



АУЭТ

Образован в 1975

**Noncommercial Joint-Stock
Company**

**ALMATY UNIVERSITY OF
POWER ENGINEERING AND
TELECOMMUNICATIONS**

Automation and Control
Department

MODELING AND IDENTIFICATION OF CONTROL OBJECTS

Methodical guidelines to laboratory work for students
of specialty 05070200 - Automation and control

Almaty 2019

COMPILED BY: Ibrayeva L.K., Tashibayeva A.E. Modeling and identification of control objects. Methodical guidelines to laboratory works for students of specialty 5B070200 - Automation and control - Almaty: AUPET, 2019 - 55 p.

Methodical instructions contain descriptions to 10 laboratory works and propose performance of these works with use of various tools of dynamic modeling system of MatLab, and also the program of processing of experiments results for identification of the linear object model on the frequency characteristic developed at Department "Automation and Control".

Methodical instructions are used at performance of laboratory works on discipline "Modeling and identification of control objects".

Ill - 5, table -1, bibliography – 5.

Reviewer: c.t.c., docent Musapirova G.L.

Printed according to the plan for publications of NC JSC "Almaty university of power engineering and telecommunications" for 2019 y.

NC JSC "Almaty university of power engineering and telecommunications", 2019 y.

Content

1 Laboratory work №1. Research of linear time-invariant systems.....	
2 Laboratory work №2. Modeling of object with lumped parameters by analytical method	
3 Laboratory work №3. The study of static and dynamic characteristics of the process in the tank.....	
4 Laboratory work №4. Linearization of non-linear equations of the object to determine the transfer functions.....	
5 Laboratory work №5. Direct methods of identification of linear objects of the first order.....	
6 Laboratory work №6. The identification of linear object using the frequency characteristics.....	
7 Laboratory work №7. Parametric identification of linear dynamic objects.....	
8 Laboratory work №8. Identification of linear objects using the System Identification Toolbox.....	
9 Laboratory work №9. Identification of second-order linear objects.....	
10 Laboratory work №10. Nonparametric identification of dynamic objects.....	
References.....	

Introduction

To develop an object control system, it is necessary to have a mathematical description of the object for any moment of time with a sufficiently high accuracy. The problem of creation a mathematical model of the control object in General is to determine the operator of the system, which determines the change in the output value at an arbitrary change in the input action.

The most widely used type of operator of the control object model is differential equations. These equations can be composed, firstly, on the basis of basic physical laws and are called *analytical* models of the object. Another approach is based on information about the object obtained as a result of observations of it in a passive and active experiment. Thus obtained models are called *empirical* (experimental). The problems of construction of such models are studied in the *theory of identification*.

The proposed methodical instructions to laboratory works on discipline "Modeling and identification of control objects" include 10 works and are intended to studying of methods of analytical modeling and identification. To perform these works, various tools of the MatLab software are mainly used.

When working in MatLab, the command window of the system is actively used: for *direct calculations*, for operational use of *built-in procedures*, for *data exchange* between individual tools of the system, etc.

Control System Toolbox is used to create and analyze models of linear stationary systems. It provides a wide range of procedures that analyze of the automatic control system from different points of view.

The popularity of the MatLab system is promoted by its powerful expansion Simulink, which provides the user with convenient and simple tools, including visual object-oriented programming, for block modeling of linear and nonlinear dynamic systems.

Simulation of systems and devices, the behavior of which depends on external events, in the MatLab + Simulink system is implemented using a special extension *Stateflow*. It has own graphical user interface that allows you to create graphical charts that serve to visually represent the operation of the simulated system.

System Identification Toolbox is used to identify the system and contains a graphical user interface for the construction of correlation, spectral and parametric models of different orders.

To solve the problem of approximation of discrete values of functions obtained as a result of calculations, the *Curve Fitting Toolbox* is used.

In addition to these MatLab tools system a special software for the studying of object identification procedures using frequency response, developed at the Department of "Automation and Control" is used.

1 Laboratory work №1. Research of linear time-invariant systems

Work purpose: creation of dynamic systems models in Command Window of MatLab; using Control System Toolbox for models analyses.

1.1 Assignment for the laboratory work

In the process of fulfilling the laboratory work the student should to:

- create the model of the system in the form of the transfer function;
- to build the equivalent ‘State-space’ and ‘Zero-pole’ model;
- to determine the gain coefficients in the steady state and the bandwidth of the system;
- to learn how to build the impulse and transfer responses, a map of the location of zeroes and poles, frequency response;
- to learn how to use *Control System Toolbox* to construct different characteristics;
- to learn how to build output processes of the linear system at the arbitrary input signal.

1.2 Basic models of linear dynamic objects

In studying the phenomena of nature, in solution of many physics and technology, chemistry, biology and other sciences problems is not always possible to establish a direct relationship between the quantities describing a particular evolutionary process. But in most cases it is possible to establish a relationship between variables (functions) and speeds of their changes relatively to other (independent) variables, i.e., find equations in which the derivative of unknown functions are used (the derivative characterizes the speed of function changing, physical meaning of derivative – speed). Such equations are called a *differential one*.

Differential equations (systems of differential equations) describing the behavior of the Control Object, are the most common form of presentation of the model operator. An order system of differential equations describing the model of the object is not directly determined by the number of inputs and outputs, and depends on the operators that transforms the input signals into the output.

Here are the main types of models of linear time-invariant dynamic objects:

- *differential equations*. The most universal model based on differential equations is described by the equation:

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_1 \frac{dy}{dt} + a_0 y = b^m \frac{d^m x}{dt^m} + b_{m-1} \frac{d^{m-1} x}{dt^{m-1}} + \dots + b_1 \frac{dx}{dt} + b_0 x,$$

where $y(t)$ – output process;

$x(t)$ – input process;

a_i, b_j , ($i = 0, n, j = 0, m$) – constant coefficients;
 n, m ($n \geq m$) – constants.

In the *linear* case, the coefficients a_i and b_j do not depend on $x(t)$, $y(t)$ and their derivatives. If, in addition, they do not depend on time, we get an equation with *constant coefficients*. If these coefficients are time-dependent, then the equation is called a linear equation with variable coefficients;

– *space-state equations*. If you select n system (object) coordinates as its state variables (such coordinates, for example, can be the output signal $y(t)$ and $n-1$ its derivatives) $x_i(t)$, $i=1,2,\dots,n$, this system can be described by the equations for *the* state variables:

$$\frac{dX}{dt} = A \cdot X + B \cdot u;$$

$$y(t) = C \cdot X + D \cdot u,$$

where $X(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$ – space state vector;

A - coefficients matrix;

B - control matrix;

C - output matrix;

D - matrix characterized the relationships between output and input signals.

In the particular case, in simple systems, the output is a scalar value. In this case, the matrix C is a row vector, and matrix X is a column vector and their product gives a scalar value;

– *difference equations*. For objects whose functioning is presented for discrete time $t_k = kT$ (in this case T is the sampling interval), that is, for discrete objects, the most general description is a difference equation (analogue of the differential):

$$y_k = a_1 y_{k-1} + \dots + a_{na} y_{k-na} = b_1 u_k + b_2 u_{k-1} + b_3 u_{k-2} + \dots + b_{nb} u_{k-nb+1},$$

where $y_{k-i} = y[(k-i)T]$;

$u_{k-j} = u[(k-j)T]$.

In a number of cases, the response of a linear time-invariant system to simple input is determined by its dynamic characteristics. The main ones are:

– *transfer function* $W(p)$ (the ratio of the Laplace-converted output signal to the Laplace-converted input signal):

$$W = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0};$$

– *impulse function* $g(t)$ – the response of the previously undisturbed object on the input signal in the form of δ -functions.

With zero initial conditions, the relationship between output and inputs is described by the convolution integral:

$$y(t) = \int_{-\infty}^{+\infty} g(t - \tau)u(\tau)d\tau,$$

where $g(t)$ – impulse response.

- *transition function* $h(t)$ – the response of the previously undisturbed object on the step input signal;
- *the frequency characteristics* of the object, which are determined by a complex transmission coefficient $W(j\omega) = W(p)|_{p=j\omega}$.

These models are equivalent, that is, knowing any of them, you can get all the rest.

1.3 Implementation and study of models of linear time-invariant systems in MatLab

In MatLab *Control System Toolbox* is used for the study of linear time-invariant systems. The toolbox implements methods of investigation of dynamic systems based on the use of transfer functions and state space models.

You can enter the model in *Control System Toolbox* in three forms: in the form of state space matrices, in the form of coefficients of transfer functions numerators and denominators, in the form of zeros, poles and gain coefficient of the system.

Some procedures of creating LTI models:

- *tf* - creates a model by the given transfer functions of the system;
- *ss* - creates the state space model by the given matrices of the state equations of the system;
- *dss* - creates a similar model by the description of the state space of more general type when the equations of state variables are not resolved concerning derivatives;
- *zpk*- creates a model according to the specified zeros, poles and transfer coefficients of the system;
- *filt* - creates a model by discrete transfer functions, written in the form of polynomials of z^{-1} ;
- *set* - assigns values to some other fields of objects (the names of inputs and outputs, names of systems, etc.).

These procedures allow you to create both continuous and discrete models. In addition, there tools of transform models from one specified form to another exists.

The transfer function in MatLab is introduced as a ratio of two polynomials of the complex variable s . Polynomials are stored as arrays of coefficients written in descending order of degrees. An object of class *tf* describing the transfer function are created in memory. The semicolon at the end of the command suppresses the output on the screen.

By transfer functions, you can easily build a model in the form of zeros and poles¹:

```
>>f_zpk = zpk(f).
```

To convert the transfer function into the state-space model the following command is used:

```
>>f_ss = ss( f ).
```

One of the most important characteristics of a linear system is the gain in the steady state or *static gain coefficient*.

To find the static gain of the model f in MatLab, the following command is used:

```
>> k = dcgain( f ).
```

To calculate the *bandwidth* of the system ω_b , the following command is used

```
>> b = bandwidth ( f ).
```

To build the frequency response in MatLab, you must first create an array of frequencies in the desired range. You can use the function *linspace*(an uniform distribution of points on a linear scale) and *logspace* (uniform distribution of points on a logarithmic scale).

Command:

```
>> w = linspace (0, 10, 100)
```

builds an array of 100 points with constant step in the range from 0 to 10, and the command:

```
>> w = logspace (-1, 2, 100)
```

builds an array of 100 points with a uniform step on a logarithmic scale in the interval from 10^{-1} to 10^2 . To not reflect the entire array on the monitor screen, put a semicolon after these commands.

Frequency response on the grid for w of the linear model f (given as the transfer function model in state space or in the form of ‘zero-pole’) is calculated with the function *freqresp*:

```
>> r = freqresp(f, w).
```

The *freqresp* function returns a three dimensional array. This is due to the fact that it is also applicable to multidimensional models (with multiple inputs and outputs), which transfer function is a matrix. The first two indices denote the row and column in the matrix and the third index is a number of points of the frequency characteristic. For a system with one input and one output, it is convenient to convert a three-dimensional array in a one-dimensional by the command:

¹ Note that when working in the command window MatLab at the end of the command either any sign is not put (in this case, the result of the command is displayed) or a semicolon is put to suppress the output to the screen. Here in after, if the punctuation mark "dot" or "comma" are put, it is because of the necessity to keep the Standard of the text design requirements adopted in AUPET.


```
>> r = r(:).
```

To plot frequency response, you can use the commands in MatLab:

```
>> plot ( w, abs(r) );  
>> semilogx ( w, abs(r) );  
>> loglog ( w, abs(r) ).
```

In the first case, the scale of both axes is linear, while in the second case we use a logarithmic scale on the x-axis (frequency) and the latter a logarithmic scale in both axes. To compute the phase (in degrees), the following command is used:

```
>> phi = angle(r)×180/pi,
```

then a phase-frequency characteristics is build, for example:

```
>> semilogx ( w, phi ).
```

Poles and zeroes. To find the poles of the transfer function f you can use the function:

```
>> p = pole ( f ).
```

Calling of the function:

```
>> [w0,zeta,p] = damp ( f )
```

allows you to find not only the poles p , but also the corresponding natural frequency $w0$ and damping coefficients $zeta$ in the form of arrays.

The zeros of the transfer function f is calculated as:

```
>> z = zero ( f ).
```

The stability of the system does not depend on the location of zeros, but they significantly influence on transient processes. Command:

```
>> pzmap ( f )
```

builds a map of the location of the zeroes (they are marked by circles) and poles (they are marked by crosses) of the system at complex plane.

A convenient tool for viewing the time and frequency characteristics of the model is the LTI-Viewer interactive browser. To call the GUI from the command window, use the *ltiview* function, and the browser's working window appears on the screen. To exchange data between the working environments of MatLab and LTI-Viewer, you should use the menu commands of the browser window File / Import and File / Export. After loading the model, a graph of the model's transition function is displayed in the browser window. With the help of the browser, the frequency and time characteristics, zeros and poles of the system are easily calculated and displayed in the form of graphs and diagrams.

1.4 The order of fulfilling the laboratory work

Clear the MatLab workspace (memory) - *clear all*; clear MatLab window -

clc.

Look through short help for the *tf* command - *help tf*.

For the transfer function:

$$W(p) = \frac{b_2 p^2 + b_1 p + b_0}{p^3 + a_2 p^2 + a_1 p + a_0}$$

in the Command Window of MatLab enter the vectors of coefficients of the numerator *num* and the denominator *den* of the transfer function (options are given in table 1.1):

```
>> num=[b2 b1 b0];
```

```
>> den = [1 a2 a1 a0].
```

Using command *model = tf(num,den)* create the object model.

Convert continuous model to discrete one:

```
>> disc=c2d(model, 0.01).
```

Here 0.01 – sampling step, and back to continuous:

```
>> cont=d2c(disc).
```

Make sure that the reverse transformation gives the initial model. As a result of these actions in the Workspace you have three models: *model*, *disc*, *cont* are saved.

Find the zeroes and poles of transfer function.

Find the gain in the steady state:

```
>> k = dcgain (f).
```

Determine the bandwidth of the system (lowest frequency where the frequency response becomes smaller than- 3 dB).

```
>> b = bandwidth ( f ).
```

Create a "state- space" model of the system:

```
>> f_ss = ss(f).
```

Create a model of the original system in the form of "zeros and poles" and check that this corresponds to the original model:

```
>> f_zp = zpk( f ).
```

Built the location of the system's zeros and poles at the figure:

```
>> pzmap ( f ).
```

Determine the damping coefficients and natural frequencies for all elementary units (first and second order):

```
>> [wc,ksi,p] = damp ( f ).
```

Type *ltiview* in Command Window. Choose File/Import in the appeared

window. Dialog box for selecting of the model is appeared. Your model will also be in this list.

Download your *model*. Build transient characteristics of systems: *context menu – Plot Types – Step*.

Build the impulse response (weighting function) of this system: *context menu – PlotTypes – Impulse*.

Download model *f_ss*: *File – Import*.

Check if the impulse response of the second system is built: *Context Menu – Systems*.

Build transient characteristics of systems.

On the chart for each function note: the maximum, transition time², rise time (from 10 % to 90 % steady-state value), the steady-state value: *context menu – Characteristics: Peak Response, Settling Time, Rise Time, Steady State*.

Clicking on circle-labels, display frame with numerical values for these parameters, and place them so that all the numbers were visible.

Choose *Edit/Plot Configuration*, the dialog box appears; choose here the set of characteristic. You can chose 6 characteristic at the same time. Select two characteristics: step response and impulse response.

Using context menu, set: labels on the graph; the size, style and other possible properties of the graph.

Do the following:

- choosing menu *File/Toolbox* set the parameters of the toolkit;
- choosing menu *Preferences/File/Line Style* select the style of graphic image.
- choosing menu *Preferences File/LTI Viewer Preferences* set the parameters of the time and frequency vectors.

Close LTIViewer.

Create an array of frequencies for plotting frequency characteristics (100 points in the interval from 10^{-1} to 10^2 with a uniform distribution on logarithmic scale):

```
>>w = logspace(-1, 2, 100).
```

Calculate frequency response of the initial system³and build it on the axes with the logarithmic scale on the *x*-axes:

```
>> r = freqresp( f, w );  
>> r = r(:);  
>>semilogx ( w, abs(r) ).
```

Explain where on the chart you can find the gain in static mode and how to determine the bandwidth of the system.

² By default, MATLAB transition time is determined for a 2% deviation from the steady-state value.

³ The frequency characteristic is returned in the form of three-dimensional massive in which element has 3 indexes: row, column (for multi-dimensional models) and number of the points of frequency characteristic. For the systems with one input and one output command $r = r(:)$; transform this data in usual one-dimensional array.

Close all unnecessary windows except the Matlab command window.

Construct a signal simulating rectangular pulses of a single amplitude with a period of 4 seconds (5 pulses in total):

```
>>[u,t] = gensig('square',4).
```

Fulfill the simulation and plot the output signal of system f at the given input:

```
>>lsim (f, u, t).
```

Table 1.1 - Assignment options

Option	b ₂	b ₁	b ₀	a ₂	a ₁	a ₀
1	1.0	1.10	0.100	3.0000	3.1600	1.2000
2	1.1	1.54	0.495	2.8000	2.9200	1.2000
3	1.2	1.08	0.096	2.3727	2.2264	0.9091
4	1.3	1.04	0.091	2.1909	2.0264	0.9091
5	1.4	-1.54	0.252	1.8333	1.5278	0.6944
6	1.5	-0.90	-0.240	1.6667	1.3611	0.6944
7	1.6	0.80	-0.224	1.3286	0.8959	0.4592
8	1.7	1.36	0.204	1.1857	0.7673	0.4592
9	1.8	-1.98	0.432	1.2000	0.7644	0.3556
10	1.9	-0.76	-0.399	1.3333	0.8711	0.3556
11	2.0	0.60	-0.360	1.2000	0.7406	0.2734
12	2.1	1.68	0.315	1.3250	0.8281	0.2734
13	2.2	-2.42	0.616	1.3059	0.7696	0.2076
14	2.3	-0.46	-0.552	1.4235	0.8401	0.2076
15	2.4	0.24	-0.480	1.3889	0.7531	0.1543
16	2.5	2.25	0.500	1.5000	0.8086	0.1543
17	2.6	0.26	-0.780	1.2421	0.6139	0.1108
18	2.7	-0.27	-0.810	1.1368	0.5717	0.1108
19	2.8	0.28	-0.840	0.8000	0.3700	0.0500
20	2.9	3.19	0.870	0.7000	0.3500	0.0500

1.5 Content of report

The report on laboratory work must contain:

- the option number;
- the screenshots of the Command window with systems' models;
- the results of calculations, graphics of the systems with setting properties (an explanatory inscription, legend, grid, color and type of fonts, etc.).

1.6 Control questions

1.6.1 Define the following concepts:

- the transfer function;
- zeroes and poles of the transfer function;
- impulse response (weighting function); -
- the transition function;

- the frequency response;
- the model in the state-space;
- model of the form of "zeros and poles";
- the gain in static mode;
- the system bandwidth;
- the time of the transition process;
- the natural frequency of the vibration element;
- the damping coefficient of the vibrating element.

1.6.2 List the procedures of creating LTI models.

1.6.3 The purpose of Control System Toolbox.

1.6.4 The units of measure: the gain in static mode; the system bandwidth; the transition process time; the natural frequency of the vibrating element; the damping coefficient of the vibrating element.

1.6.5 Dependency between natural frequency and time constant of the vibrating element?

1.6.6 How does influence the coefficient of direct transmission (matrices in the state space model) on the static gain?

1.6.7 What features does the module LTIViewer?

1.6.8 How to find: the gain in steady state by the amplitude-frequency characteristic; the system bandwidth by the amplitude-frequency characteristic?

1.6.9 How to build an array of 200 values in the range from 10^{-3} to 10^3 with a uniform distribution on a logarithmic scale?

1.6.10 What values are on the axes at the chart of the amplitude-frequency characteristic?

2 Laboratory work №2. Modeling of object with lumped parameters by analytical method

Work purpose: the acquisition of skills of application of the fundamental conservation laws for modeling objects with lumped parameters; the mastering of techniques for modeling and analysis of objects in Simulink of MatLab.

2.1 Assignment for the laboratory work

In the process of fulfilling the laboratory work the student must:

- study analytical methods for the modeling of Control Objects with the lumped parameters [1,2];
- be familiar with the techniques of modeling objects in the Simulink of MatLab system [1,3];
- understand the modeling process;
- build a block-diagram of a dynamic model of the considered process in Simulink;
- in the purpose of the study to implement simulation experiments on the model;

- analyse the obtained results; to make the conclusions.

2.2 Process description

The studying object is the mill to which input stream of initial recyclable material Φ_0 that contains large fractions. The output stream from the mill Φ_{out} is proportional to the material mass M in the mill:

$$\Phi_{out} = \alpha M,$$

where α is the constant of proportionality.

In the mill there is a grinding of the large fraction with velocity W , which is proportional to the large fraction mass and is described by the following relationship:

$$W = k(M)(1 - C)M,$$

where $C = \frac{m}{M}$, $k(M) = \frac{k_0(2M^+ - M)}{M^+}$.

Here m is the mass of small fraction in the mill;

C - is the share of small fraction;

k_0 (1/час)- is the constant of the grinding velocity;

M^+ - is the mass corresponding to the maximum grinding speed (maximum productivity of the mill).

Value $M^+ = 10 m$; $k_0 = 1,5$ (1/hour). The mill productivity is a function of the output stream $P = C \cdot \Phi_{out}$, i.e. $P = \alpha \cdot m(t)$.

Mathematical model of the process in the form of a system of differential equations is derived at the base of the analytical modeling methods. When compiling the equations of the mathematical model of the process, the following balance equations are used:

а) balance of the total mass in the mill:

$$\frac{dM}{dt} = \Phi_0 - \Phi_{out};$$

б) balance of small fractions mass (final product) in the mill:

$$\frac{d(CM)}{dt} = W - C \cdot \Phi_{out}.$$

The final system of the differential equations describing the considered process has the form:

$$\begin{cases} \frac{dM}{dt} = \Phi_0 - \alpha M, \\ \frac{dm}{dt} = 0,15 \cdot (20 - M) \cdot (M - m) - \alpha \cdot m. \end{cases} \quad (2.1)$$

2.3 Building block-diagram of the object's model in Simulink

Simulink is intended to the mathematical modeling of linear and nonlinear dynamical systems and devices, which are represented by their functional block-diagram.

Simulink of MatLab has an extensive library of block components and convenient block diagrams editor to build of a functional block- diagram of the modeling devices. Input of the studied systems' characteristics is performed in the dialog mode, through the graphical collecting of the elementary parts connection. As a result of this collection the model of the studied system is formed, which is called *S-model*. The model is kept in a file with the extension *.mdl*.

Simulink automates the following most time-consuming stage of modeling: it constructs and solves complex systems of algebraic and differential equations describing the given functional scheme (the model), providing convenient and visual control over the behavior of the virtual device created by the user.

To build *S-model* the blocks from Simulink Library are used. The library of Simulink blocks is a set of visual objects, using which you can collect, as from cubes, an arbitrary construction. For any block you can get the required number of copies and use each of them independently. Moreover, for almost all units there is a possibility of individual settings: the user can change both internal parameters of the blocks (for example, the number of inputs) and external design (size, color, name, etc). Any restrictions are also not imposed on the order of the connection of blocks with each other. Of course, when linking blocks, it is necessary to follow certain rules, which are due mainly to the logic of working of the model itself, and not to the special requirements of Simulink.

The library can be filled with users blocks.

The model of the considered process is a system of differential equations to determine the mass of large $M(t)$ and small $m(t)$ fractions.

To convert the model equations (2.1) into the block-diagram the following idea is applied. Initially the right-hand sides of the equations are formed, assuming that the variables M and m are known. For this purpose the blocks *Fcn* are used (for each equation is used a separate instance of the block), in which the expressions for the equations right parts are formed (by the rules of programming languages). *Fcn* block has only one input port. If you need to give to *Fcn* input several signals, they should be combined into one vector by *Mux* block. The input signal (scalar or vector) of the *Fcn* block is designated by the letter u ; if this signal is a vector, then its components are denoted by $u(1), u(2), \dots$, numbering of the vector components is performed in the order which they are given on the *Mux* block. The values of the expressions for dM/dt and dm/dt are obtained as a result of the forming of the right parts of the equations.

Integrating these expressions with the help of the special block *Integrator*, we obtain the desired values of $M(t)$ and $m(t)$. The obtained values can now be used for the formation of the right parts of equations (by the feedback). At the same time you can display them on the viewing window *Scope*.

2.4 The order of fulfilling the laboratory work

Run Simulink from the Command Window of Matlab, choosing *File/New/Model* or press the appropriate button on the Toolbar.

In the opened window, create an empty *untitled* window - a blank for the new model. Start to collect a model block-diagram of the considered process.

Note: always give meaningful names (names of variables used in the description of the process and in the equations) to the blocks in block diagrams,. This will help in the future to easily navigate in the diagram.

Form the input flow Φ_0 (the options are given in the table 2.1):

- place in the diagram window, the *Mux* blocks (to combine the variables into one vector), *Fcn* (to form the expressions of the input stream), *Constant* (to set the values of the expression coefficients). Since the input stream is specified as an explicit function of time, the *Clock* unit is needed;

- configure the number of input ports of *Mux* block, connect it with *Fcn* block;

- open the settings window of *Fcn* block, write down the expression for the input stream, using as input vector the output signals of the *Mux* block;

- the output of the *Fcn* block leave free.

Collect the block-diagram for the first equation of the process model:

- place in the diagram window the *Mux* blocks (for combining of all input variables in one vector), *Fcn* (to form the expression of the right-hand side of the equation), *Integrator* (to integrate the right-hand side of the equation) and *Scope* (to look through the result of integration);

- connect the created blocks in listed order;

- set up the number of inputs of the *Mux* block (on one input of a *Mux* block give a signal from *Fcn*);

- configure the *Fcn* block, using as input variables the input stream function Φ_0 and the value of $M(t)$ resulting from the modeling;

- apply the output of *Fcn* block to the input of the *Integrator* block;

- connect the output of the *Integrator* with *Scope* block, as well by inverse connection to the input of the *Mux* block.

Similarly to p. 2.4.3 collect the diagram for the second equation; here as input variables are used the values $M(t)$ obtained from the diagram of the first equation and the values of $m(t)$ resulting from the modeling; the output of the *Fcn* block also is supplied to the *Integrator* block, the output of which will be the desired value.

Connect output $m(t)$ of the second equation diagram with *Gain* block and receive the investigated productivity reflect P in the *Scope* window.

Save the file with the model.

Fulfill the simulation experiments on the model. During experiments it is necessary to change the meanings of the coefficients of the input stream (in *Constant* blocks), run the diagram and track charts of the output variable.

You have to choose of the coefficients values by no arbitrary way, but highlight areas of possible, valid, special, etc. variables, and explore the trend of the behavior of the output variable in these ranges.

Find the most productive modes.

Table 2.1 – Assignment options

№	Φ_0	Range
1	$5,2\sin(\omega t + \varphi)$	$\omega = 0:0.5:1; \varphi = 30:15:60$
2	$3.7\cos(\omega t - \varphi)$	$\omega = 0:0.5:1; \varphi = 30:15:60$
3	$\frac{a_1 + a_2 t^2}{t}$	$a_1 = 1:0.5:2; a_2 = 0.1:0.05:0.2$
3	$3.5tg(\omega t + \varphi)$	$\omega = 0:0.5:1; \varphi = 30:15:60$
4	$\frac{a_1 t + a_2 t^2}{a_1 + a_2}$	$a_1 = 1:0.5:2; a_2 = 0.1:0.05:0.2$
6	$\sqrt{a_1 + a_2 t^2}$	$a_1 = 1:0.5:2; a_2 = 0.1:0.05:0.2$
7	$2.3ctg^2(\omega t + \varphi)$	$\omega = 0:0.5:1; \varphi = 30:15:60$
8	$\sqrt{\frac{t}{a_1 + a_2 t^2}}$	$a_1 = 1:0.5:2; a_2 = 0.1:0.05:0.2$
9	$2.1\sin^3(\omega t + \varphi)$	$\omega = 0:0.5:1; \varphi = 30:15:60$
10	$\sqrt[3]{a_1 + a_2 t^2}$	$a_1 = 1:0.5:2; a_2 = 0.1:0.05:0.2$
11	$1.5\cos(\omega_1 t) + \sin(\omega_2 t)$	$\omega_1 = 0:0.5:1; \omega_2 = 0:0.3:0.9$
12	$\frac{a_1 t + a_2 t^2}{\sqrt{a_1 + a_2}}$	$a_1 = 1:0.5:2; a_2 = 0.1:0.05:0.2$
13	$2.4\cos(\omega_1 t) - 3.4\sin(\omega_2 t)$	$\omega_1 = 0:0.5:1; \omega_2 = 0:0.3:0.9$
14	$\frac{a_1 t^3 + a_2 t^2}{\sqrt{a_1 + a_2}}$	$a_1 = 1:0.5:2; a_2 = 0.1:0.05:0.2$
15	$5.7tg(\omega_1 t) + \sin(\omega_2 t)$	$\omega_1 = 0:0.5:1; \omega_2 = 0:0.3:0.9$
16	$\frac{\sqrt{a_1 t}}{a_1 + a_2 t}$	$a_1 = 1:0.5:2; a_2 = 0.1:0.05:0.2$
17	$4.2\cos^2(\omega_1 t) + \sin^2(\omega_2 t)$	$\omega_1 = 0:0.5:1; \omega_2 = 0:0.3:0.9$
18	$\frac{\sqrt{a_1 t + a_2 t^2}}{a_1 - a_2}$	$a_1 = 1:0.5:2; a_2 = 0.1:0.05:0.2$
19	$3.6\sqrt{\cos(\omega_1 t) + \sin(\omega_2 t)}$	$\omega_1 = 0:0.5:1; \omega_2 = 0:0.3:0.9$
20	$\frac{5a_1 t + \sqrt[3]{a_2 t^2}}{a_1 + a_2}$	$a_1 = 1:0.5:2; a_2 = 0.1:0.05:0.2$

2.5 Content of report

The report on laboratory work must contain:

- work purpose and assignment for the laboratory work;
- the equations of the considered process model;
- the block-diagram of the model with explanatory comments;
- the listings of the most significant experimental results;
- conclusions of the work.

2.6 Control questions

2.6.1 What are the main types of mathematical models?

2.6.2 What is the difference between analytical and experimental methods of modeling?

2.6.3 What conservation laws are used for the derivation of the equations of the considered process model?

2.6.4 What class is the model of the considered process belonged to?

2.6.5 How does equation define whether it is a dynamic or not by the model?

2.6.6 Explain the purpose of the Simulink.

2.6.7 Explain the order of creating block-diagrams for differential equations.

2.6.8 Explain the purpose of blocks *Clock*, *Integrator*, *Fcn*, *Gain*, *Mux*, *Scope*, *Constant* for the realized diagram.

2.6.9 Why must be implement an input stream used the *Clock* block?

2.6.10 What letter is used to the input signal of the *Fcn* block?

3 Laboratory work №3. The study of static and dynamic characteristics of the process in the tank

Work purpose: mastering the practical skills of modeling and analysis of dynamic characteristics of objects in Simulink; learn event simulation Stateflow Tool of MatLab.

3.1 Assignment for laboratory work

In the process of carrying out the laboratory work the student must:

- study the Stateflow Tool of MatLab [1,3];
- familiarize with the modeling object;
- build the block-diagrams of static and dynamic models of the considered object in Simulink;
- fulfill calculations by the static and dynamic models in the different periods of the overall modeling interval;
- implement simulation experiments on modeling of object's controlling.

3.2 The object description

The investigated object is the tank with independent input of the fluid flow $G_{in}(t)$ and the dependent output flow $G_{out}(t)$. The flow at the output is determined by the liquid level over the output hole H and by the square of section area of output hole f_{out} . The amount of liquid in the tank, above the level of the output hole is determined from the relation:

$$M = F \cdot H \cdot \rho ,$$

where F is the square of the cross sectional area of the tank;
 ρ is the density.

At drawing up of the object's mathematical model equations there used:

– the equation of material balance of the system:

$$\frac{dM(t)}{dt} = G_n(t) - G_c(t); \quad (3.1)$$

– the equation describing the conservation law of motion:

$$G_c(t) = \mu \cdot f_c \cdot \rho \cdot \sqrt{2 \cdot g \cdot H}. \quad (3.2)$$

As the coordinates of the object's state we choose the level of liquid in the tank H and the output flow G_{out} .

Static model of the object will look like:

$$H = \frac{1}{2g\rho^2 \cdot \mu^2} \cdot \frac{G_n^2}{f_c^2},$$
$$G_c = G_n = const. \quad (3.3)$$

The object dynamic is described as a differential equation:

$$\frac{dH(t)}{dt} = \frac{1}{F\rho} \cdot [G_n(t) - \mu \cdot \rho \cdot \sqrt{2g} \cdot \sqrt{H(t)} \cdot f_c(t)], \quad (3.4)$$

here for G_{out} the relation (3.2) remains.

In the laboratory work you have to fulfill calculations by static (3.3) or dynamic (3.4) models on different intervals of the whole modeling interval. Transient processes in the object are the result of changes in input variables $G_{in}(t)$ and f_{out} .

To switch between static and dynamic models, as well as to transmit new values of parameters to the model it is necessary to use the subsystem *Statefiow*.

3.3 Stateflow subsystem for modeling of object's controlling

The main purpose of Simulink is to implement simulation of the systems. The effect of external events (situations) on the behavior of the systems is not taken into account or is reflected in the temporal dependencies on the impact sources. Meanwhile, many systems operate under conditions where the occurrence of certain events may materially affect on their work. The modeling system whose behavior depends on external events (situations) is called *event-driven* or situational. In the Simulink of MatLab this modeling is implemented by using special extensions *Stateflow*. The tool has a graphical user interface that allows you to create graphical diagrams, called *SF-diagrams*. These diagrams serve as visual representations of the work of the modeled system. It is achieved by analysis of all stages of its work with pointing the blocks which are active and passive in the moment and the transitions between them according to the results of the analysis of certain conditions.

To create Stateflow-diagram you must first create a model in Simulink, or open an existing one. Then you need to choose in the Library a block that refers to a chart Stateflow, and transfer it to your model. After double-clicking on it with the mouse, Stateflow window (*Chart*) will open (figure 3.1) in which, using the menu and toolbar, you can describe the conditions, the relationships between them, identify variables, switching conditions and other elements required to the control.

At the Stateflow diagram there are two main groups of elements: graphic and nongraphic. All of the graphical elements of the diagram is represented on the left panel of the main window of the states map graphical editor.

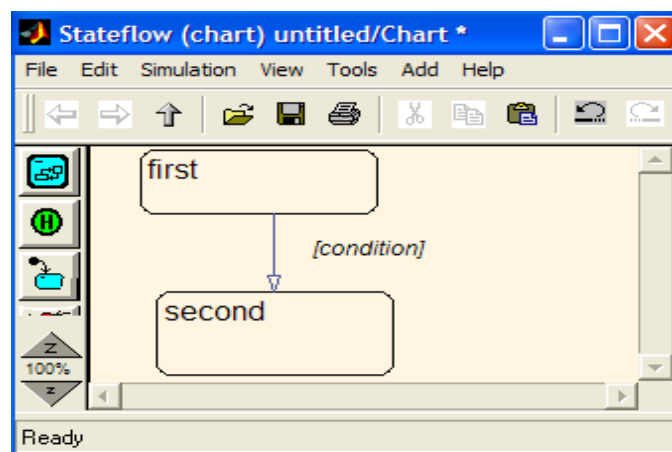


Figure 3.1 – The example of the simple *SF*-diagram

An important object of the *SF*-diagram is the *state*. Each state describes one mode of the event-driven system operation. State becomes active if the condition for the transition leading to it, gets the value “true” or if this state is initial. Each state in the diagram has its parent and can have descendants. If the state is unique, then its parent is the *SF*-diagram itself (*root chart*).

To create the graphic image of the state you must select the corresponding button on the panel, and then click in the place where it should be placed.

The state has text labels that define the actions fulfilled during its activity. The state name is entered first. When describing the state the following actions can be defined:

- *name* – state name;
- *entry* – the action executed on the entering to state;
- *during* - the action executed while the state is active;
- *exit* – action executed on exiting the state;
- *on*: {event name} is the action executed at the time of occurrence of the event (the name of which is specified in brackets <>), provided that the system will remain in this state.

One of the more important graphical objects of *SF*-diagrams – the *transitions*. The transitions represent the connection of one state with another, and are usually represented by arrows, which beginning applies to the source state, and end - to the destination state. The transitions do not have their buttons on the toolbar. To draw a transition, it is enough to set the mouse at the source state press its left button and hold it, stretching the arrow that appears to the destination state. The transitions have labels which describe the conditions under which a transition occurs from one state to another. The text of the label has the following format:

Event [condition] {condition_action}/transition_action,

where *Event* – defines the event that initiates the transition. If the event is not specified, then the transition is initiated by the execution of the Boolean expression. If the events that initiate this transition, are several, they are all listed, separated by OR operator;

condition – the Boolean expression that initiates the transition, becoming true;

condition_action – the action which executes after a true condition for the transition, but before the entire transition is determined true (and determine the destination state);

transition_action – defines the actions executed when the transition occurs, if all the preceding conditions are true and the destination state is already defined.

The most important concept of the Stateflow tool – *Event*; it is a non-graphical objects that control the diagram. For visualization of the event you can use transitions labels. All the events related to the Stateflow diagram must be defined. The following types of events: input, output, local, imported from a source which is external relatively to Stateflow diagram and Simulink; exported to a destination, which is external relatively to Stateflow diagram and Simulink.

The editor of *SF*-diagrams has the menu *Add/Event*, with the help of which you can specify event type and its properties.

Data is non-graphical objects in the Stateflow diagram for the storage of numerical data. Variables can be used at any level of the hierarchy. To create and modify the data, *Add/Data* menu should be used.

3.4 The order of fulfilling the laboratory work

In Simulink build the block-diagrams of static and dynamic model of the object.

Parameter values and initial variable values of the object are presented in table 3.1.

Decorate them as a subsystem - *Subsystem*, i.e. the *S*-models containing models of the lower level. One of the ways of creating a subsystem is to allocate all the blocks, which should be included in the subsystem and to select the appropriate point (*Create Subsystem*) from the context menu.

The connection between the subsystem with the main system is fulfilled by entering in subsystem the standard building blocks *In* (Input) and *Out* (Output). All values which are formed in the main model, and are used in the subsystem must come into the subsystem through the blocks *In*, and the values are generated in the subsystem and then used in the main system, should come out of the subsystem via the blocks *Out*. Let us recall that it is recommended to give meaningful names to variables (blocks) in the block-diagrams.

Review the purpose and order the use of the *Switch* block [1,3]; place in the diagram window this block.

Use the *Switch* block to switch between static and dynamic models: the system time is served on the control port of this block (*Clock* block), outputs of static and dynamic models are served the information ports.

Execute the simulation experiments on the developed diagram:

- the time interval ($t_0 - t_1$) reflects the steady state;
- the time interval ($t_1 - t_2$) reflects the transition process in the tank (the values t_0, t_1, t_2 are selected by the student independently on accordance with the general modeling time);
- graphs of transient processes should be reflected on the *Scope* block.

Fulfill modeling of object's controlling. Drag into the model window the block *Chart*. Open this block and develop a *SF*-diagram for model:

- place in the window three instances of the *State* object;
- in the first instance of that object to set up the initial values of the variables G_n, G_c, f_c ;
- connect the states by the objects *Transition*; the event that initiates the transition is the system time: after the time t_1 the flow of the liquid in the tank is changed; after time t_2 the area of flow section of the drain hole is changed;
- form a list of input (system time) and output (the values of the object variables which change during the modeling process) data of the block.

Copy subsystem of the object dynamic model to another place in the same window of the block-diagram. Connect the block *Chart* for the dynamic model of the tank.

The transition process is reflected on the *Scope* block.

Execute the simulation experiments of object's controlling on the developed

diagram.

Table 3.1 – Assignment options

№	G	ρ	μ	f	F
1	250	995	0,997	1,6	3
2	298	1000	1,004	2	3,6
3	268	1002	0,995	1	3,9
4	290	1002	1,002	1,3	3,6
5	230	1001	1,001	2	3,9
6	286	996	0,995	1,4	3,4
7	257	998	0,996	1,5	3,6
8	232	1001	1	1,3	3,7
№	G	ρ	μ	f	F
9	279	997	1,005	1,2	3,3
10	242	1000	1,002	2	3,7
11	272	995	0,999	1	3,8
12	294	999	0,995	1,2	3,4
13	224	999	0,995	2	3,5
14	230	996	1,001	1,7	3
15	228	1004	1,001	1,8	3
16	275	996	1,003	1,8	3,4
17	235	996	1,005	1,7	3,3
18	248	1005	1,002	1,9	3,8
19	294	1003	1	1,3	3,1
20	257	996	1	1	3,1

3.5 Content of report

The report on laboratory work must contain:

- work purpose and assignment for the laboratory work;
- the equations of static and dynamic models of the tank;
- the models block-diagrams with explanatory comments;
- the results of the experiments executed on these models;
- SF-chart describing the operation of the system at different time intervals;
- the graphics of simulation experiments;
- the conclusions of the work.

3.6 Control questions

3.6.1 What class is the worked out model belonged to?

3.6.2 What variables of the modeled object are the inputs of the model?

3.6.3 What variables of the modeled object are the outputs of the model?

3.6.4 Define the static and dynamic models.

3.6.5 What can transient process in the object occur?

3.6.6 What is the Stateflow done for?

3.6.7 What is the SF-diagram?

- 3.6.8 Explain the notion of "state" in the Stateflow.
 3.6.9 What are the transitions?
 3.6.10 How does the data create or change in the Stateflow?

4 Laboratory work №4. Linearization of non-linear equations of the object to determine the transfer functions

Work purpose: to study linearization procedure of the non-linear equation of the object in order to obtain transfer functions and to study the transient processes of the object

4.1 Assignment for the laboratory work

In the process of fulfilling the laboratory work the student must:

- derive the final model equation;
- linearize non-linear equations;
- obtain model transfer functions through channels;
- explore transients through various channels;
- analyze the results and make conclusions.

4.2 Description of the object

In a closed vessel under pressure P_3 , a fluid flow G_1 is supplied at a pressure P_1 . A stream G_2 with a pressure P_2 is discharged from the vessel. Supply and drain openings are located below the liquid level, which makes the variables G_1 and G_2 dependent on the liquid level in the vessel H

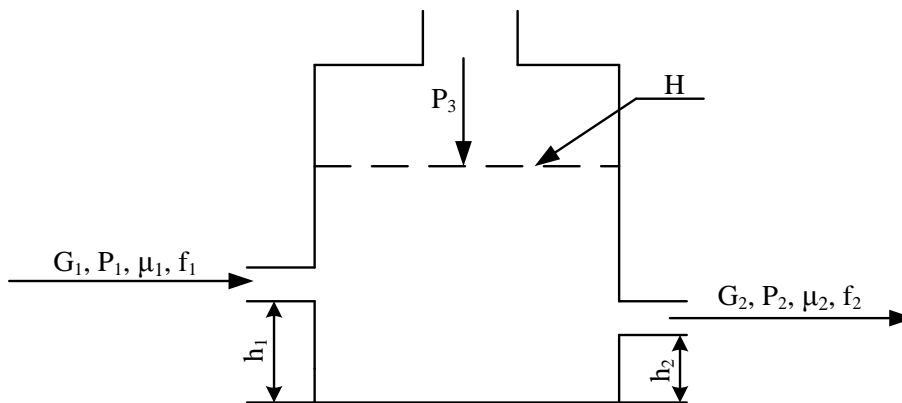


Figure 4.1 – Simplified object diagram

The quantities $\mu_1, \mu_2, f_1, f_2, F_0, \rho_0$ have the same meaning as in the laboratory work №3.

Let us use mass conservation equation (3.1):

$$F_0 \rho_0 \frac{dH}{dt} = G_2(t) - G_1(t). \quad (4.1)$$

For the conditions of the problem under consideration:

– equation of motion for the input stream:

$$G_1 = \mu_1 f_1 \cdot \rho \cdot \sqrt{2[P_1 - P_2 - \rho g(H - h_1)]}. \quad (4.2)$$

– equation of motion for the output stream:

$$G_2 = \mu_2 f_2 \cdot \rho \cdot \sqrt{2[P_3 - P_2 - \rho g(H - h_2)]}. \quad (4.3)$$

Considering the input variables P_1, P_2, P_3, f_1, f_2 , it is necessary to obtain a linear model of the process in the vessel. The initial equilibrium state of the system corresponds to the time t_0 .

To estimate the values of the state coordinates in the initial equilibrium state, we derive the static model. The state of equilibrium is achieved under the following condition:

$$\frac{dH}{dt} = 0 \rightarrow G_1(t) = G_2(t) \rightarrow G_1 = G_2 = \text{const.}$$

Then,

$$\mu_1 f_1 \sqrt{2\rho[P_1 - P_2 - \rho g(H - h_1)]} = \mu_2 f_2 \sqrt{2\rho[P_3 - P_2 - \rho g(H - h_2)]}.$$

Hence, the static model:

$$H = \frac{k_1^2 P_1 - (k_1^2 + k_2^2) P_2 + \rho g(k_1^2 h_1 + k_2^2 h_2)}{\rho g(k_1 + k_2)}, \quad (4.4)$$

where $k_1 = \mu_1 f_1$ and $k_2 = \mu_2 f_2$.

In order to linearize expressions (4.2) and (4.3), we expand them in a Taylor series and confine ourselves to linear terms:

$$\begin{aligned} G_1 \approx G_{10} &+ \left(\frac{\partial G_1}{\partial H}\right)_0 (H - H_0) + \left(\frac{\partial G_1}{\partial f_1}\right)_0 (f_1 - f_{10}) + \left(\frac{\partial G_1}{\partial P_1}\right)_0 (P_1 - P_{10}) \\ &+ \left(\frac{\partial G_1}{\partial P_2}\right)_0 (P_2 - P_{20}), \end{aligned}$$

$$\begin{aligned} G_2 \approx G_{20} &+ \left(\frac{\partial G_2}{\partial H}\right)_0 (H - H_0) + \left(\frac{\partial G_2}{\partial f_2}\right)_0 (f_2 - f_{20}) + \left(\frac{\partial G_2}{\partial P_2}\right)_0 (P_2 - P_{20}) \\ &+ \left(\frac{\partial G_2}{\partial P_3}\right)_0 (P_3 - P_{30}), \end{aligned}$$

where the index "0" denotes the value of the variables in the steady state. Let us find the coefficients of the expansion:

$$\left(\frac{\partial G_1}{\partial H}\right)_0 = -k_1^0 \frac{2\rho^2 g}{2\sqrt{2\rho[P_1^0 - P_2^0 - \rho g(H^0 - h_1)]}} = A_1;$$

$$\left(\frac{\partial G_1}{\partial P_1}\right)_0 = k_1^0 \frac{2\rho}{2\sqrt{2\rho[P_1^0 - P_2^0 - \rho g(H^0 - h_1)]}} = A_2;$$

$$\left(\frac{\partial G_1}{\partial P_2}\right)_0 = -k_1^0 \frac{2\rho}{2\sqrt{2\rho[P_1^0 - P_2^0 - \rho g(H^0 - h_1)]}} = A_3;$$

$$\left(\frac{\partial G_1}{\partial f_1}\right)_0 = \mu_1 \sqrt{2\rho[P_1^0 - P_2^0 - \rho g(H^0 - h_1)]} = A_4.$$

The coefficients for G_2 are determined in a similar way. Then:

$$G_1 = G_{10} + A_1 \times \Delta H + A_2 \times \Delta P_1 + A_3 \times \Delta P_2 + A_4 \times \Delta f_1; \quad (4.5)$$

$$G_2 = G_{20} + B_1 \times \Delta H + B_2 \times \Delta P_2 + B_3 \times \Delta P_3 + B_4 \times \Delta f_2. \quad (4.6)$$

Finally, equation (4.1) takes the form:

$$F_0 \rho_0 \frac{d\Delta H}{dt} = \Delta G_1(t) - \Delta G_2(t) + (A_1 - B_1)\Delta H + A_2 \cdot \Delta P_1 + (A_3 - B_2) \cdot \Delta P_2 - B_3 \cdot \Delta P_3 + A_4 \cdot \Delta f_1 - B_4 \cdot \Delta f_2.$$

We use the following notation:

$$\Delta H \rightarrow y, \Delta G_1 \rightarrow x_1, \Delta G_2 \rightarrow x_2, \Delta P_1 \rightarrow x_3, \Delta P_2 \rightarrow x_4, \Delta P_3 \rightarrow x_5, \\ \Delta f_1 \rightarrow x_6, \Delta f_2 \rightarrow x_7,$$

then:

$$F_0 \rho_0 \frac{dy}{dt} - (A_1 - B_1)y = x_1 - x_2 + A_2 \cdot x_3 + (A_3 - B_2) \cdot x_4 - B_3 \cdot x_5 + \\ + A_4 \cdot x_6 - B_4 \cdot x_7.$$

To proceed to the transfer function, it is necessary to use the Laplace operator $\frac{d}{dt} \rightarrow s$, denoting $F_0 \rho_0 = T$, we obtain:

$$TsY(s) - (A_1 - B_1)Y(s) = X_1(s) - X_2(s) + A_2 \cdot X_3(s) + (A_3 - B_2) \cdot X_4(s) - \\ - B_3 \cdot X_5(s) + A_4 \cdot X_6(s) - B_4 \cdot X_7(s)$$

From this algebraic equation we obtain transfer functions along various channels:

$$\begin{aligned}
 W_{x_1 \rightarrow y} &= \frac{1}{TS - (A_1 - B_1)}; \\
 W_{x_2 \rightarrow y} &= \frac{-1}{TS - (A_1 - B_1)}; \\
 W_{x_3 \rightarrow y} &= \frac{A_2}{TS - (A_1 - B_1)}; \\
 W_{x_4 \rightarrow y} &= \frac{(A_3 - B_2)}{TS - (A_1 - B_1)}; \\
 W_{x_5 \rightarrow y} &= \frac{-B_3}{TS - (A_1 - B_1)}; \\
 W_{x_6 \rightarrow y} &= \frac{A_4}{TS - (A_1 - B_1)}; \\
 W_{x_7 \rightarrow y} &= \frac{-B_4}{TS - (A_1 - B_1)}.
 \end{aligned}$$

4.3 Block of the transfer function with variable coefficients

In the laboratory work, transfer functions are obtained on seven channels, which differ only in numerical values of the parameters. To build a block diagram of the model, let's use the advanced features of the Simulink.

In the standard library Simulink there is no block transfer function, for which it would be possible to set the numerator and denominator coefficients in the form of input vectors. The presence of such a block would allow modeling objects whose properties change with time. To solve the problem, it is necessary to create a block of model of the transfer function, which implements that kind of algorithm [5].

Figure 4.2 shows the subsystem *Transfer Fcn (vp)* with the following elements:

- *S-Function* block that implements the algorithm for setting parameters of the transfer function model from the outside;
- it is necessary to set an input signal (for example, a step signal) to input port of *In*;
- the output signal is taken from the port *Out*;
- the vector of the numerator coefficients should be set at the input *Num*, and the vector of the denominator coefficients should be set at the *Den* input.

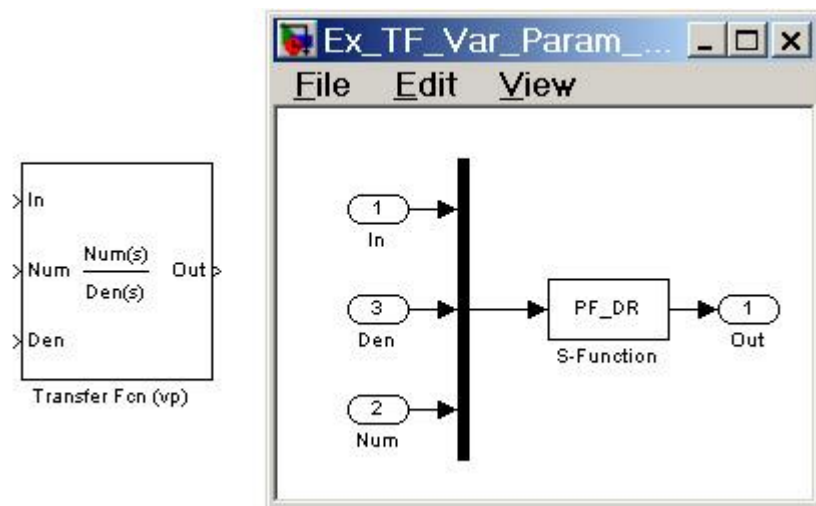


Figure 4.2 – Transfer *Fcn* (*vp*)

In the window for specifying the parameters of the transfer function, you must specify the dimension of the vectors of the numerator and denominator coefficients. Block parameters:

- *Numerator length* [The dimension of the vector of the coefficients of the numerator]. The dimension of the vector of the numerator's coefficients should not exceed the dimension of the vector of the denominator coefficients.
- *Denominator length* [Dimension of the vector of the denominator coefficients].

4.4 The order of the fulfilling laboratory work

Build the block diagrams of the static model of the object in Simulink (the parameter values and the initial values of the object variables are given in table 4.1). Model output should be sent to the Display unit.

Obtain the coefficients B_1, B_2, B_3, B_4 for equation (4.6), differentiating equation (4.3).

Pre-download the transfer function block with variable coefficients from the website [5].

Build 7 block diagrams of transfer functions on different channels, using a transfer function block with variable coefficients, instead of the standard *Transfer Fcn*.

It is necessary to send to block's input:

- to port *In* – step signal *Step*;
- to the input *Num* – the vector of the parameters of the numerator of the transfer function;
- to the input *Den* - vector of the denominator coefficients of the transfer function.

The output of the block must be transferred to *Scope*.

For each block, conduct a transient analysis and determine the following quality characteristics:

- maximum of the transient process;
- transient time; buildup time (from 10% to 90% of the steady-state value);
- steady state value.

Table 4.1 – Assignment options

№	μ_1	μ_2	f_1	f_2	ρ	P_1	P_2	P_3
1	0,27	0,94	0,6	0,4	996	4,3	2,1	2,6
2	0,06	0,42	0,4	0,4	1001	4,2	2,2	2,6
3	0,61	0,93	0,6	0,5	1001	4,9	1,7	3,4
4	0,16	0,97	0,3	0,3	995	4	1,8	3,5
5	0,75	0,43	0,3	0,3	997	4,3	2,2	3,4
6	0,7	0,29	0,5	0,3	998	4,4	2,4	3,3
7	0,12	0,43	0,6	0,3	997	4,5	1,7	3,3
8	0,75	0,23	0,6	0,3	1000	4,6	2,3	2,5
№	μ_1	μ_2	f_1	f_2	ρ	P_1	P_2	P_3
9	0,52	0,11	0,5	0,4	997	5	1,9	3
10	0,64	0,45	0,3	0,4	1001	4,2	2,9	3,2
11	0,89	0,72	0,6	0,5	996	4,5	2,3	3,1
12	0,6	0,82	0,4	0,3	995	5	2,8	2,8
13	0,65	0,73	0,6	0,6	999	4,5	2,5	2,8
14	0,39	0,03	0,6	0,3	996	4,5	1,7	2,6
15	0,04	0,73	0,3	0,3	997	4,8	1,9	3,4
16	0,06	0,39	0,4	0,6	998	4,5	2,9	2,8
17	0,28	0,79	0,5	0,4	996	4	1,7	3
18	0,03	0,33	0,5	0,6	996	4,2	2,9	2,7
19	0,28	0,59	0,6	0,3	998	4,6	3	3,4
20	0,32	0,98	0,6	0,3	1002	4,7	2,1	2,8

4.5 Content of report

The report on laboratory work must contain:

- work purpose and assignment for the laboratory work;
- linearization procedures for the model;
- equations of static and dynamic reservoir models;
- block diagrams of the model with explanatory comments;
- results of experiments conducted to these models;
- graphs of simulation experiments;
- conclusions on work.

4.6 Control questions

4.6.1 What is the class of the developed model?

4.6.2 What variables of the modeled object are model inputs?

4.6.3 What variables of the modeled object are model outputs?

- 4.6.4 Procedure for linearizing a dynamic object.
- 4.6.5 Taylor series expansion.
- 4.6.6 How does transients through different channels get?
- 4.6.7 What can cause a transient process in the object under consideration?
- 4.6.8 Transient parameters.
- 4.6.9 Block of the transfer function with variable coefficients.
- 4.6.10 What does "Numerator length" and "Denominator length" block's parameters mean?

5 Laboratory work №5. Direct methods of identification of linear objects of the first order

Work purpose: to learn an identification method of linear dynamic objects using the transient function.

5.1 Assignment for the laboratory work

In the process of fulfilling the laboratory work the student must:

- study the graphical methods of the first order linear objects identification [1,2];
- collect a block-diagram to identify the object via the transient function;
- determine parameters of the object transfer function by graphical method of identification;
- evaluate the accuracy of identification.

5.2 Identification method using the transition function

The direct identification methods require special input signals: step signals for the identification by step response, pulse signals for the identification by impulse response and sinusoidal input signals with different frequencies to determine frequency characteristics.

In many cases, to determine the transfer function of the system you can use the recording of its transition function. This method is applicable to most types of linear systems (1st and 2nd orders and a periodic system of higher order). The most correctly the identification method using the transition functions is applied to the first order processes.

There is discussed the object of the first order. The model equation for the first order object has the form:

$$T \frac{dy(t)}{dt} + y(t) = kx(t) \text{ or } W(p) = \frac{k}{Tp + 1}. \quad (5.1)$$

The experiments are executed on the object: at time $t_0=0$ the input value x changes spasmodically for the amount a and the graphic of the transition process is

written. Using this graphic we must to determine the parameters T and k of the transfer function.

The general expression of the differential equation (5.1) solution has the form:

$$y(t) = k \cdot a \cdot \left(1 - e^{-\frac{t}{T}} \right). \quad (5.2)$$

Function (5.2) describes the transition process graphic. For simplicity, let us assume that $a = 1$.

The determination of the transfer function parameters can be implemented by the following ways:

a) from the expression (5.2) at $t \rightarrow \infty$ we have $y(t) = k \cdot a$. On the other side at $t \rightarrow \infty$ we have the steady value of the output signal, i.e. through the ordinate of the object steady state (on the graphic) you can define the parameter k .

When $t = T$ we have the function value $y(t) = k \cdot (1 - e^{-1}) = k \cdot (1 - 0.37) = 0.63 \cdot k$. It means that the time constant T of the first order system equals to the interval of time for which the transition function reaches 63 % of its steady-state value; the coefficient k is determined, as described in a).

For determining the parameter T we differentiate the expression (5.2):

$$y' = \frac{dy(t)}{dt} = k \cdot a \cdot \frac{1}{T} \cdot e^{-\frac{t}{T}}.$$

In the initial point

$$\frac{dy(0)}{dt} = k \cdot a \cdot \frac{1}{T},$$

so, we have (from the geometric meaning of the derivative):

$$\operatorname{tg} \alpha = k \cdot a \cdot \frac{1}{T},$$

where α is the angle of the tangent inclination to the function graphic at $t = 0$.

On the other hand, the tangent line, the perpendicular line (from the point of intersection of the tangent line and the steady state line) on the x-axis, and the x-axis create a rectangular triangle, where $\operatorname{tg} \alpha$ is the ratio of the opposite cathetus to the adjacent cathetus, that is the triangle adjacent cathetus will be T value.

So, the value $T = \frac{k \cdot a}{\operatorname{tg} \alpha}$ is section of the x-axis that is equal to the distance from the coordinates origin to the point at which the tangent line intersects the asymptote.

5.3 The order of fulfilling the laboratory work

Download from the folder *Work/Objects5* your "object" (by option) in Simulink.

Use the step signal as the input signal, the output signal is reflected in the *Scope*.

Using the transition process graphic received in *Scope*, identify by two ways (a and b) the first order transfer function parameters. The result will be two models in the form of transfer functions (

Figure 5.1). Write down expressions for these models.

On the restored models, giving the same step signal to their inputs, you will get the output signals.

Compare the graphics of the "object" and models transition functions. Which model gives more accurate result? What is the reason?

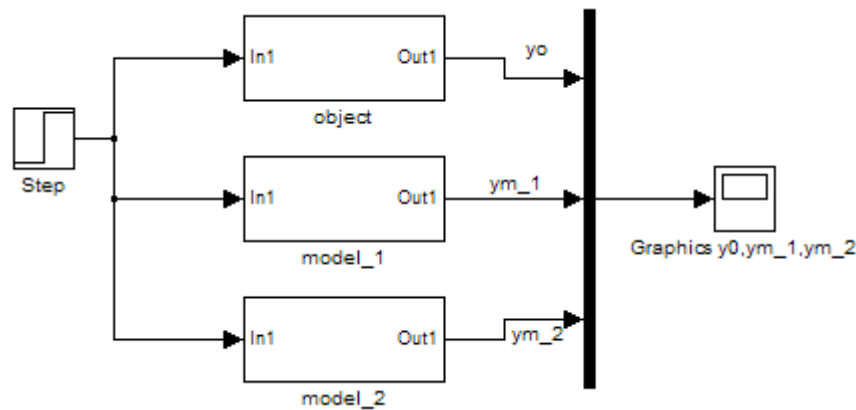


Figure 5.1- Block- diagram of comparing of the object and models outputs

5.4 Content of report

The report on laboratory work must contain:

- work purpose and assignment to laboratory work;
- *Scope* window with the transition process, on which tangent to the graph of the transition function, 63% mark and steady state are marked;
- the values of the transfer functions parameters that are defined by both ways;
- block-diagrams of the object and both models (similar to figure 5.1);
- the graphics of the object and both models transient processes in one *Scope*;
- comparative analysis of graphics;
- conclusions.

5.5 Assignment options

Selected from Work/Objects5 of Simulink.

5.6 Control questions

5.6.1 What is identification?

5.6.2 What methods are called the direct methods of identification?

5.6.3 What function is used for graphical identification of linear object?

5.6.4 What signal at the object input was used for the identification procedure using graphical method?

5.6.5 How is the parameter k of the transfer function defined?

5.6.6 Please explain the methods of determining the T parameter of the transfer function.

5.6.7 How was the tangent to the transition process graphic in the *Scope* window built?

5.6.8 What block is used to reflect all the graphics in one *Scope* window?

5.6.9 By what methods of identification a more accurate result is given? Why?

5.6.10 For what objects can be applied the considered identification method?

6 Laboratory work №6. The identification of linear object using the frequency characteristics

Work purpose: the learning of methods of identification using harmonic signals.

6.1 Assignment for the laboratory work

In the process of carrying out the laboratory work the student must:

- learn the frequency methods of linear object identification [**Ошибка! Закладка не определена., Ошибка! Закладка не определена., 1**];
- assemble a block-diagram for object identification;
- execute the experiments;
- determine the values of amplitudes and phase shifts of the output signal;
- familiarize with the program of experiments results processing;
- using this program, determine of the object frequency response;
- determine parameters of the object transfer function;
- evaluate the identification accuracy.

6.2 Determining of the transfer function coefficients using the frequency characteristics

Frequency characteristics are widely used for the analysis and synthesis of

control systems. They also give the possibility of determining the object's model equations.

As it is known, if to the object input is supplied the sinusoidal impact:

$$x(t) = A_0 \cdot \sin(\omega \cdot t),$$

then the measured steady-state value of the output signal will be

$$y(t) = A_1 \cdot \sin[\omega \cdot t + \varphi(t)] + n(t),$$

where $n(t)$ is the measurement error'

$$A_1/A_0 = |W(j\omega)|;$$

$$\varphi = \text{Arg}[W(j\omega)].$$

The frequency characteristics of $W(j\omega)$ is determined by supplying the sinusoidal input signals $A_0 \sin(\omega t)$ at different frequencies ω and recording the corresponding output signals $A_1 \sin[\omega t + \varphi]$. For the purpose of obtaining the required frequency response, the values of A_1/A_0 and φ are determined for each of the considered frequency ω . That is determined the amplitudes ratio at the frequency ω_i , and get $W(j\omega)$ by recording of inputs and outputs signals. The phase shift $\varphi(\omega_i)$ is received from the comparison of the maxima positions of the curves $x(t)$ and $y(t)$.

The received frequency characteristics give the possibility to determine the object model equation. The method is the following. For each frequency the real and imaginary frequency characteristics are determined:

$$P(\omega_i) = A(\omega_i) \cdot \cos[\varphi(\omega_i)];$$

$$Q(\omega_i) = A(\omega_i) \cdot \sin[\varphi(\omega_i)].$$

This data is used to determine the coefficients $(a_1, a_2, \dots, a_{n-1}, b_0, b_1, \dots, b_m)$ of the transfer function:

$$W(p) = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_0}{p^n + a_{n-1} p^{n-1} + a_{n-2} p^{n-2} + \dots + a_1}.$$

To calculate the coefficients of the transfer function, we replace p for $j\omega$, and write the function as a sum of real and imaginary parts $P(\omega_i) + j \cdot Q(\omega)$, then get an expression valid for any values ω . From this complex expression by equating coefficients of the real and imaginary parts, we get the system of two equations. Substituting different values ω_i (so many meanings, that the system would be square) in these equations, we obtain the system to determine unknown coefficients.

To clarify the obtained values of the coefficients, we repeat the calculations for other frequencies and take the average of the two calculations.

If the true order of the object is above estimated, the values of the corresponding coefficients will be very different.

6.2 Program of experiments results processing

To determine the coefficients of the transfer function it is necessary to solve the large order system of equations. With this purpose the program developed in Delphi - *Lab5* is used in the laboratory work. The main window is shown at figure 6.1.

The work in this window is fulfilled as follows:

- enter the values of frequencies ω_i and define for each frequency the values of the amplitudes and phase shifts;
- click "Calculate P(j ω) and Q(j ω)", the table will have two rows with calculated values of real and imaginary characteristics;
- set the order of the object by switch;
- from the drop-down list select frequencies;
- click on the "Solve the system" button;
- the values of the calculated coefficients of the object's transfer function will appear in the right side of the lower window.

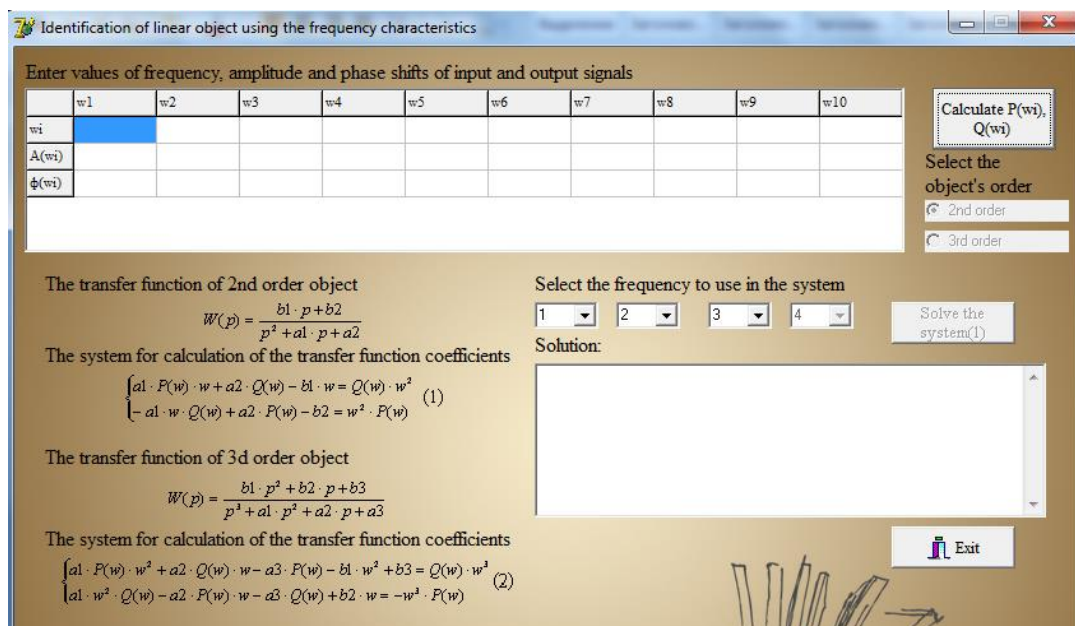


Figure 6.1– The main window of the experiments results processing program

6.3 Assignment for the laboratory work

From the folder *Work/Objects6* load your "object" in Simulink (by option).

Apply the sinusoidal signal with the known amplitude to the "object" input, set *Scope* block at the "object" output.

Fill in the table 6.1: changing the values of the input signal frequencies,

record the corresponding output signals. Determine the amplitude and phase characteristics of the "object" for each frequency.

Load program *Lab6*.

Transfer the data from table 1 to the program window. Calculate $P(j\omega)$ and $Q(j\omega)$.

Select the order of the object – 2, choose the frequency values, and calculate the system (1).

To clarify the obtained coefficients values, repeat the calculation for other frequencies (the order of the object – 2) and calculate the average of the two calculations.

Select the order of object – 3, choose the frequency values, and calculate system (2).

To clarify the obtained coefficients values, repeat the calculation for other frequencies (the order of object – 3) and calculate the average of the two calculations.

Select the most appropriate order of the object and write down the expression for the recovered transfer function.

Create a model's block-diagram of the object with the obtained transfer function in Simulink. Apply to the input of the model the same signals as on "object" and compare the responses of the object and the model (in the same *Scope* window).

Table 6.1 – Signal processing

ω_i	ω_1	ω_2	...	ω_{10}
$A(\omega_i)$				
$\varphi(\omega_i)$				

6.4 Content of report

The report on laboratory work must contain:

- work purpose and assignment to laboratory work;
- the block-diagram for experiments
- the graphics obtained experimentally at the object;
- the filled-in table 6.1;
- the system of equations for determining the transfer function coefficients
- the values of model coefficients for different orders of the object;
- the block-diagram for comparing the outputs of the object and the model;
- the comparative results of simulation experiments on the model;
- the conclusions of the work.

6.5 Assignment options

Assignment options are selected from folder *Work/Objects6* of Simulink.

6.6 Control questions

- 6.7.1 What are direct methods of identification?
- 6.7.2 What signals in addition to harmonic, can be used in direct methods?
- 6.7.3 What objects can be identified by the frequency method?
- 6.7.4 Explain of identification frequency method.
- 6.7.5 How is the model order selected?
- 6.7.6 How is the phase shift of the output signal determined?
- 6.7.7 How was the amplitude of the output signal defined?
- 6.7.8 How is the system of equations for determining the transfer function coefficients received?
- 6.7.9 How are the values of the model coefficients specified?
- 6.7.10 How accurately the model is defined, what is the reason for the difference?

7 Laboratory work №7. Parametric identification of linear dynamic objects

Work purpose: mastering methods of parametric identification of objects, the approximation of functions and solution problem of residual functional minimization.

7.1 Assignment for the laboratory work

- In the process of fulfilling the laboratory work the student must:
- learn the methods of linear dynamic objects parametric identification [2,3];
 - study the procedure of minimizing the least square criterion [**Ошибка! Закладка не определена.**, 1];
 - conduct the experiments on the "object": to the input object give the specified input signal, register a numerical value of the output signal;
 - using the Curve Fitting Toolbox, to fulfill the approximation of the output signal by the chosen equation;
 - determine the structure of a linear dynamic object's model;
 - write down the residual functional for the model of the selected structure;
 - solve the problem of minimizing this functional using the results of the experiments; determine the model parameters;
 - evaluate the accuracy of the obtained model.

7.2 Parametric identification of dynamic objects

In the parametric case the model is defined by a set of parameters that must be evaluated in the identification process. The problem of parametrical identification can be formulated as follows: based on experimental data to find values of the vector of parameters for which the model best (or accurate enough) in a sense approximates the object operator.

The model creation starts from the selection of the model structure that defines the relationship of the observed data through a set of parameters. Then the input actions are formed and supplied to the object, the responses to these impacts (output signals) are measured. Then input and output signals and the selected structure are used to estimate the parameter values in accordance to the adopted quality criterion. Least square criterion is often used, according to which the parameters that minimize least square criterion are estimated; least square criterion is the square of the difference model output signals and object output signals for the same input influence.

In the one-dimensional case, the relation between the input $x = x(t)$ and output $y = y(t)$ is an ordinary differential equation:

$$\sum_{i=0}^p a_i \cdot y^{(i)} = \sum_{j=0}^l b_j \cdot x^{(j)}, \quad (7.1)$$

where

$$y^{(i)} = \frac{d^i y}{dt^i}, \quad x^{(j)} = \frac{d^j x}{dt^j}, \quad a_p = 1, \quad l \leq p$$

with the initial conditions for $\frac{d^i y(0)}{dt^i}, i = 0, 1, \dots, n-1$.

The model is determined by $(p+l+1)$ parameters $c = (a_0, \dots, a_{p-1}, b_0, \dots, b_l)$.

The initial information for building the identification procedure is an identifying models (7.1) and observation:

$$\{x_t, y_t\} \text{ in the interval } [0, T]. \quad (7.2)$$

The residual function is constructed as the mean square of the difference of the right and left side of equation (7.1) when the functions X_t, Y_t (the observation of the object) are substituted:

$$Q(c) = \int_0^T \left[\sum_{i=0}^p a_i \cdot y_t^i - \sum_{j=0}^l b_j \cdot x_t^j \right]^2 dt. \quad (7.3)$$

The problem of minimizing the functional (7.3) is solved by the usual methods of finding a smooth function's minimum. Equating to zero the derivatives of (7.3) for all unknown parameters, we get the system of linear equations, whose solution is a solution of a minimization problem. The resulting system can be solved by standard computational methods. However, for calculating the coefficients of the system it is necessary to know the derivatives of the input x_t and output y_t signals of the object. If these signals are given in analytic form, there is no problem in determining the derivatives of any order. In the laboratory work inputs are given as some functions of time. The values of output signals during the experiment are obtained in the form of a data array. That is why it is necessary to use the decomposition in a series by the system of functions given in the analytical form. The approximation of function will be executed using the *Curve Fitting Toolbox*.

7.3 Curve Fitting Toolbox

The work with this toolbox must start from the download of data. The data can be generated in the Command Window of MatLab in the respective vectors.

The main window of the Curve Fitting Tool opens by the command *cftool* (in Command Window of MatLab). Initially it has an empty graphics window and 4 buttons above it: *Data...* – download of data; *Fitting...* – fulfilling of fit; *Plotting...* - graphical visualization of fit; *Analysis...* - analysis of the results of the fit.

Import of data. Usually work with the *Curve Fitting Tool* starts with specification of the type of data being downloaded, which is necessary to activate the *Data* button.... This leads to the appearance of the window *DataImport*. In this window, using the appropriate tabs can be set the type of data sets, to look through, to adjust and smooth them. Clicking the *Apply* button and closing the window causes you load of the adjusted data in the main Curve Fitting Tool window. As a result the data points appear in the field of graphics of this window.

The implementation of the specified type fit. The selection of buttons *Fitting/New Fit* allows you to start the fit procedure.

The opening list *Type of fit* allows you to set any of the following types of fitting:

- 1) Custom equations - the regression function specified by the user.
- 2) Exponential - exponential regression.
- 3) Fourier - approximation by Fourier series.
- 4) Gaussian - approximation by Gaussian curve.
- 5) Interpolant - with the choice of method.
- 6) Polynomial - polynomial regression.
- 7) Power - polynomial regression.
- 8) Rational - the regression by rational function with the choice of its numerator and denominator.
- 9) Smoothing Spline -the regression by smoothing splines;
- 10) Sum of Sin Func - regression by the sum of sinusoidal functions;
- 11) Weibull - regression by Weibull curve.

There is also a possibility to define the own regression function, of course, on the basis of built-in functions of MatLab.

It can be specified of number of fit options if you click the *Fit Options* button... of *Fitting* window. In the options of fitting window it is possible to set of different fitting algorithms, the range of derivatives changing, maximum value of function and the initial values of regression parameters. All of these values significantly influence on the convergence and fit speed.

Graphical visualization of the regression. The window of graphics settings is opened by the clicking *Plotting...* button window of Curve Fitting Tool.

The analysis of the regression results. An important step in the approximation is the analysis of the results. For this purpose you should to activate the *Analysis...* button of *Curve Fitting Tool* window. The analysis window *Analysis* is opened. In the left part of the window there are a number of analysis options: the calculations, the derivatives graphics and integral for the regression function display, etc. The needed options are specified by ticking the sign their names. After that, clicking the *Apply* button the analysis results table is formed at the top of the window and a window with graphical representation of analysis results is displayed.

The main criterion for selecting the regression curve is the minimum error *Residuals* in a specified range of argument changing; the absence of rough errors of the curve at the edges of this range or in its separate parts, etc. It is often important to match the regression curve to the possible law of the source data points location. We recommend that you test multiple regression curves for a given set of points.

7.4 The order of fulfilling the laboratory work

Realize the "test bench" for conducting the identification procedure:

- download "object" block-diagram from the folder *Work/Objects7* in Simulink (by option);
- define input signal (the options are given in table 7.1) and apply it to the object input; receive the output signals;
- the values of the output signal pass to the Workspace of Matlab; for this purpose output signal is set *To Workspace* block, where specify the name of the corresponding variable.

Download the Curve Fitting Tool, transfer data about the output signal in its window; by the form of the location of the signal points on the graph select the structure of the approximating polynomial.

Fulfill the approximation of measured values of the output signal by various functions: various orders polynomials or some selected equations.

To specify your regression equation in the list of types of the regression it is necessary to select the item *Custom equations*. This leads to the appearance of the window *Create Custom Equations*. This window has two tabs. On the first tab of *Linear Equations*, you can specify the parameters of the equations, linear with respect to the regression coefficients. Note that the dependence can be nonlinear at

this. On the other tab, *General Equations* can be set arbitrary non-linear regression equation, i.e. to carry out nonlinear regression.

Completing the adjustment, select the function that best approximates the output signal. Write down the expression for this function (with calculated coefficients).

Define the structure of the linear model (write an equation).

Write the residual functional: instead of the input function and its derivatives substitute the expression for the input signal and its derivatives; instead of output function and its derivatives substitute the received in the result of approximation the expression and its corresponding derivatives.

Equating to zero the derivatives of the residual functional with respect to all unknown parameters of the model write a system of equations to determine the coefficients of this model; modify the resulting system to a standard form of a system of linear algebraic equations.

Solve the resulting system of linear algebraic equations using Matlab procedure (in the Command Window).

For the systems of standard form:

$$Ax = b,$$

the solution is found by the following way:

$$x = \text{inv}(A) \cdot b,$$

where A is the matrix of the coefficients of a linear algebraic equations system;

b is the vector of the right side of the system;

x is the vector of unknowns (these are the unknown parameters of the model).

Preliminary the massive A (matrix) and b (a column vector) are entered in the Command Window of Matlab, and also check the condition of the solution existence: $\det(A) \neq 0$.

Using the received parameters values, write the desired equation of the dynamic model.

To the diagram that is implemented in section 7.4.1, add the block to the model; at its input apply the same input signal; the output of the "object" and the model are reflected in the same viewing window. Compare the results.

Make the conclusions on work.

7.5 Content of report

The report on laboratory work must contain:

- work purpose and assignment for laboratory work;
- block diagram of the "measurement equipment";
- the results of experiments on the object in the Workspace of Matlab;

- the sequence of fulfilling the procedure of approximation of the output signal;
- the selected approximating polynomial;
- the residual functional for the determination of the model parameters;
- the condition of this functional minimization;
- the system of linear algebraic equations to determine the parameters of the model;
- the solution of this system of equations;
- the received equation of the model;
- the comparative graphics of the object and the model outputs;
- the conclusions of the work.

7.6 Assignment options

Assignment options of the "objects" are selected from the folder Work/Objects6 of Simulink.

The structural model parameters and input signals are selected from the table.

Table 7.1 – Assignment options

№	Expression for input signal	Structural parameters of the model	№	Expression for input signal	Structural parameters of the model
1	$x(t) = t+1$	$p=1, l=0$	11	$x(t) = t+3$	$p=1, l=0$
2	$x(t) = 3t+2$	$p=2, l=1$	12	$x(t) = 5t+2$	$p=2, l=1$
№	Expression for input signal	Structural parameters of the model	№	Expression for input signal	Structural parameters of the model
3	$x(t) = t^2+t+1$	$p=3, l=1$	13	$x(t) = t^2+2t+1$	$p=3, l=1$
4	$x(t) = t-7$	$p=2, l=1$	14	$x(t) = t-5$	$p=2, l=1$
5	$x(t) = 2t^2-6$	$p=3, l=2$	15	$x(t) = 3t^2-6$	$p=3, l=2$
6	$x(t) = t$	$p=2, l=1$	16	$x(t) = 2t$	$p=2, l=1$
7	$x(t) = 2t+4$	$p=1, l=0$	17	$x(t) = 3t+4$	$p=1, l=0$
8	$x(t) = t^2-t+5$	$p=3, l=1$	18	$x(t) = t^2-4t+5$	$p=3, l=1$
9	$x(t) = 4t+3$	$p=2, l=1$	19	$x(t) = 3t+3$	$p=2, l=1$
10	$x(t) = t^2+2$	$p=3, l=2$	20	$x(t) = t^2+5$	$p=3, l=2$

7.7 Control questions

7.7.1 What is the parametric identification?

7.7.2 What is the parameter estimation?

7.7.3 What criterion is used for estimation of the model parameters?

7.7.4 How can be received in the Workspace of Matlab the values of the output signal formed in Simulink?

7.7.5 Give the definition of functions approximation.

7.7.6 What criterion is used for the functions approximation?

- 7.7.7 What Matlab tool was used for the approximation of functions?
 7.7.8 What is the residual functional?
 7.7.9 How to solve the task of minimization of the residual functional?
 7.7.10 What is the difference between the outputs of the object and the model?

8 Laboratory work №8. Identification of linear objects using the System Identification Toolbox

Work purpose: to master the methods of objects identification using the System Identification ToolBox.

8.1 Assignment for the laboratory work

In the process of fulfilling the laboratory work the student must:

- get familiar with the methods of parametric identification of linear dynamic objects [2,3,4];
- fulfill experiments on the object;
- fulfill the preliminary processing of the experimental data;
- identify of the correlation model;
- evaluate the frequency characteristics of the object;
- for the parametric model to select the structure of the model;
- define the model parameters using the results of experiments and selected structure;
- to evaluate the accuracy of the models.

8.2 The System Identification Toolbox

The System Identification Toolbox of Matlab contains the means for creating the mathematical models of linear dynamic systems based on observed input and output data. In the laboratory work for the identification of linear systems, the graphic interactive interface *ident* is used.

To download the graphical identification tool in Command Window you need to type *ident* on prompt. As a result main window of the program is appeared.

The identification of the object is fulfilled by the following way:

- select the drop-down menu *Data-Import...* to download the experiment data; in the appeared dialog box enter the names of input and output variables of the model; also you can enter the time interval, the text of the comment. Then click the *Import* button and the data appears in the icon. Close the data import window. Please note that data will also appear in the *Working Data* and *Validation Databoxs*;
- the box *Time Plot* allows to reflect a window with inputs and outputs graphs. This window has a control menu. You can highlight with the mouse the rectangular area on the graph to enlarge the chart fragment, clicking the left mouse button you can increase/decrease the zoom, double click returns to the initial scale.

To clear the graphics window, click the icon with the data. Further, in order to work with the new data, you must move them with the mouse in the Working Data area.

Double click on the data icon calls up another window where you can change the name of the data array;

- using the *Select Range* of the *Preprocess* drop-down menu you select a subset from the dataset. This can be done using editable text field of the popup window, or by highlighting with the mouse a rectangular area on the signals graphs. Pressing *Insert* you can place the selected data in the new icon.

For verification of the model typically data that were not used for parameter identification is used. You can divide the initial array of the signals values into two halves, one of which is used for identification, and the second - to confirm the model. In real work, especially when dealing with complex objects, for verification several different data sets are used.

To identify the selected data is transferred to the *Working Data* area, for verification of the model the relevant data is transferred in the *Validation Data* area (by dragging with the mouse);

- to assess the model of the selected type in the *Estimate* drop-down menu the corresponding row is selected; in appearing (in accordance to the selected model type) dialog boxes it is necessary to set the required parameters by finding the evaluation model;

- when performing operations of the previous point in the main window of the interface (on the right side – *Model Views*) you should see icons with the names of the models.

Model verification: to verify how well the built model describes the object, it is necessary to analyze its behavior. To analyze and validate the model, *ident* program provides the user with a wide set of tools. These procedures are performed using the check boxes are located at the bottom right of the panel.

For comparison of the observed and simulated output signals, select “*Model output*” box. In the window the graphs of measured (black line) and calculated by the model outputs are represented. For consideration of separate fragments, hold down the left mouse button on the chart to highlight a rectangular region that will be displayed on the entire graphic window.

8.3 The order of fulfilling the laboratory work

Implement a "test bench" for fulfilling the identification procedure:

– load in the Simulink the "object" from the folder Work/Objects8 (by the variant);

– set a signal $x=a$ at the object input (the value a is chosen arbitrarily); get the output signals;

– the results of the experiment, i.e. the measured values of input and output signals, pass to the Workspace using the blocks *To Workspace*, in which you specify the names of the corresponding variables.

Type a command *ident* in the Command Window. A dialog box of the

interface appears.

Load data of the experiment from the Workspace:

- select the *Data- Import* command from the drop-down list;
- enter the names of input and output signals in the window that appears, the start time is 0, the sampling interval is 0.08; click *Import*, and then close the data import window;

- explore the original data by checking the *Time Plot* box.

Fulfill the preliminary processing of the signals of the investigated object, excluding of constant component:

- *Preprocess-Remove Means*; in the *Data Views* information about new data will display;

- again activate the *Time Plot* window and look through signals without constant component;

- select *Preprocess-Select Range* and specify the range of data in the appeared window; new data will be reflected in the interface window;

- drag-and-drop these new data into the *Working Data* area.

Repeat the operation range of the diapason in the aim of forming the data for checking up the model (for example, from the remaining part of the initial diapason), these data is also reflected in the interface window; drag them with the mouse to the *Validation Data* area.

Estimate the correlation model:

- in the *Estimate* drop-down list select *Correlation Model*;

- in the window that appears you can define the data filter (don't change anything), click *Estimate*;

- in the *Model Views* window interface icon *imp* will appear, i.e. the selected model is constructed;

- set the *Transient Response* box, and a graph of the transition function which was found by the correlation method will appear in a separate window;

- using the menu *Options*, instead of the transition function graph you can plot the impulse response.

Evaluate the frequency characteristics of the object:

- select *Estimate-Spectral Model*;

- repeat the process of the preceding paragraph;

- in the *Model Views* window interface icon *spad* appears representing the model in the form of frequency characteristics, which was found by a spectral method;

- reflect the result by setting the *Frequency Response* box.

Estimate the parametric model:

- select *Estimate-Parametric model*. There appears the window of the model structure definition (box *Structure*); by default, a model type ARX is supposed (you can select another model from this drop-down list) with parameters $na = 4$, $nb = 4$, $nk = 1$. The parameters na and nb determine the order of the right and left sides of the regression equation. The parameter nk refers to the presence in

the subject of lag with the magnitude $nk \cdot T = 1 \cdot 0,08 c = 0,08 c$. To change the parameters the values of the constants (the *Order Editor* button) type;

– use the supposed parameters of the model and click on the button *Estimate*. You will see an icon with a model named *arx441*;

– repeat the steps for the estimation of the parametric models ARX models with parameters $na = 2, nb = 2, nk = 3$. $nk=3$ means the delay $nk \cdot T = 3 \cdot 0,08 c = 0,24 c$. There will be one more icon of model - *arx223*;

For comparison of the received models click the *Transient Response* box; it will reflect the graphs of transient processes corresponding to the three built models.

Comparison of parametric models can be performed by selecting the *Model output* box .

To reflect in the Command Window of Matlab the model parameters estimation double-click on the model icon and in the opened window press the button "*Present*".

8.4 Content of report

The report on laboratory work must contain:

- work purpose and assignment for laboratory work;
- block diagram of the "measurement equipment";
- the results of the experiments on the object derived into the Workspace of Matlab system;
- the *ident* interface, that reflects the stages of loading, processing and placement of data in the field of identification and verification of data;
- the result of object identification by correlation method;
- the result of object identification by a spectral method;
- the comparative graphics of the received models;
- the result of identification of the different orders parametric models;
- the comparative graphs of parametric models;
- the estimations of the model parameters in the Command Window of Matlab;
- the conclusions of the work.

8.5 Assignment options

Assignment options are selected from Work/Objects8 folder of Simulink package.

8.6 Control questions

8.6.1 What is the identification?

8.6.2 The models of what objects can be estimated using the System Identification Toolbox?

- 8.6.3 What is the GUI *ident* intended to?
- 8.6.4 What block of Simulink is used to exchange the data between different tools of Matlab system?
- 8.6.5 How can I load the data of the experiment in graphical user interface of identification?
- 8.6.6 What is the pre-processing working out of the experiment data used for?
- 8.6.7 How in the interface window can you obtain the graph of the transition function?
- 8.6.8 How can you do a comparison of different structures parametric models?
- 8.6.9 What is model verification?
- 8.6.10 How can you evaluate the accuracy of the recovered model in the *ident*?

9 Laboratory work №9. Identification of second order linear object

Work purpose: study a procedure second order linear object identification.

9.1 Assignment for the laboratory work

In the process of fulfilling the laboratory work the student must:

- learn the procedure of converting a continuous model to discrete one;
- form residual functional for model identification;
- obtain system of normal equations;
- collect block-diagram for identification of second order linear object;
- fulfill experiments at the object;
- write down values of input and output signals;
- using experiments results evaluate model parameters;
- write down model equation and evaluate its accuracy.

9.2 Main concepts

Consider a linear continuous model of second order:

$$T_1 \frac{d^2 y}{dt^2} + T_2 \frac{dy}{dt} + y(t) = k_0 u(t). \quad (9.1)$$

We know the measurements of input and output signals in discrete time moments $k = 1, 2, \dots$. Identification problem is the definition of unknown model parameters.

One of the methods of solution this problem is discretization of the model by second order difference equation.

Applying the discretization procedure to (9.1) model, we get the difference equation:

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + b \cdot u(k-1), k = 1, 2, \dots, N, \quad (9.2)$$

where

$$a_1 = \frac{2T_1 - T_2 \times \Delta t}{T_1}, a_2 = \frac{(T_2 - \Delta t) \times \Delta t - T_1}{T_1}, b = \frac{k_0 \Delta t^2}{T_1} \quad (9.3)$$

are parameters of discrete model, which should be evaluate, Δt is the quantization period.

We will use also the residual functional minimization procedure for this identification problem. We minimize the following functional:

$$Q(a_1, a_2, b) = \sum_{i=1}^N \{y(k) - [a_1 y(k-1) + a_2 y(k-2) + b \cdot u(k-1)]\}^2 \rightarrow \min. \quad (9.4)$$

As the functional (9.4) is smooth function, the problem of evaluating model parameters is solved by classical way: equating to zero the derivatives by all unknown parameters, we get the system of algebraic equations; solution of last system gives us the values of unknown model parameters.

9.3 The order of fulfilling the laboratory work

Implement a "test bench" for fulfilling the identification procedure:

- load in the Simulink the "object" from the folder Work/Objects8 (by the variant);
- set a step signal at the object input; get the output signals;
- the results of the experiment, i.e. the measured values of input and output signals, pass to the Workspace. This data will be used further for identification procedure.

Transform the equation (9.1) to discrete form (9.2).

Write down the residual functional for discrete function.

Solve the problem of this functional minimization:

- equating to zero the functional derivatives by unknown model parameters, write down the system of normal equations for determining the model coefficients;
- convert the obtained system to standard form (of linear algebraic equations system);
- solve this system using Matlab procedure in the Command Window.

Using the obtained parameter values, write down the obtained discrete model equation.

Using (9.3) formulas, calculate the initial model (9.1) parameters. Convert model equation to transfer function.

Add the block with model to the diagram realized in p.9.3.1; set the same signal (as at the "object") at its input port; the outputs of the 'object' and model are set in one Scope. Compare results.

Conclusion by the work.

9.4 Content of report

The report on laboratory work must contain:

- work purpose and assignment for the laboratory work;
- block diagram of the "measurement equipment";
- the results of the experiments on the object derived into the Workspace of Matlab system;
- the procedure of converting continuous model to discrete one;
- the residual functional for model parameters determination;
- the condition of this functional minimization;
- the system of algebraic equations for determination of model parameters;
- the solution of this system;
- the equation of continuous model;
- the comparative graphics of the received model and objects outputs;
- the conclusions of the work.

9.5 Assignment options

Task options are selected from Work/Objects9 folder of Simulink

9.6 Control questions

9.6.1 How is the discrete model represented?

9.6.2 How does continuous model to discrete convert?

9.6.3 What criterion is used for model parameters evaluate?

9.6.4 How can you get in Matlab Workspace the output signal values which were formed in Simulink?

9.6.5 What is the residual functional?

9.6.6 How is the problem of the residual functional minimization solved?

9.6.7 Difference between the residual functional for continuous and discrete models.

9.6.8 Why do the model and object outputs have the difference? How can you get more accurate results?

10 Laboratory work №10. Nonparametric identification of dynamic objects

Work purpose: to learn the methods of non-parametric identification of system and skills to work in the Curve Fitting Toolbox of Matlab.

10.1 Assignment for the laboratory work

In the process of fulfilling the laboratory work the student must:

- to learn the methods of non-parametric identification and the methods of functions approximation [1,2,3];
- to study the methods of working in the Curve Fitting Toolbox;
- to form a system of model equations for non-parametric identification;
- to fulfill the procedure of non-parametric identification to determine the discrete values of the impulse transient function;
- using the Curve Fitting Toolbox, to receive a smoothed solutions of the nonparametric identification problem.

10.2 The use of correlation functions for non-parameterized identification

In the simplest case, the nonparametric identification is the definition of some functions in the object operator. The specificity of a linear dynamic object is uniquely determined by its response to a single impulse influence. This circumstance is in the basis on the definition of *nonparametric* models, which is characterized by impulse (*weight*) transition function.

The methods of determining the dynamic characteristics of the investigated object using the specific signals as input signal measuring the corresponding responses are often nonapplicable for many reasons. Consider a method based on statistical representations. In the statistical method the natural random signals of identifiable object are used as the source information. This may occur a number of errors, since the probability that the random signals have the required frequency spectrum is very small. Therefore, the correlation and spectral functions are widely used for the identification by statistical method.

When identifying the linear stationary dynamic object by statistical methods the weighting function is determined from the *Wiener-Hopf* equation:

$$R_{yx}(\tau) = \int_0^T g(t)R_{xx}(\tau - \theta)d\theta. \quad (10.1)$$

One way of solving this equation is numerical. The essence of the numerical method lies in the possibility of presenting of this equation by a system of linear algebraic equations. To do this, we replace the integral by a finite sum:

$$R_{yx}(\tau) = \sum_{n=0}^N g(n\Delta)R_{xx}(\tau - n\Delta) \times \Delta,$$

where Δ is the sampling interval;

$n\Delta$ is the discrete time;

N is the number of intervals.

This equation is true for $\tau = \Delta, \dots, N\Delta$.

Substituting in this equation N values of τ , we get the system of N equations. It is a system for determining the values of the impulse transition function in discrete points $\Delta, \dots, N\Delta$.

So, we get the system of equations:

$$Ag = b, \quad (10.2)$$

where A is a square matrix.

$$A = \begin{vmatrix} R_x(1) & R_x(2) & R_x(3) & \dots & R_x(N) \\ R_x(0) & R_x(1) & R_x(2) & \dots & R_x(N-1) \\ R_x(1) & R_x(0) & R_x(1) & \dots & R_x(N-2) \\ \dots & \dots & \dots & \dots & \dots \\ R_x(N) & R_x(N-1) & R_x(N-2) & \dots & R_x(0) \end{vmatrix},$$

$b = [b_1, b_2, \dots, b_m]^T$ is a column vector with components $b_i = R_{yx}(\tau)/\Delta$, $i = 1, \dots, m$;
 $g = [g_1, g_2, \dots, g_m]^T$ is a column vector of the impulse transition function desired values.

The solution of this system allows to determine the discrete values of the ordinates of the weighting function $g(\tau)$ at the points $t, 2t, \dots, mt$. These solutions are received with large errors, since instead of the true values of correlation functions their estimations are used, and the system is badly-conditioned. Although the received values of the pulse transition functions have low average squared error close to the minimum, these functions do not correspond to the physical sense of the processes in the object. The physical meaning have a smooth solution. Therefore, different procedures can be used to smooth the impulse transition function.

10.3 The approximation of discrete impulse transition function

To obtain the smoothed impulse transient functions, we will use the procedure of this function values approximation using any approximating polynomial. Most commonly used the decomposition for any system of *orthogonal* polynomials:

$$g(t) = \sum_{k=0}^N a_k \varphi_k(t),$$

where $\varphi_k(t)$ is the specified system of function;

N is the order of decomposition;

a_k is the series coefficients that are computed by the formula:

$$a_k = \sum_{t=0}^m g_k \varphi_k(t).$$

Series functions are selected so as to achieve good approximations with a small number of them. It is advisable to choose orthogonal functions, transformed by Laplace (the Chebyshev polynomials, Laguerre, Legendre).

10.4 The order of fulfilling the laboratory work

During a certain time interval the input and output signals of the object were registered. By the results of these measurements the autocorrelation and co-correlation functions of input and output signals were calculated (see table of options).

Write down the system of linear algebraic equations (10.2) for your option.

Solve the system (10.2) using the command window of MatLab (same as item 7.4.6 of the laboratory work № 7).

Using the Curve Fitting Tool, fulfill the approximation of the received discrete values by the various orders orthogonal polynomials (for even options Laguerre polynomials are used, for odd options - Legendre polynomials).

Select the polynomial that best approximates an impulse weighting function.

Using the received impulse transient function, write the non-parametric model:

$$y(t) = \int_0^T g(\theta) \times x(t - \theta) d\theta .$$

Assemble the block-diagram by the received equation.

Fulfill the experiments, giving to the model input the different signals; the output signals reflect on the *Scope*.

Table 10.1 – Assignment options

№ of options	t, min	0	1	2	3	4	5	6	∞
1	R _x (t)	1	0,37	0,16	0,05	0,02	0,01	0,005	0
	R _{xy} (t)	0,1788	0,4729	0,3866	0,2565	0,1454	0,06	0,03	0
2	R _x (t)	1	0,47	0,2	0,1	0,05	0,01	0,005	0
	R _{xy} (t)	0,2376	0,5128	0,1343	0,2848	0,155	0,003	0,003	0
3	R _x (t)	1	0,53	0,26	0,15	0,1	0,02	0,003	0
	R _{xy} (t)	0,3049	0,5792	0,5575	0,4152	0,2176	0,002	0,002	0
4	R _x (t)	1	0,5	0,3	0,2	0,09	0,03	0,003	0
	R _{xy} (t)	0,3978	0,7428	0,7030	0,5960	0,3200	0,001	0,001	0
5	R _x (t)	1	0,52	0,28	0,18	0,1	0,02	0,005	0
	R _{xy} (t)	0,4872	0,9122	0,8998	0,7320	0,4088	0,002	0,002	0
6	R _x (t)	1	0,46	0,22	0,12	0,1	0,03	0,004	0
	R _{xy} (t)	0,3825	0,8008	0,7846	0,6214	0,3794	0,001	0,002	0
7	R _x (t)	1	0,38	0,19	0,04	0,02	0,01	0,005	0
	R _{xy} (t)	0,1735	0,4398	0,3637	0,2415	0,1026	0,003	0,002	0
8	R _x (t)	1	0,45	0,2	0,1	0,05	0,01	0,004	0

	$R_{xy}(t)$	0.2531	0.5655	0.5020	0.3365	0.1445	0,003	0,002	0
9	$R_x(t)$	1	0,39	0,19	0,06	0,01	0,001	0,0001	0
	$R_{xy}(t)$	0.2428	0.6008	0.5659	0.4181	0.1166	0,0005	0,003	0
10	$R_x(t)$	1	0,35	0,18	0,05	0,02	0,01	0,005	0
	$R_{xy}(t)$	0.2495	0.6569	0.5835	0.3389	0.1058	0,002	0,003	0

10.5 Content of report

The report on laboratory work must contain:

- the array of results of solving this system;
- the sequence of fulfilling the approximation of the impulse transition function discrete values;
- the approximating expressions for the impulse transition function;
- the reasoned choice of the best approximating polynomial;
- the block-diagram of the non-parametric model;
- the graphic of the model output signal.

10.6 Control questions

10.6.1 What is the essence of nonparametric identification?

10.6.2 Give the definition to the impulse transient function.

10.6.3 Give the definition to the autocorrelation and co-correlation functions.

10.6.4 What is the purpose of using the correlation function in nonparametric identification?

10.6.5 What do objects describe the equation of Wiener-Hopf?

10.6.6 How is the Wiener-Hopf equation transformed in the system of algebraic equations?

10.6.7 Definition of orthogonal polynomial.

10.6.8 What are the advantages of approximation by orthogonal polynomials?

10.6.9 Explain the purpose of the Curve Fitting Tool.

10.76.10 What does function the best results of the approximation give?

References

- 1 Ибраева Л.К. Моделирование и идентификация объектов управления. Конспекты лекций для студентов специальности 5В070200 – Автоматизация и управление.- Алматы: АУЭС, 2016 – 56 с.
- 2 Ибраева Л.К., Хисаров Б.Д. Моделирование и идентификация объектов управления. Учебное пособие. – Алматы: АИЭС, 2009. – 209 с.
- 3 Дьяконов В.П., MATLAB и SIMULINK для радиоинженеров - М. : ДМК Пресс, 2011. – 975 с.
4. Karel J. Keesman. System Identification- Springer-Verlag London Limited 2011. – 333 p.
- 5 <http://matlab.exponenta.ru/simulink/book2/15.php>

Lida Kuandykovna Ibrayeva
Ainur Erlanovna Tashibayeva

MODELING AND IDENTIFICATION OF CONTROL OBJECTS

Methodical guidelines to laboratory works for students of specialty
05070200 - Automation and Control

Editor Kurmanbekova M.D.
Specialist for standardization Muhametsaryeva G.I.

Signed for printing __.__.__.
Circulation 30 copies
Volume 3.5 e.-p. sheets

Format 60x84 1/16
Printing paper №1
Order . Price 1750 tg.

Copying Bureau of the Noncommercial Joint-Stock Company
"Almaty University of Power Engineering and Telecommunications"
050013 Almaty, 126/1, Baitursynov str.