**DISCIPLINE: CONTROL IN TECHNICAL SYSTEMS**

*Kozlov Oleg Stepanovich*

*Shchekaturov Alexander Mikhailovich*

**LABORATORY WORK No. 3**

**ANALYSIS IN SIMINTECH ENVIRONMENT**

**OF DYNAMIC SYSTEMS ASSIGNED IN**

**THE CAUCHY’S FORM AND STATUS VARIABLES**

**Moscow, 2013**

**CONTENTS**

PURPOSE OF THE PAPER 3

INTRODUCTION 4

1 NEW METHODS OF GENERATION OF ACS MATHEMATICAL MODEL IN SIMINTECH 5

1.1 Transformation of ACS Mathematical Description and Formulation of Problems of a Regular Additional Task 5

1.2 Creation of “Parallel” ACS in the Form of a New Submodel 7

1.3 Setting of ACS Parameters via the Mechanism of Global Parameters 10

1.4 Generation of Dynamic Equations Using “Programming Language” Block 11

1.5 Generation of ACS Dynamic Equations in Status Variables 18

1.6 Implementation of “Wireless” Data Transfer 23

2 INDEPENDENT STUDY OF NR ACS DYNAMICS SET IN STATUS VARIABLES AND IN THE CAUCHY’S FORM 28

# PURPOSE OF THE PAPER

* introduction of new methods of generation of mathematical models of ACS dynamics in SimInTech environment including:
  + procedures for creation of multi-layer diagrams;
  + implementation of Global Parameters mechanism;
  + description of mathematical model of ACS dynamics in status variables;
  + application of the integrated Programming language (and its mathematical function Interpreter) for setting ACS dynamic equations;
  + methods for implementation of “wireless” data transfer;
* independent study of transient processes in the simplest ACS model of nuclear reactor (NR ACS) including:
  + reduction of a mathematical description of NR ACS dynamics to the Cauchy's problem;
  + generation of a NR ACS dynamic model using:
    - description in status variables (in matrix form);
    - “Programming Language” block, which enables qualitative expansion of options of ACS structural diagram compact transformation;
* modeling of transient processes in NR ACS in the process of delivery of controlling and disturbing responses.

# INTRODUCTION

In laboratory works No. 1 and No. 2 you have considered the procedures of operation in the SimInTech environment as applicable to the analysis of dynamic processes in ACS, whose dynamics model was described in “input-output” variables. Nevertheless such type of assignment of ACS structural diagram is rather adoptable for simple tasks of the training plan.

In many training and in branch oriented tasks, in particular, differential equations of dynamics of an object under research are written down in the Cauchy’s form, i.e.: as a system of non-linear differential equations of the 1st order solved relative to derivates. An attempt to implement even the simplest non-linear operations in “input-output” variables in the right parts of the equations, using related typical blocks (\*, /, sin, ln etc.) for that, leads to an abrupt increase of the structural diagram size. Therefore, generation of the ACS dynamics structural diagram with the use of only relatively simple typical blocks is ineffective: the structural diagram becomes “unreadable” since, in many cases, it cannot be completely displayed within the screen size of one display.

During generation of a relatively complex structural ACS diagram (or any other dynamic diagram) it is advisable to “split” the structural diagram mentally (or on a sheet of paper!) into several separate (relatively low-bulk) fragments. Then generate the structural diagram for each fragment as a separate substructure (Submodel) with inputs and outputs (input and output ports, correspondingly) and then set up the structural diagram of the whole system from Submodels.

Such approach is used in many software systems that implement structural modeling methods including in the SimInTech environment. Nevertheless, a number of new methodical solutions have been implemented in the SimInTech environment, which allows reducing the number of blocks in the structural diagram in a quality manner (up to orders).

Now let us become “acquainted” with the new methodological solutions.

# 1 NEW METHODS OF GENERATION OF ACS MATHEMATICAL MODEL IN SIMINTECH

## 1.1 Transformation of ACS Mathematical Description and Formulation of Problems of a Regular Additional Task

This section will deal with the most significant new software methodological solutions implemented in the SimInTech environment, which have no equivalents in domestic software systems of such purpose and in majority of known foreign ones.

New SimInTech environment software capabilities for generation of a mathematical model of dynamics of an object under research will be considered in the process of execution of a new multi-stage additional task for demonstrative and familiarization problem, whose structural diagram has had an appearance close to the one depicted in Figure 1.1, on completion of tasks for calculation of amplitude and phase frequency ACS parameters.

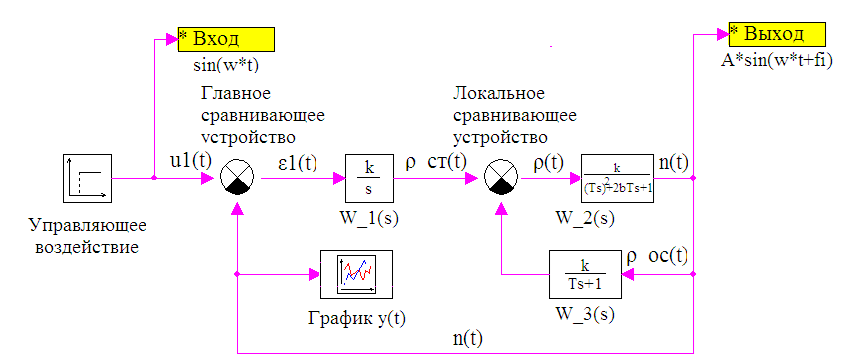


Fig. 1.1

In the demonstrative and familiarization problem under consideration the Controlled Object is described by means of three blocks (elements): blocks with transfer functions **W\_2(s)** and **W\_3(s)**, as well as block **Local Comparison Device**.

It can be easily seen that dynamics of the Controlled Object in this problem is described via the following system of linear equations:

|  |  |
| --- | --- |
|  | (1.1) |
|  |  |

with zero initial conditions, i.e.

On introducing new dynamic variables , one can reduce the mathematical description of dynamics of the Controlled Object to the Cauchy’s form.

|  |  |
| --- | --- |
|  | (1.2) |

where  and initial conditions for new dynamic variables are equal to zero.

The mathematical description of ASC dynamics equations can be reduced to the Cauchy’s form in general, after addition of the dynamic equation for integrating regulator (block titled as W\_1(s)) and algebraic relation for the Main comparison device to the system (1, 2). In this case the system of ACS dynamic equations “prepared” for transfer to description of status variables will have the following appearance:

|  |  |
| --- | --- |
|  | (1.3) |

where 



After introduction of new regular dynamic variables , that are, correspondingly, equal to , description of ACS dynamics can be presented in matrix form, i.e., in status variables:

|  |  |
| --- | --- |
|  | (1.4) |

where the column vector of derivates of status variables *x'(t)*, the column vector of status variables *x(t)*, system matrix *А*, input matrix *B*, output matrix *C* and bypass matrix *D* are equal to (in this task):

|  |  |
| --- | --- |
|  | (1.5) |

Let us finally formulate individual problems that are to be solved by you during execution of a *regular additional* task:

1. Add a new “parallel” ACS to the structural diagram of “main” ACS (see Fig. 1.1) as a nested **substructure** (Submodel).
2. Set coefficients of the system of equations of “parallel” ACS via the mechanism of Global Parameters.
3. Set the description of equations of the Controlled Object dynamics (see the system (1, 2)) in “Parallel” ACS with the use of **“Programming Language”** block from the **Dynamic** links library.
4. Set the description of ACS dynamic equations in general (see systems (1.3) and (1.4)) using **Status Variables** typical block from the **Dynamic** links library located inside the “parallel” ACS.
5. Implement “wireless” data exchange between “main” and “parallel” ACS using **To memory** and **From memory** blocks from the **Substructures** library.

It should be noted that items 2 and 3 suggested to you as the stages of a regular additional task cannot be executed in the environment of other software systems of the same purpose (neither domestic, nor foreign ones), while items 1 and 4 can be fully executed only in the most popular foreign software system – i.e., in SIMULINK.

## 1.2 Creation of “Parallel” ACS in the Form of a New Submodel

Addition of “parallel” ACS as a new Submodel to the structural diagram (see Fig. 1.1) will be executed as follows:

**Stage 1** – transfer of typical **SimInTech Submodel** block to the Diagram Window.

Considering the fact that the project (task) with its structural diagram the same as depicted in Fig. 1.1. has been saved by you on the hard disk, open it (the project). Transfer the cursor onto the Substructures tab in the “Line” of typical blocks and initiate it by clicking it with the left mouse key. Then transfer the **SimInTech Submodel** block from the “Line” of typical blocks to the Diagram Window, as you have done it before with other blocks, when executing the demonstrative and familiarization problem. In the field of the Diagram Window a new block – **SimInTech Submodel** (without input and output ports) – will appear.

**Stage 2** – filling of the **SimInTech Submodel** internal structure.

Shift the cursor in the Diagram Window to the Submodel block and double-click that with the left mouse key: the submodel diagram window will be opened (i.e., transfer to the 1st nesting level).

Transfer the Input Port block into the opened blank (submodel) diagram window (preferably into the left part of the diagram window) and the Output Port block (preferably into the right part of the diagram window) from the same library (Substructure). Comparison device, Integrator, Time schedule and “Programming Language” block.

Although at this stage of task execution all links among blocks cannot be plotted in the submodel window (since the “Programming Language” block has not yet got either input or output ports), draw those links among blocks in the submodel diagram window which can be drawn. After execution of explanatory subscriptions, the submodel diagram window will have the appearance similar to Fig. 1.2.

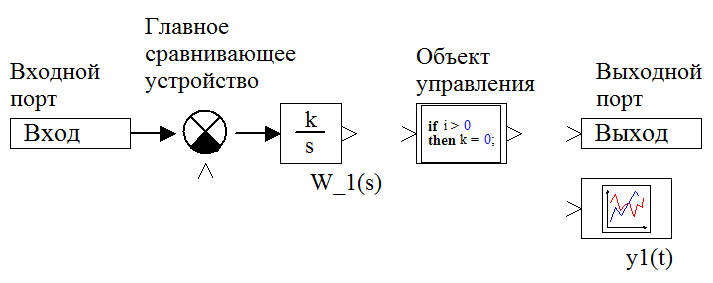


Fig. 1.2

Close the submodel diagram window by double-clicking the mouse left key in a blank space of the submodel diagram window. The Submodel icon in the Main Diagram Window will be changed: input and output ports will appear.

If conditions of any problem to be solved by you requires for the submodel structure to have, for example, 3 input ports and 2 output ports, then you should “transfer” three **Input Ports** and two **Output Ports** to the submodel diagram window. The upper input port (with **SimInTech Submodel** block oriented from left to right) will correspond to the first **Input Port** transferred to this window, the middle input port – to the second one and the lower input port (with **SimInTech Submodel** block oriented from left to right) to the third one. In the same manner, the upper output port (with **SimInTech Submodel** block oriented from left to right) will correspond to the first Output Port transferred to the submodel window, etc.

Input **substructure** ports can be vector (multi-core), for example, 5 signals are simultaneously transmitted at the first input, while 6 signals are transmitted at the second one and 4 signals – at the third one. It is evident that such vector signals shall be pre-generated, by means of Multiplexer unit, for example. Output signals of Input Port blocks inside the submodel diagram window usually shall be demultiplexed for the further processing.

Output substructure ports can be also vector, which requires relevant pre-multiplexing of output signals transmitted inside the substructure (SimInTech Submodel) to Output Port blocks.

Attention! If there are several Input Ports and Output Ports in a submodel, it is recommended immediately after transfer of each Port to make a proper subscription using a standard procedure for execution of explanatory subscriptions. For example, give a “unique” name to the 1st transferred **Input Port**: Input No. 1, etc.

**Stage 3** – inclusion of **SimInTech Submodel** into the main structural diagram.

Connect the SimInTech Submodel to the Control Action block and Curve y(t) block by links after preliminarily changing the number of inputs in the Time Schedule block by two. Make an explanatory subscription under the new SimInTech Submodel block. Main Diagram Window will have the appearance as depicted in Fig. 1.3.

By this the process of inclusion of a new substructure (“Parallel” ACS) created by you into the main structural ACS diagram is about completed.

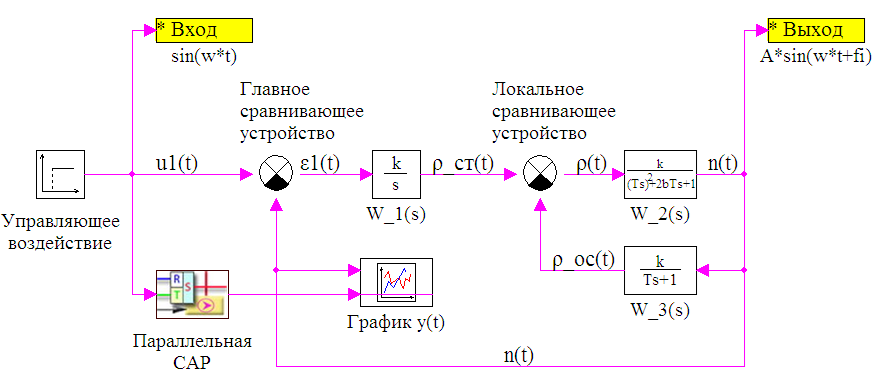


Fig. 1.3

In the same manner some more new substructures can be introduced into the Main Diagram Window, and also new substructures can be introduced into the submodel diagram window (substructures of the 2nd nesting level).

On clicking the **SimInTech Submodel** block with the right mouse key, a context window will be activated, which is provided with an additional option: Actions->Save and bind that with a file; this allows the Submodel to be saved on the hard disk under an individual name (.sub extension) and be used as a “blank” for generation of a structural diagram of another task.

**Submodel** dialog window (see Fig. 1.4) that is activated by a right click on the block and then by a left click on the Properties option has, among others, the following two frequently used dialog rows in the “Common” tab:

* Graphic presentation – is intended for changing the unit icon for a new one (in bmp-format);
* Color of the submodel background – is intended for changing monotone background of the submodel window (white, by default) for an image (in bmp-format), for example, depicting a mnemonic, against which the structural diagram is created in the given submodel window.

It should be noted that a change of the submodel window background for a picture will require for the User to “pre-design” this image.

Setting of Global Parameters for each Substructure can (and sometimes must) be carried out after entering inside the substructure and going to the “Parameters” tab – in the same manner as the “Parameters” tab on the Main Diagram Window.

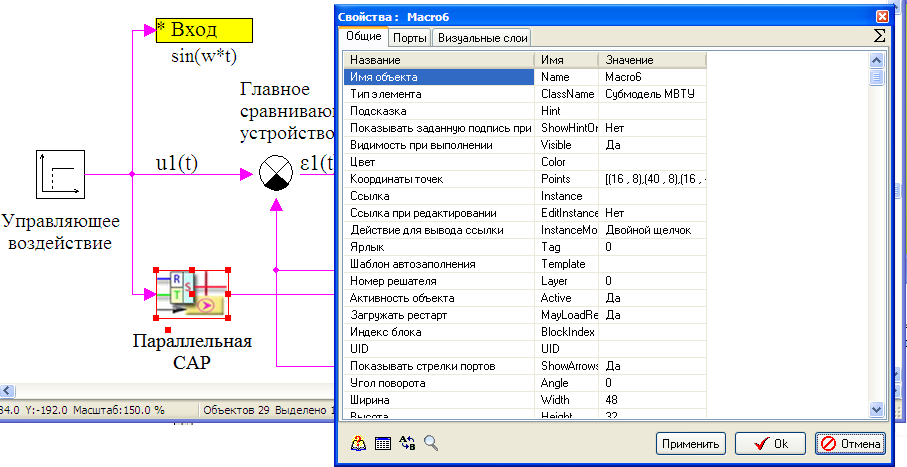


Fig. 1.4

## 1.3 Setting of ACS Parameters via the Mechanism of Global Parameters

Interpreter of mathematical functions (and Programming Language in a block with the same name) integrated into the SimInTech environment ensures functioning of the **“Global Parameters Editor”** window of the Project (Submodel), by means of which parameters of structural diagram blocks can be set via the mechanism of Global Parameters.

Project (Submodel) Global Parameters Editor window is an actual window of the text editor, in which you can set values or expressions of a number of ACS parameters called as Global Parameters that remain unchanged in the process of modeling. Interpreter of mathematical functions “recognizes” over 30 operators including purely mathematical   
(+, -, \*, /, sin, tg, ln, etc.), logical (if, for, etc.) and functional operators (time, step, interpol, etc.). More detailed information on the Interpreter of mathematical functions will be presented in the following subsection.

Open the Project (Submodel) Global Parameters Editor window by left-clicking the **Parameters** tab of the macroblock in the toolbar located at the left in the Diagram Window.

Enter a text that described the setting of global parameters and comments (in squiggle brackets) from the keyboard in a way as depicted in Fig. 1.5.

After entering the basic text (font color is black) and comments on that (font color is green), go back to the “Diagram” tab, i.e., close the Project (Submodel) Global Parameters Editor window for a while. For that click on the “Diagram” button (the upper from the left above the “Parameters” tab).

By this the procedure for setting the Global Parameters is completed.

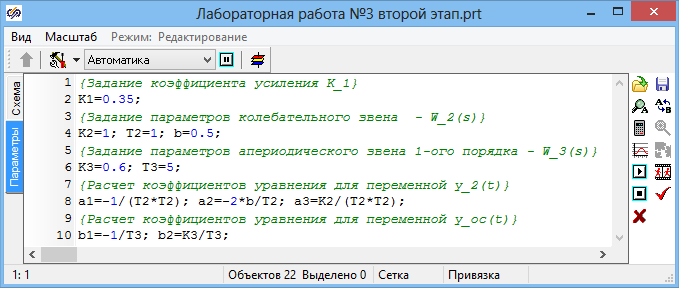


Fig. 1.5

If any ACS parameters in the project (task) are set as global ones in the Main Diagram Window, then those can be used for setting parameters of particular blocks not only in this window but in all nested structures (Submodels).

If any global parameter (set in the Main Diagram Window) is set again as a global one in a submodel diagram window then the latter will redefine the value of the parameter set earlier and it can be used for setting parameters of particular blocks both in the given submodel diagram window and in all “subsidiary” submodels belonging to a deeper nesting level.

## 1.4 Generation of Dynamic Equations Using “Programming Language” Block

Considering the fact that it is impossible to generate an absolutely full library of typical blocks, tools have been designed in the SimInTech environment that allow the User to expand the content of his personal library due to creation of new types of blocks, for example, by means of the programming language integrated into SimInTech.

One of “non-standard” typical blocks of the Dynamic links library also functions on the basis of the programming language; that is “Programming Language” block that enables creation of samples of blocks with their own individual mathematical models directly in the process of operation.

Dialog window of this block is fully equivalent to the Project (Submodel) Global Parameters Editor window. The User writes down dynamic equations in text form that is close to their appearance if written down on a sheet of paper with a pen.

Mathematical description of the block can comply with multi-dimensional non-linear dynamic system in the Cauchy’s form.

|  |  |
| --- | --- |
|  | (1.6) |

where  are known non-linear functions of variables  and input actions ; besides, coefficients (both constant and variable) included into any one of the equations can act as input actions (1.6).

The first equation of the system (1.6) can be missing: in this case the “**Programming Language**” block executes algebraic transformations of input values. Application of this block as a functional one is rather effective in case complex functional transformations are available in the model when application of elementary functional typical blocks for these purposes will lead to unreasonable complication of the structural diagram.

“**Programming Language**” block “identifies”, in particular, the following standard mathematical operations and functions:

|  |  |
| --- | --- |
| **+** | – addition; |
| **abs** | – module; |
| **sin** | – sine; |
| **arcsin** | – arcsine; |
| **-** | – subtraction; |
| **sign** | – sign; |
| **cos** | – cosine; |
| **arccos** | – arccosine; |
| **\*** | – multiplication; |
| **exp** | – exponent; |
| **tg** | – tangent; |
| **arctg** | – arctangent; |
| **/** | – division; |
| **ln** | – logarithm; |
| **ctg** | – cotangent; |
| **arcctg** | – arccotangent; |
| **^** | – power; |
| **pi** | – 3.1415; |
| **e** | – 2.71828; |
| **( )** | – brackets; |

More details about the programming language can be found in the SimInTech answer wizard.

Programming language and, respectively, “**Programming Language**” block of this version of SimInTech software includes also 8 special functions to implement determination of basic thermodynamic properties of water and water steam (known Vukolovich tables within the precritical range: for pressure – from 0.09 to 50 MPa and for temperature – from 10 to 800°С). Record of these functions appears as follows:

|  |  |
| --- | --- |
| waterps(P,flag); | calculation of properties of water by saturation line pressure; |
| waterts(T,flag); | calculation of properties of water by saturation line temperature; |
| steamps(P,flag); | calculation of properties of steam by saturation line pressure; |
| steamts(,flag); | calculation of properties of steam by saturation line temperature; |
| waterpt(P,T,flag); | calculation of properties of water by pressure and temperature; |
| waterph(Р,H,flag); | calculation of properties of water by pressure and enthalpy; |
| steampt(P,T,flag); | calculation of properties of steam by pressure and temperature; |
| steamph(P,H,flag); | calculation of properties of steam by pressure and enthalpy. |

For example, record v\_sp = waterps(1е6,4); corresponds to calculation of specific water volume v\_sp (flag is equal to 4) on the saturation line with pressure of .

Values of parameter flag can vary from 1 to 8 and correspond to calculation of the following thermodynamic characteristics of water or water steam:

1 – pressure; 2 – temperature;

3 – enthalpy; 4 – specific volume;

5 – Prandtl number; 6 – dynamic viscosity;

7 – heat conductivity factor; 8 – entropy;

9 – specific heat capacity Cp; 10 – specific heat capacity Сv;

11 – derivative density for enthalpy with constant pressure ;

12 – derivative density for pressure with constant volume ;

Input parameters and return values of thermodynamic parameters of water or water steam are presented in SI system (except temperature that is measured in Celsius degrees).

Mastering of procedures for generation of mathematical model of dynamics of any device or operating process fragment using the “Programming Language” block will be considered by an example of generation of dynamic equations of the Controlled Object located in the submodel window of the 1st nesting level (see Fig. 1.2.).

Open the submodel “parallel” ACS window, shift the cursor onto the “Controlled Object block (see Fig. 1.2) and double-click with the left “mouse” key: a new Editor window will be opened, in which you should put down the expressions and differential equations corresponding to the mathematical model of this block (equations system (1.2.)).

Toolbar (command buttons) of the Editor window and User interface are fully identical to the Project (Submodel) Global Parameter Editor window described above.

Enter the mathematical model of dynamics of the Controlled Object block from the keyboard (in text form) as it is shown in Fig. 1.6, in which a screen copy of the Editor window with mathematical model of dynamics of the considered block and comments on that is presented.

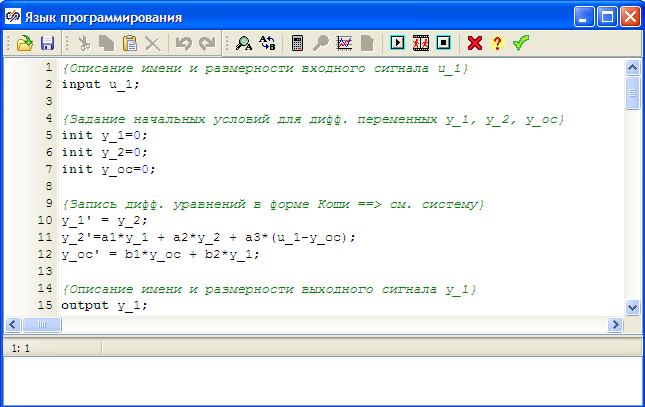


Fig. 1.6

If the “Programming Language” block has inputs (input ports) then the first executable row (apart from the comment row) shall contain input operator that describes input signals to this block including the input name and its dimension.

In this example the 1st executable row (input u\_1;) assigns a unique name – u\_1 to the 1st (and only) input. If this block, for example, had got 2 inputs and the 1st input had been a “three-core” (vector) one and the 2nd input had been a “five-core” one, then the 1st executable row would have the following appearance: input u1[3], g[5]; Rectangular brackets are used for description of input dimension.

If the “Programming Language” block describes the dynamics of the simulation object as a system of differential equations in the Cauchy’s form then the second executable row (apart from the comment row) shall contain the initialization operator that describes initial conditions for dynamic (differential) variables, for which hereinafter ordinary differential equations in the Cauchy’s form will be written down.

In this example the 5th and next two executable rows (init y\_1=0; init y\_2=0;   
init y\_fb=0;) specifies initial conditions (at t = 0) for 3 dynamic variables: y\_1(t), y\_2(t) and y\_fb(t).

In the case, when the mathematical model of block dynamics is described by a large number of differential equations, by 5, for example, then the 2nd executable row will have the following appearance: init x1=0; init x2=1; init x3=2; init z=0; init R=0; , where х1, х2, х3, z, R are dynamic variables, for which hereinafter differential equations (a system of equations, in more exact terms) in the Cauchy’s form will and must be written.

Differential equations of dynamics of the Controlled Object are directly written down in 10–13 executable rows where the apostrophe symbol means a time variable, while values of coefficients a1, a2, a3, b1, b2 are transferred to the block via the mechanism of Global Parameters (see the previous subsection).

If the “Programming Language” block has outputs (output ports) then the last executable row shall contain the output operator that describes output signals from the “New” block including output names and their dimensions.

In this example the last row (output y\_1;) describes one output signal (y\_1) without indication of the output signal dimension in rectangular brackets.

If the block had got 2 vector outputs (2-core and 3-core ones) then the last executable row would have, for example, the following appearance: output z1[2]; z2[3]; In this case hereinbefore all components of output signals shall be identified (calculated), for example:

z1[1]=a1+sin(y1); z1[2]=exp(y2);

z2[1]=y1\*y2; z2[2]=sqrt(abs(y\_fb)); z2[3]=(y2)^a2;

After entering the whole text into the Editor window shift the cursor onto the “Apply” command button (the 1st one from the right) and click with the mouse left key: the Editor window will be closed and a submodel diagram window will be opened, in which the Programming Language will have one input and one output ports.

If the “Programming Language” block had formed two outputs (for example, z1 and z2 ==> see above) then the image of the block on the structural diagram would have two output ports. If the block is oriented “from left to right” the 1st output port will be the top one, while the 2nd output port will be the bottom one.

Complete the execution of submodel diagram window by connecting all units with links and also by entering a new value of amplification factor K1 into the block dialog window subscribed as W\_1(s). The structural diagram will take an appearance similar to the one in Fig. 1.7.

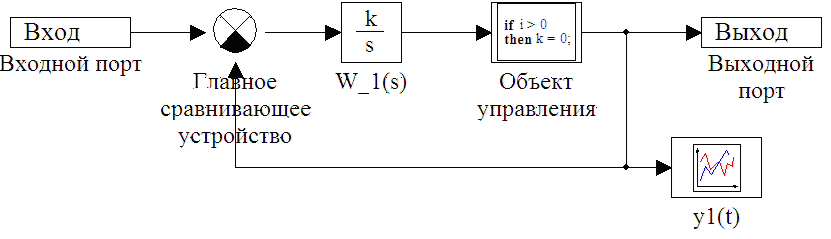


Fig. 1.7

Close the submodel diagram window (by double-clicking a blank space of the diagram window with the mouse left key) and make sure that the amplification factor in block W\_1(s) in the Main Diagram Window is equal to “optimal” value (k1 = 0.35).   
By a left click on the Continue button start up the calculation task and make sure that results of calculation for “main” and “parallel” ACS absolutely coincide. For that purpose set the following parameters in the Curve y(t) block Setting dialog window for the 1st line: type of the line – solid, double-thickness, line color – pink, and for the 2nd line: type of the line – dotted, color – blue; then overlaid calculation curves will have the appearance close to the one in Fig. 1.8.

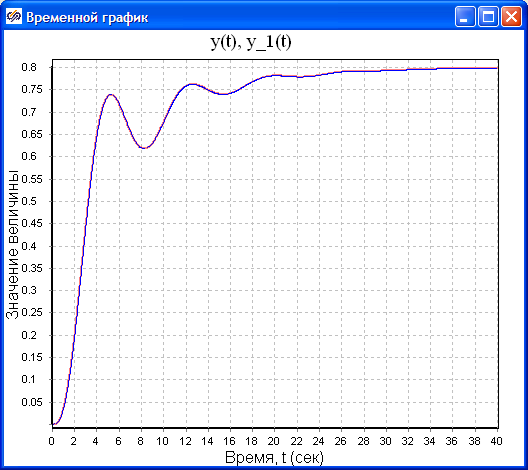


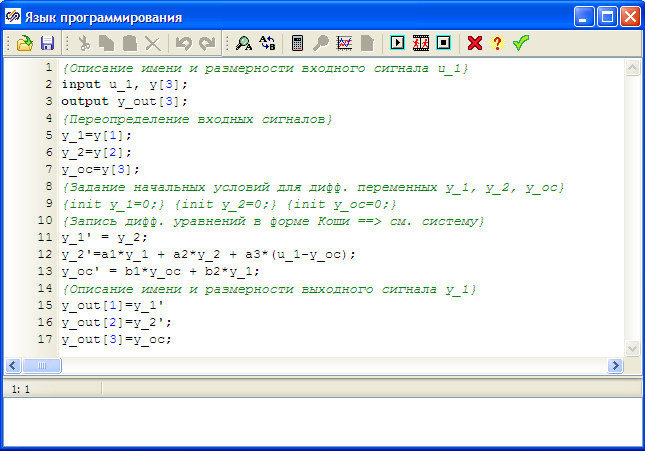
Fig. 1.8

To create the mathematical model of an operating process described by complex non-linear differential equations with variable coefficients (for example, thermohydraulics processes in components of thermophysical equipment) it is more advisable to use the Programming Language block, besides, for generation of right parts of differential equations only (written down in the Cauchy’s form).

In such variant the procedure of integration can be implemented using the Integrator typical block (block is vectorized) located behind the “Programming Language” block; besides, the output signal of the Integrator block (vector, as a rule) is delivered to the input of “Programming Language” block and does not form any algebraic “loop” (circuit).

We implement such approach for generation of the mathematical model of Controlled Object block in “parallel” ACS.

A screen copy of the Editor window is shown in the top part of Fig. 1.9, in which the algorithm of calculation of right parts of differential equations describing the dynamics of the Controlled Object block is implemented.



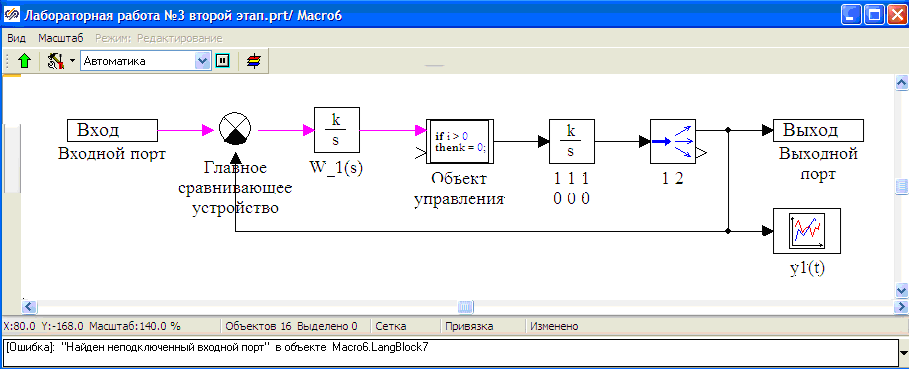


Fig. 1.9

The main text has been only slightly changed only in its appearance (if compared with Fig. 1.6). The “meaning” of the apostrophe symbol has been cardinally changed: now record y\_1'=... does not mean the first differential equation but just identifies a new variable named as y\_1'. A new input signal y has appeared which is a “three-core” (vector) one and introduces values of dynamic variables y\_1, y\_2 and y\_fb into the block after each step of integration (including a test step). Vector signal y\_out with dimension equal to three ==>   
y = [y1 y2 y\_fb] is generated for the output of the “Programming Language” block.

Screen copy of the submodel diagram window with a changed structural diagram of “parallel” ACS is presented in the bottom part of Fig. 1.9. Subscription under the Integrator typical block (in two rows) informs that this block executes an operation of integration in “vector” variant; besides, amplification factors 1 1 1 (three spaced 1 numerals) and zero initial conditions 0 0 0 (three spaced 0 numerals) are set in its dialog window. Demultiplexor block “extracts” signal y\_1 from the vector.

Change the text in the Editor window of “Programming Language” block and correct the structural diagram in the submodel window as it is done in Fig. 1.9.

Execute simulation (click on the Continue button) and make sure that curves of transitional processes (formed by the Curve y(t) block in the Main Diagram Window) in the “main” ACS and in new “parallel” ASC absolutely coincide.

Save this variant of the project (task) on the hard disk under a new name.

## 1.5 Generation of ACS Dynamic Equations in Status Variables

Let us execute the 4th stage of this task, main purpose of which (the stage) is to master the methods of generation of linear ACS dynamic equations with the use of a typical Status Variables block. For that purpose let us create the second “parallel” ACS located in the “Parallel” ACS submodel.

Open the project, whose structural diagram in the submodel window has the appearance similar to the diagram in the bottom part of Fig. 1.9.

Initiate the Dynamic links tab in the “Line” of typical blocks and transfer the typical Status Variables block to the submodel window. On connecting the block with links and executing explanatory subscriptions, make the structural diagram in the submodel window appear as depicted in Fig. 1.10.

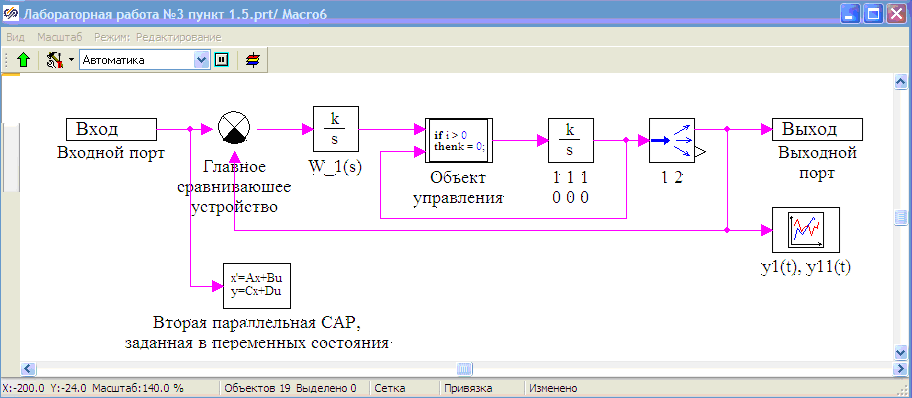


Fig. 1.10

Matrixes A, B, C and D in the Status Variables block are set not by lines (as usual) but by columns.

Matrix of input B that actually is a column vector and zero matrix of bypass D (see the relations (1.5.)) do not require any additional clarifications for entering into corresponding rows of the dialog window of the Status Variables block.

Matrix A determined for the second “parallel” ACS by the relations (1.5) during its setting in the dialog window of the Status Variables block shall be presented as a “special” row vector containing 4 components, each one of which is numerical vector and contains by 4 elements (numerical or symbolic values of corresponding column of matrix A) ==> see below.

Matrix C also shall be presented as a “special” row vector containing 4 elements, each one of which is numerical vector and contains by one element only in this task ==> see below.

Let us set the parameters of the Status Variables block via the mechanism of Global Parameters. Shift the cursor to the Parameters tab (of the macroblock) and click that with the left mouse key: Global Parameters Editor window of the “Parallel” ACS submodel will be opened. Enter the text into the Editor window as shown in Fig. 1.11.

Since coefficients а1, а2, а3, а4 and K1 are determined as global ones in the Global Parameters Editor window... of the Main Diagram Window, in this Editor window... those are used for setting matrix A elements that are not equal to zero or one. The left part of the expression А1[4] = [0 a21 a31 a41] determines that vector variable A1 has 4 elements and the right part (in rectangular brackets) sets values of these elements in numerical (0) or symbolic (a21) form.

**Attention!**

When setting the values of elements for any vector variable in symbolic form, minus sign is not allowable before a symbolic element.

For example, record А1[4] = [0 a21 a31 -K1] is incorrect due to the last element (‑K1).

If the last element of vector A1 is set in numerical form, then record А1[4] =   
[0 a21 a31 -0.35] is correct.

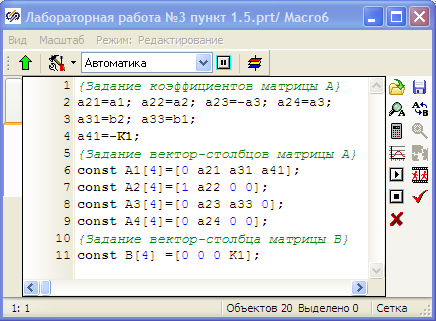


Fig. 1.11

On filling the Global Parameters Editor window of submodel diagram window, close it by clicking Apply button with the mouse left key: submodel diagram window will appear on the monitor screen.

Open the dialog window of the Status Variables block and fill its dialog rows corresponding to the Parameters tab in the same way as it is shown in Fig. 1.12, i.e., setting all elements of matrixes A, B, C and D by columns. Any column vector in dialog rows is embraced in round brackets. By default there is no space in the dialog window between the round brackets separating one column vector from another, but to ensure more visualization for previously entered data it is recommended to make 1...2 spaces (as it is done in Fig. 1.12 for setting matrix A).

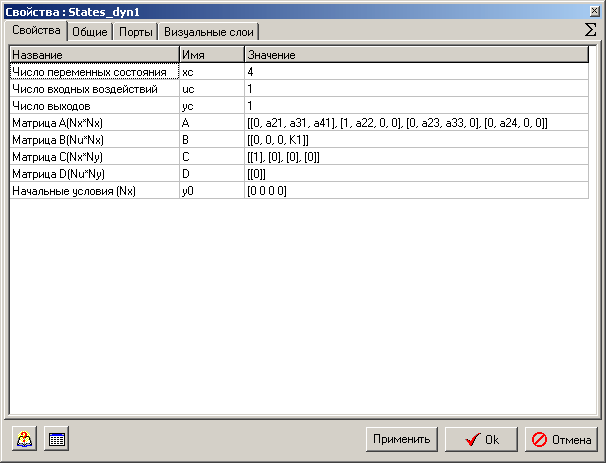


Fig. 1.12

Matrixes A, B, C and D in the dialog window can be set in more compact form using vector variables (А\_1, А\_2, А\_3, А\_4 and В\_1) determined as global parameters in the Global Parameters Editor window... ==> see Fig. 1.13.

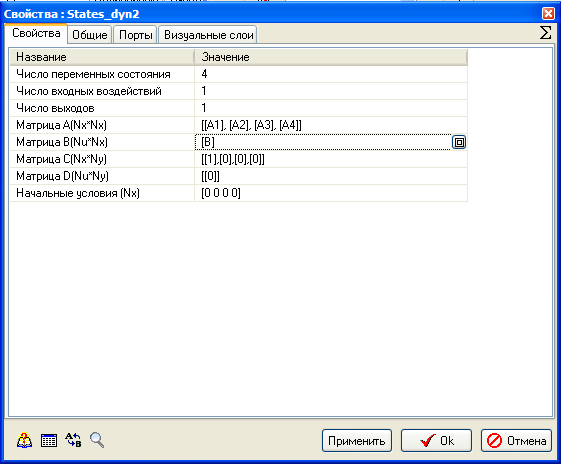


Fig. 1.13

After filling the Status Variables dialog window, as it is done in Fig. 1.12, close it by clicking the Yes button with the mouse left key.

Start the task for calculation and make sure that the curves of transitional processes in “parallel” ACS (set with the use of “New” block) and in the second “parallel” ACS (set with the use of the Status Variables block) absolutely coincide. For that purpose set the following parameters in the Time Curve block Setting dialog window for the 1st line: type of the line – solid, double-thickness, line color – pink, and for the 2nd line: type of the line – dotted, line color – blue. Overlaid calculation curves will have the appearance identical to Fig. 1.8.

Note. Curves of transitional processes are formed by the Time Curve block (subscription y1(t), y11(t)) located in the submodel diagram window (see Fig. 1.10). Absolute coincidence of curves can be checked by transferring the graphics window to table one (List option in the Graphics window pop-up menu).

Open the dialog window of the Status Variables block again and change the form of matrixes A and B setting for a compact one (see Fig. 1.13). Repeat the simulation process and make sure that curves of transitional processes in “parallel” ACS and in the second “parallel” ACS (with compact setting of matrixes A and B) absolutely coincide.

Save this variant of the project (task) on the hard disk under a new name.

## 1.6 Implementation of “Wireless” Data Transfer

Typical **To memory** unit was used in the laboratory work No. 2 for setting input and output points during calculation of ACS amplitude-phase-frequency characteristics and during calculation of coefficients, poles and zeroes of transfer functions.

Nevertheless the main purpose of this block is implementation of the 1st stage of “wireless” data transfer from one part of structural diagram to another. Therefore, the typical **From memory** block implements the 2nd stage of “wireless” data transfer. In radio engineering terms the **To memory** block is “transmitter”, while the **From memory** block is “receiver” in the mechanism of “wireless” data transfer.

In fact the following mechanism is implemented in the SimInTech environment. After transfer to the **To memory** block diagram window and assignment of a name to the variable some dynamic array is reserved in the PC RAM. When a task is started for calculation against the dimension of a signal, the dimension of the named variable is determined at the **To memory** block input, and a corresponding RAM volume is assigned for that wherein data will be recorded at each step of calculation (step of integration) in the process of simulation.

If typical **From memory** block is available in structural diagram (in any submodel window, for example), then it can be used for reading out named data from RAM and for delivering those to the **From memory** block output. Then by means of links the data can be transferred to the input of another block arranged in the same diagram window.

We will master the procedure for organization of “wireless” data transfer within the framework of the following small task.

Make sure that the project, in which the structural diagram in the submodel window appears as shown in Fig. 1.10, is not closed and go to the Main Diagram Window by double-clicking with the left mouse key in a blank space of the submodel window. If you have already closed the project open it again...

Shift the cursor onto the link transferring the control action signal to the “Parallel” ACS submodel input and click there with the mouse right key, and then select the Delete the Link option in the pop-up menu by clicking that with the left mouse key: branch of the link to the “Parallel” ACS submodel will be deleted.

In the same way delete the link (the first and only link) connecting the “Parallel” ACS submodel to the block subscribed as Curve y(t).

Initialize the Substructures library and transfer Macroblock to the Main Diagram Window and arrange that, for example, in the left bottom corner of the diagram window. Enter “Copy” subscription under this submodel (see Fig. 1.14).

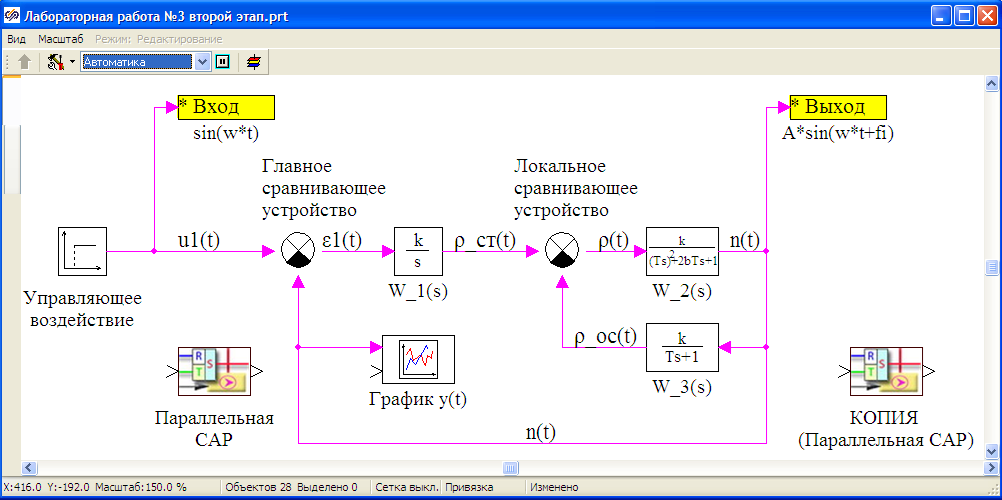


Fig. 1.14

Delete the “Parallel” ACS (by clicking that and using the Cut button).

Open the diagram window of the “Copy” submodel, then shift the cursor onto the command Insert button (the 4th one from the left in the Additional toolbar of the submodel diagram window) and then click in the submodel window field with the left mouse key: preciously cut off “Parallel” ACS submodel will be displayed in the submodel window of the 1st nesting level, while the “internal content” of the “Parallel” ACS submodel (see Fig. 1.10) will be transferred to the 2nd nesting level. Check that by opening the “Parallel” ACS submodel.

Transfer the To memory block to the “Copy” submodel diagram window from the Substructures library and arrange that to the right of the “Parallel” ACS submodel. Open its dialog window and change the variable name for Output\_Y.

Transfer the From memory block to the “Copy” submodel diagram window and arrange that to the left of the “Parallel” ACS submodel: the block has not yet got either an icon or a text inside it – there is only green background... Open its dialog window (double-click). Select the Input variable in the Source-list window (see Fig. 1.15) and then click the red single arrow (down): the Input variable will be transferred to the Receiver-list window (see Fig. 1.16).

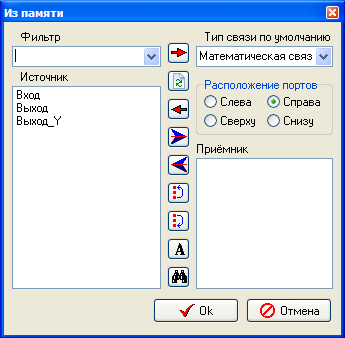


Fig. 1.15

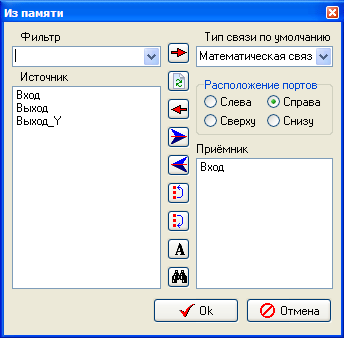


Fig. 1.16

Let us explain the meanings of command buttons (in the form of arrows) in the **From memory** block dialog window:

- red single left arrow (see Fig. 1.16) implements transfer of a selected variable from the Receiver-list window to the Source-list window;

- blue large right arrow implements transfer of all variables available in the Source‑list window to the Receiver-list window, while blue large left arrow – transfer of all variables available in the Receiver-list window to the Source-list window;

- two extreme lower buttons are intended for forced grouping of variables in the Receiver-list window (if variables >= 2).

Attention!

1. When the **From memory** block dialog window is being closed, the number of output ports on the block will be the same as the number of variables in the Receiver-list window.

2. **To memory** and **From memory** blocks implement “wireless” transfer of both scalar and vector data.

Connect the **From memory** and **To memory** blocks to the “Parallel” ACS submodel with links as it is shown in Fig. 1.17.

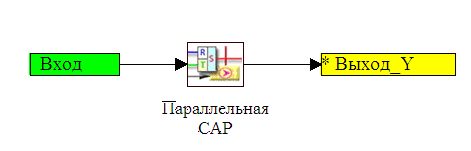


Fig. 1.17

Return to the Main Diagram Window and transfer the **From memory** block there arranging that approximately in the same place where the “Parallel” ACS submodel has been arranged before. Open the **From memory** block dialog window and transfer the Output\_Y variable to the Receiver-list window. Connect the **From memory** block output to the Curve y(t) block. The structural diagram will take an appearance similar to the one in Fig. 1.18.

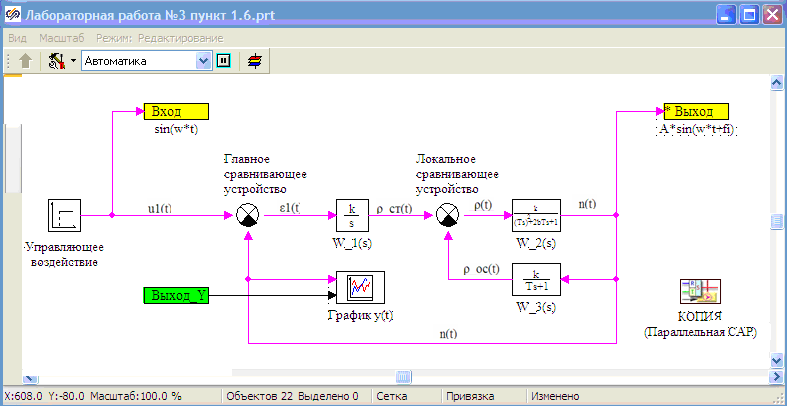


Fig. 1.18

Let us sum up the main stages of transformation of the structural diagram performed:

A new submodel under conventional name “Copy” has been created.

“Parallel” ACS submodel has been “cut” from the structural diagram of the “main” ACS and transferred to the “Copy” submodel.

Two new blocks have been transferred to the “Copy” submodel window from the Substructures library and connected with links to the “Parallel” ACS submodel; at the same time the **To memory** block is intended for “wireless” transfer of data on behavior of variable y1(t) to the Main Diagram Window (see Fig. 1.10), while the **From memory** block is intended for “wireless” reception of data on behavior of control action u(t) from the Main Diagram Window.

New **From memory** block has been transferred to the Main Diagram Window from the Substructures library and connected to the Curve y(t) block with a link. This **From memory** block implements “wireless” reception of data on behavior of variable y1(t) from the “Copy” submodel window.

As a result of changes of ACS structural diagram, in general, the “Parallel” ACS submodel receives information on control action u(t) and transmits information on behavior of variable y1(t) not by means of traditional links but using the mechanism of “wireless” data transfer.

Start up the calculation task and make sure that curves of variables y(t) and y1(t) absolutely coincide.

Now introduction to the procedures of implementation of “wireless” data transfer is completed and we can go to independent section of the laboratory work.

# 2 INDEPENDENT STUDY OF NR ACS DYNAMICS SET IN STATUS VARIABLES AND IN THE CAUCHY’S FORM

In the process of execution of laboratory work No. 1 you generated the structural diagram of the simplest mathematical model of dynamics of nuclear reactor ACS. Since the task (project) has been stored on the hard disk, open “your own” model of dynamics of NR ACS, whose structural diagram appearance (as precise as your artistic talent is) is approximately the same as the structural diagram in Fig. 2.1.

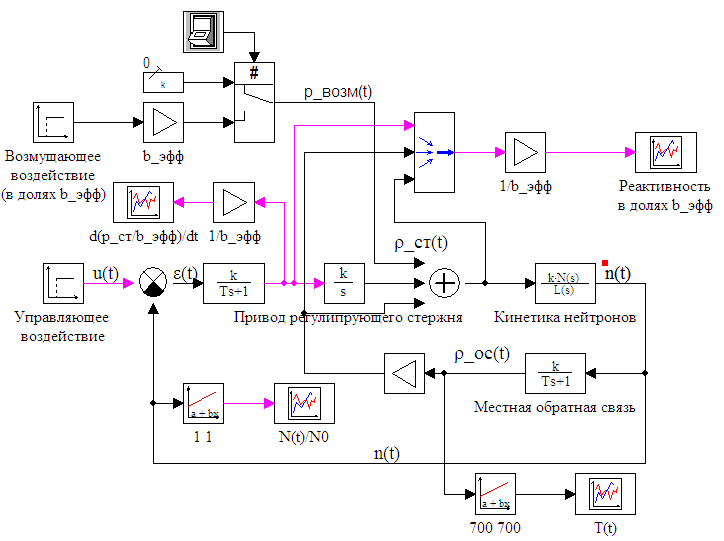


Fig. 2.1

Subscriptions under the blocks that forms transformation and display of signals (see Fig. 2.1) provide minimum information that shall help you “recall” the purpose of the target in laboratory work No. 1 and methods for its solution you have applied...

Nevertheless repeat the initial equations and relations that helped you to execute laboratory work No. 1 and that are definitely necessary for your independent studies of dynamics of NR ACS in this laboratory work.

Transfer to rated deviation of variables and further linearization of the differential equation for rated deviations of neutron density enable representation of a mathematical model of neutron pointwise kinetics with one effective group of delayed neutrons as follows:

|  |  |
| --- | --- |
|  | (2.1) |

where: – effective dose of delayed neutrons;

– prompt-neutron lifetime;

 – decay constant of precursors nuclei of delayed neutrons;

– rated deviations of concentration of precursors nuclei of delayed neutrons.

Local feedback is determined by negative thermal effect of reactivity and is described by the following equations:

|  |  |
| --- | --- |
|  | (2.2) |

where: – is reactivity temperature coefficient;

 – is stationary fuel temperature in the reactor core and rated deviation of fuel temperature in the core from the stationary value, correspondingly;

– time constant (inertia) of fuel in the core;

– nondimensional coefficient.

**Control rod drive** (see Fig. 2.1) consists of DC motor, gearbox, couplings, motion converter, control rod itself, etc. but all this elements are united into one unit to simplify the task.

Nonstationary processes in the **Control rod drive** block (unit) are described by the following differential equation:

|  |  |
| --- | --- |
|  | (2.3) |

where – speed effectiveness factor; – time constant (inertia) of **Control rod drive**.

It should be noted that although the system of equations (2.1) is reduced to the standard Cauchy’s form to include that into the complete system of equations (that describes the dynamics of all elements of NR ACS under consideration) change of reactivity  shall be expressed through its components .

Equations that describe the dynamics of local feedback require “cosmetic” edition, while the equation of the **Control rod drive**dynamics requires a transition from description in “input-output” variables to status variables.

In the independent section of this laboratory work the following stages shall be carried out for each sub-group:

1. Execute necessary transformations and record a mathematical model of dynamics of linearized ACS in status variables after calculating all elements of relevant matrixes (A, B, C and D) and vectors (which ones?!) in symbolic form.
2. Transform the ACS structural diagram (see Fig. 2.1) by adding “parallel” nuclear reactor ACS that is completely described in status variables (using the typical Status Variables block and setting matrixes A, B, C and D via the mechanism of Global Parameters in compact from).
3. Perform simulation of transitional processes in “main” ACS (described in “input-output” variables) and in “parallel” ACS (described in status variables) with control action applied , and build time relations  for the both ACS in one Graphics window and time relations  for the both ACS in the other Graphics window, correspondingly (using overlaying of curves).
4. Perform simulation of transitional processes in “main” ACS (described in “input-output” variables) and in “parallel” ACS (described in status variables) with reactivity perturbation applied , and build time relations  for the both ACS in one Graphics window and time relations for the both ACS in the other Graphics window, correspondingly (using overlaying of curves).
5. Generate a new project (task) in the SimInTech environment that describes a mathematical model of dynamics of linearized ACS in status variables with the use of typical Status Variables block and mathematical model of non-linearized ACS in the Cauchy’s form using the “Programming Language” block.

**Notes:**

* + It is advisable to implement generation of mathematical model of dynamics of linearized ACS in status variables in a new project using the copying procedures in the following sequence: first copy the content of the same window from a previous project to the **Project** (Submodel) **Global Parameters Editor** window of a new project, then copy the **Status Variables** block to the Diagram Window of the new project from the previous project.
  + Mathematical model of non-linearized ACS is obtained from equations of dynamics of linearized ACS. To that end it is necessary to “return” components omitted during linearization to the linearized equations as components of 2nd order of smallness (see lectures on the course of “Control in Technical Systems).

1. Perform simulation of a transitional process in linearized ACS and in non-linearized ACS with control action applied  and compare time relations  on one curve, and time relations  on the other curve for the both variants of mathematical models of ACS dynamics, correspondingly (using overlaying of curves).
2. Perform simulation of a transitional process in linearized ACS and in non-linearized ACS with perturbing action applied and compare time relations  on one curve, and time relations on the other curve for the both variants of mathematical models of ACS dynamics, correspondingly (using overlaying of curves).