора времени
% создание векторов выходных переменных процесса
t_v=zeros(1,n); % температура воды в барабане
t_k=zeros(1,n); % температура в камере сгорания
p=zeros(1,n); % давление пара в барабане
M_v=zeros(1,n); % масса воды в барабане
M_p=zeros(1,n); % масса пара в паровом пространстве
G_p=zeros(1,n); % расход отбираемого пара
pi=zeros(1,n); % измеренное значение давления пара
% начальные условия

```

```

t_v(1)=173.9754;
t_k(1)=278.1349;
p(1)=8.9951e+005;
M_v(1)=2.5e+003;
pi(1)=p(1);
M_p(1)=32.6929;
% описание теплофизических характеристик
c_v=1*4.19e3;           % удельная теплоемкость воды
c_k=1500;              % удельная теплоемкость газовой среды
M=18e-3;               % молярная масса воды
R=8.31;                % универсальная газовая постоянная
rho_p=8.4;             % плотность пара
rho_k=1;                % плотность газовой среды
rho_v=1000;            % плотность воды
r_t=3.6e7;             % удельная теплота сгорания топлива
Lamda=1945e3;          % удельная теплота парообразования
% описание характеристик парового котла и значений входов
V_0=10;                % общий объем котла
K=1500;                % коэффициент теплопередачи
F=10;                  % поверхность теплопередачи
G_v_n=0.6032;          % расход воды в номинальном режиме
t_v_vx=20;             % температура воды на входе в котел
t_t_vx=20;             % температура газовой среды на входе

p_t=2*9.81e4;          % давление газовой среды на входе
p_pot=5*9.81e4;        % давление пара в линии потребления
p_k=1*9.81e4;          % давление в камере сгорания

% проводимости клапанов
mu1=0.0000008;
mu2_n=0.0000002;
% характеристики насыщенного пара
xx=[0.98 0.99 1 1.01 1.013 1.02 2 3 4 5 6 7 8 9 10 20 40]*1e5;
% давление насыщенного пара
yy=[99.23 99.46 99.7 99.93 100 100.27 120 133 143 151 158 164 169 174 179
212 250];
% температура насыщенного пара
% переменные для определения интегральных составляющих
ПИ-регуляторов
sum=0;
sum2=0;
% настройки ПИ-регуляторов
Kp2=0.0000000000002;
Ti2=100;

```

```

Kp=20;
Ti=20;
p_z=9.9951e+005;      % заданное значение давления пара
M_v_z=2500;          % номинальное (заданное) значение массы воды
% характеристики датчика и ИМ
Td=10;              % постоянная времени датчика
Tim=20;             % постоянная времени ИМ
% цикл решения уравнений модели методом Эйлера
for i=2:n
    % поддержание номинальной массы воды в котле
    eps=M_v(i-1)-M_v_z; % ошибка регулирования
    sum=sum+eps;
    G_v=G_v_n-Kp*(eps+sum*dt/Ti); % уравнение регулятора
    % ограничение на расход воды
    if G_v<=0
        G_v=0;
    end
    if G_v>=5;
        G_v=5;
    end
    % определение давление пара по температуре воды
    p(i-1)=interp1(yu,xx,t_v(i-1),'linear');
    % определение измеренного значения давления
    pi(i)=pi(i-1)+dt*((p(i-1)-pi(i-1))/Td);
    eps2=-pi(i)+p_z; % ошибка регулирования
    sum2=sum2+eps2;
    mu2_=mu2_n+Kp2*(eps2+sum2*dt/Ti2); % уравнение регулятора
    % ограничение на проводимость клапана
    if mu2_ < 0
        mu2_=0;
    end
    if i==2
        mu2_old=mu2_;
    end
    % определение реальной проводимости клапана
    mu2=mu2_old+dt*(mu2_-mu2_old)/Tim;
    mu2_old=mu2;
    % решение уравнений модели
    V_p=V_0-M_v(i-1)/ro_v;

    M_p(i)=p(i-1)*V_p*M/(R*(t_v(i-1)+273));

    if p_pot > p(i-1)

```

```

G_p(i-1)=0;
else G_p(i-1)=mu1*sqrt(p(i-1)^2-p_pot^2);
end

G_pi=G_p(i-1)+(M_p(i)-M_p(i-1))/dt;
if G_pi<=0
G_pi=0;
end

M_v(i)=M_v(i-1)+dt*(G_v-G_pi);

G_sm=mu2*sqrt(p_t^2+p_k^2);
if p_t < p_k
G_sm=0;
end
t_k(i)=(K*F*t_v(i-1)+G_sm*(c_k*t_t_vx+r_t))/(K*F+G_sm*c_k);

t_v(i)=t_v(i-1)+dt*(K*F*(t_k(i)-t_v(i-1))-G_v*c_v*(t_v(i-1)-t_v_vx)-
G_pi*Lamda)/(c_v*M_v(i));

end
% построение графиков зависимостей
figure(1)
plot(t,t_v);
figure(2)
plot(t,t_k);
figure(3)
p(end)=p(end-1);
plot(t,p);
figure(4)
G_p(end)=G_p(end-1);
plot(t,G_p);
figure(5)
plot(t,M_v);

```

## ЗАКЛЮЧЕНИЕ

В учебном пособии изложены научные основы моделирования систем, рассмотрены возможности компьютерного моделирования, показана ориентация основ моделирования на новые информационные технологии в исследованиях, проектировании и управлении технологическими и техническими системами, обучении студентов и специалистов.

Моделирование стало эффективным средством исследования систем. Актуальность математических моделей при выполнении условия адекватности реальным системам непрерывно возрастает из-за гибкости, невысокой стоимости реализации на базе современных ЭВМ. Особенно эффективно применение моделирования на этапах проектирования изделий, технологического оборудования и систем управления, когда особенно высока цена ошибочных решений.

Использование ресурсов современной вычислительной техники позволило увеличить сложность моделей при исследовании систем, учитывающих многообразие факторов, имеющих место в реальных технологических и технических системах.

Принимая во внимание тенденции развития технического прогресса и возрастающие возможности вычислительной техники, дальнейшее развитие теории моделирования систем следует ожидать в направлении создания интеллектуальных самосовершенствующихся моделей, способных без участия разработчика изменять свою структуру и подстраиваться под требования новых задач.

## СПИСОК ЛИТЕРАТУРЫ

1. Лазарев, Ю.Ф. Начала программирования в среде MATLAB : учебное пособие / Ю.Ф. Лазарев. – К. : НТУУ «КПИ», 2003. – 424 с.
2. Поляков, К.Ю. Лабораторные работы по теории автоматического управления. Исследование САУ в среде MATLAB / К.Ю. Поляков. – URL : <http://kpolyakov.narod.ru/uni/labs.htm>.
3. Лазарев, Ю.Ф. Моделирование процессов и систем в MATLAB : учебный курс / Ю.Ф. Лазарев. – СПб. : Питер : Издательская группа BHV, 2005. – 512 с.
4. Закгейм, А.Ю. Общая химическая технология: Введение в моделирование химико-технологических процессов : учебное пособие / А.Ю. Закгейм. – М. : Логос, 2009. – 304 с.
5. Гартман, Т.Н. Компьютерное моделирование простых гидравлических систем / Т.Н. Гартман, В.Н. Калинин, Л.И. Артемьева ; под общ. ред. д-ра техн. наук Т.Н. Гартмана. – М. : РХТУ им. Д.И. Менделеева, 2002. – 40 с.

## ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ .....	3
1. КРАТКОЕ ОПИСАНИЕ РАБОТЫ В MATLAB .....	4
1.1. Работа в командном режиме .....	4
1.2. Программирование в среде MATLAB .....	10
2. КРАТКОЕ ОПИСАНИЕ РАБОТЫ В SIMULINK .....	17
2.1. Создание моделей с SIMULINK .....	17
Лабораторная работа 1 ИССЛЕДОВАНИЕ ЛИНЕЙНОЙ СТАЦИОНАРНОЙ ДИНАМИЧЕСКОЙ СИСТЕМЫ В СРЕДЕ MATLAB .....	21
Лабораторная работа 2 МОДЕЛИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ В ПАКЕТЕ SIMULINK ...	32
Лабораторная работа 3 ПОСТРОЕНИЕ МАТЕМАТИЧЕСКОГО ОПИСАНИЯ ОБЪЕКТА УПРАВЛЕНИЯ ЭКСПЕРИМЕНТАЛЬНЫМИ МЕТОДАМИ .....	36
Лабораторная работа 4 ПОСТРОЕНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ АНАЛИТИЧЕСКИМ МЕТОДОМ (МОДЕЛИРОВАНИЕ ТЕПЛООБМЕННОЙ АППАРАТУРЫ) ....	44
Лабораторная работа 5 ПОСТРОЕНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ АНАЛИТИЧЕСКИМ МЕТО- ДОМ (МОДЕЛИРОВАНИЕ ХИМИЧЕ- СКИХ РЕАКЦИЙ) .....	59
Лабораторная работа 6 МОДЕЛИРОВАНИЕ ПРОСТЫХ ГИДРАВЛИЧЕСКИХ СИСТЕМ .....	71
Лабораторная работа 7 МОДЕЛИРОВАНИЕ ПРОЦЕССА ПОЛУЧЕНИЯ ПАРА .....	81
ЗАКЛЮЧЕНИЕ .....	94
СПИСОК ЛИТЕРАТУРЫ .....	95