MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY
«KHARKIV POLYTECHNIC INSTITUTE»

L.V. Asmolova, V.M. Shamardina, T.S. Chudovska

FUNDAMENTALS OF STUDYING TYPICAL DYNAMIC CONTROL ACTIONS IN MATLAB

Study Guide to Lab Classes are for students of specialty 141 «Electric Power Engineering, Electrical Engineering and Electromechanics» learning the discipline «Theory of Automatic Control. Part 1» on Educational Program in English

APPROVED by
Editorial board of the University,
Protocol № 3 from 06.10.2021

NTU «KhPI»
PromArt
2022

УДК 681.5
А 90

Рецензенти:
М. В. Аніщенко, канд.техн.наук, проф. НТУ «ХПІ»
Т. Є. Гончаренко, канд.пед.наук, доцент, зав. кафедри іноземних мов «НТУ ХПІ»
О. П. Чорний, докт.техн.наук, проф. Кременчуцького НУ ім. М. Остроградського
А. С. Маслій, канд.техн.наук, доц. УкрДУЗТ

Автори: Л. В. Асмолова, к.т.н., доцент, В. М. Шамардіна, к.т.н., доцент, Т.С. Чудовська

Asmolova L. V.
А 90 Fundamentals of Studying Typical Dynamic Control Actions in MATLAB: навч. посіб. / L. V. Asmolova, V. M. Shamardina, T. S. Chudovska – Kharkiv: PromArt, 2022. – 56 p. – англ. мова

ISBN 978-617-7944-48-4

Study guide includes materials for basic student's knowledge of time and frequency responses of typical dynamic control actions described by transfer functions. To perform tasks in lab classes MATLAB software is recommended to use. The study provides instructions on how to perform the lab classes, how tasks should be performed. Specialized icons and commands of Simulink for this purpose are described

Study Guide to Lab Classes are for students of specialty 141 «Electric Power Engineering, Electrical Engineering and Electromechanics» learning the discipline «Theory of Automatic Control. Part 1» on Educational Program in English.

Table 6 Il. 36 Bibliogr. 10 УДК 681.5

ISBN 978-617-7944-48-4

CONTENTS

INTRODUCTION

The Study Guide for Lab Classes is intended to improve the quality of teaching students how to learn complex automated control systems of technological processes and industrial mechanisms.

The first part of the Study Guide allows you to learn the basic concepts of classical theory of linear dynamic systems analysis:

- transfer functions of typical dynamic control actions;
- building and learning their time and frequency responses;
- building simple systems algoritmic block diagrams from typical dynamic control actions and research them.

Study Guide includes the material for preparation and performing of lab classes. Every description of the lab class includes the objective of the work, a theoretical part, tasks and self-control questions. To perform tasks in lab classes applying of MATLAB and Simulink software is recommended. For each lab class practical recommendations on performance of tasks and fragments of the report are provided. Details you can see in appendixes.

Lab class 1 «Basic knowledge of MATLAB and Simulink. Building simple models» is dedicated to introduction to the operational and graphical software MATLAB and Simulink software. Main commands and main principles of system modeling are overviewed.

Lab class 2 «Study of time responses of typical dynamic control actions» is dedicated to time responses of the linear dynamic control actions to standard input signals. The main tools for typical dynamic control actions analysis in time domain using Simulink are considered.

Lab class 3 «Study of the frequency response of typical dynamic control actions» is dedicated to frequency response of linear dynamic control actions. The features of writing the scripts for each typical dynamic control actions in MATLAB to build them frequency responses are considered.

Study Guide to Lab Classes are for students of specialty 141 «Electric Power Engineering, Electrical Engineering and Electromechanics» learning the discipline «Theory of Automatic Control. Part 1» on Educational Program in English.

Lab class 1

Basic knowledge of MATLAB and Simulink. Building simple models

Objective of the work:

- introduction to MATLAB/Simulink;
- learning the base and getting skills in Simulink;
- building of simple models in MATLAB.

1.1 Theoretical part

MATLAB is a programming and numerical computing platform for data analysis, develop algorithms and create models.

Simulink is MATLAB application for modelling dynamic systems. You can create models using block diagrams from the standard block library and perform calculations. During simulation, you can observe transient responses that occur in the system. Special observation devices, which are included in the Simulink standard block library, are used for this purpose. The results of the simulation can be presented in the form of graphs or tables.

Run Simulink

To run Simulink, MATLAB must be started first. To start it, double left-click on the MATLAB icon. When MATLAB is started, the MATLAB Command Window opens as shown in Figure 1.1.

When MATLAB Command Window is open, you need to run Simulink.

There are three ways to run Simulink:

– click **Simulink** button  on MATLAB toolbox (number 1 in Figure 1.1);

– type *Simulink* on your keyboard in MATLAB Command Window and press **Enter** (number 2 in Figure 1.1);

– click **Open** on the File menu (number 3 in Figure 1.1) and open a model file with the extendent **.mdl**, for example, *my_lab.mdl*.

After starting Simulink, the **Simulink Library Browser window** opens and shows the list of standart block libraries in hierarchical structure. The Simulink Library Browser window is shown in Figure 1.2.

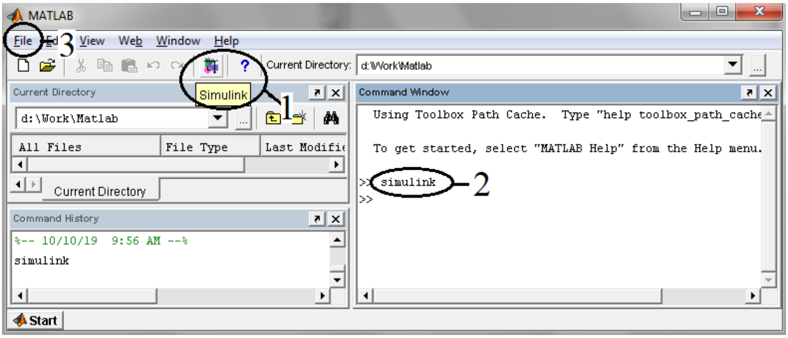


Figure 1.1 – MATLAB Command Window

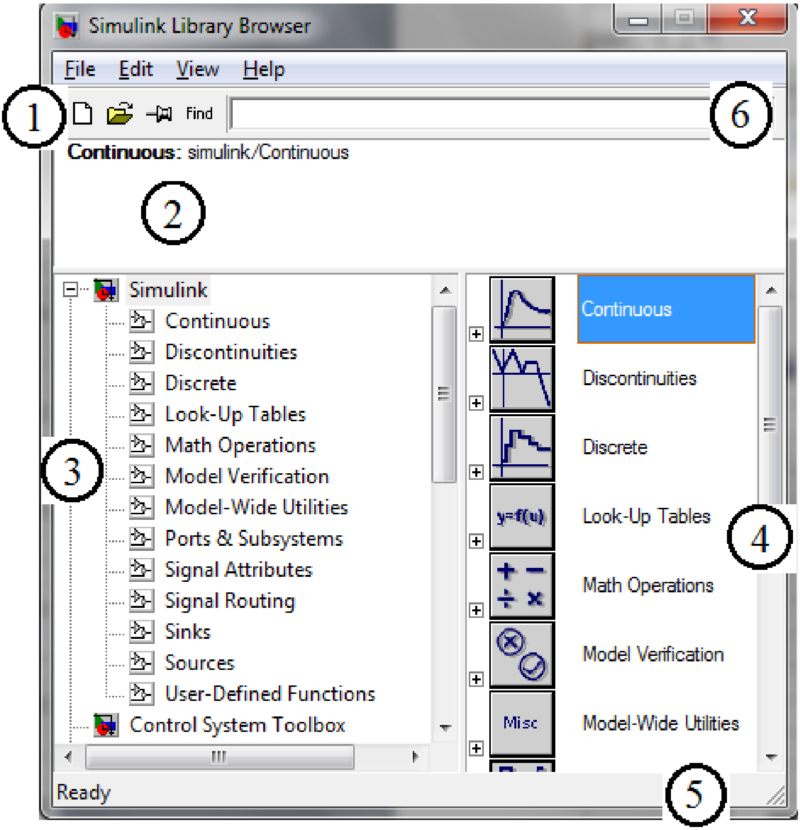


Figure 1.2 – Simulink Library Browser Window

Figure 1.2 uses the following notations:

1 – MATLAB Command Window toolbox with command buttons is the most commonly used;

2 – Comment window to display an explanatory comment about the selected library subpart or block;

3 – list of library blocks:

- **Continuous** – Model linear functions;
- **Discontinuities** – Model nonlinear functions;
- **Discrete** – Model discrete functions;
- **Look-Up-Table** – Approximation and interpolation blocks;
- **Math Operations –** Model mathematical and logical functions;
- **Model Verification –** Checking the adequacy of the model;
- **Model-Wide Utilities –** Utilities (tuning) of the model as a whole;
- **Port & Subsystems –** Subsystem input (output);
- **Signal Attributes –** Properties of signals;
- **Signal Routing –** Distribution of signals;
- **Sinks –** Display or export output;
- **Sources –** Generate or import system inputs;

4 – list of library subparts or nested blocks;

5 – status bar that contains a prompt for the taken action;

6 – address bar to find a block icon

To find the specific block, you can use automatic search by name. Write down the name in address bar to find a block icon 6 (some letters are possible) and press **Enter**. After that, the automatic search block starts. If the block is found, it is displayed in the list of nested blocks 4 as selected.

Create a new file and Simulink model

To creat a new block diagram in the Simulink, you should open a window Simulink Editor. From the **Simulink Library Browser menu**, select **File** → **New** → **Model** or click ***Create a new model***  on the Simulink toolbox. It opens an **Untitled** window (highlighted by the ellipse in Figure 1.3), where the blocks from the **Simulink Library Browser subpart** are moved, making up the block diagram.

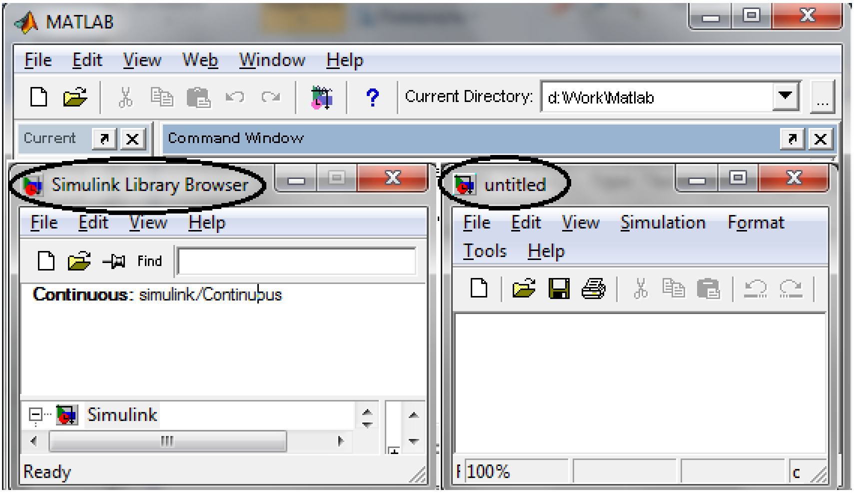


Figure 1.3 – Untitled Window from the Simulink Library Browser, where making up the block diagram.

An existing Simulink model from the Simulink Library Browser menu is opened with **File** → **Open** or by clicking ***Open a model***  on the Simulink toolbox. Select the model file that you want to open and then click Open. The selected model opens in the Simulink Editor.

To build a model, need by copying or draging blocks from the Simulink Library Browser to the Simulink Editor. To place blocks in the Untitled window, open the specific subpart of the Simulink Library Browser, then select the specific block and drag the block from the Library Browser to the Simulink Editor model window.

Figure 1.4 shows that Sources library is selected in Simulink Library Browser in the left pane (number 1 in Figure 1.4). The right pane of Simulink Library Browser displays icons from the selected library. Select the Clock icon (number 2 in Figure 1.4). Place the cursor on it, press and hold down the left mouse button and move this icon to the created Untitled window. Then release the left mouse button. Therefore, the Clock as the block has moved to the Untitled window (shown by the arrow in Figure 1.4)

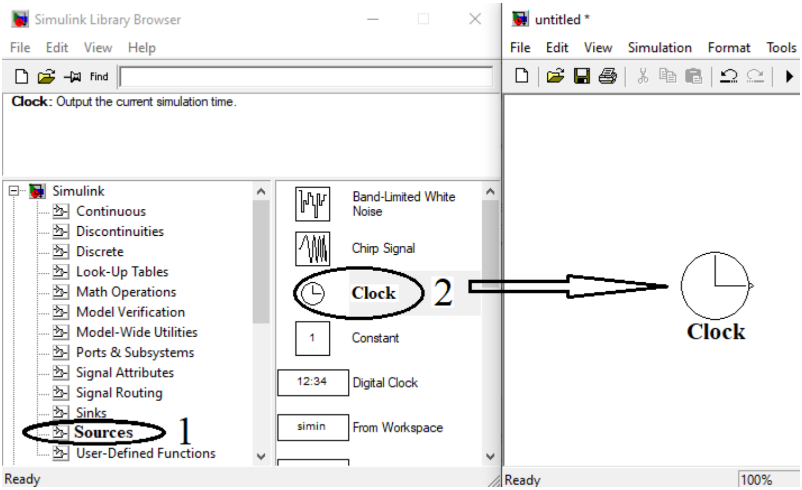


Figure 1.4 – Moving the Clock as the block from the Simulink Library Browser to the Untitled Window

Add other blocks to your model using the same way as for adding the Clock block. Now you have the blocks to build a simple model.

Blocks in the Untitled Window can be copied. To do this, highlight this block, press and hold the left mouse button, and then drag this block to where it will be in the Untitled Window as shown in Figure 1.5.

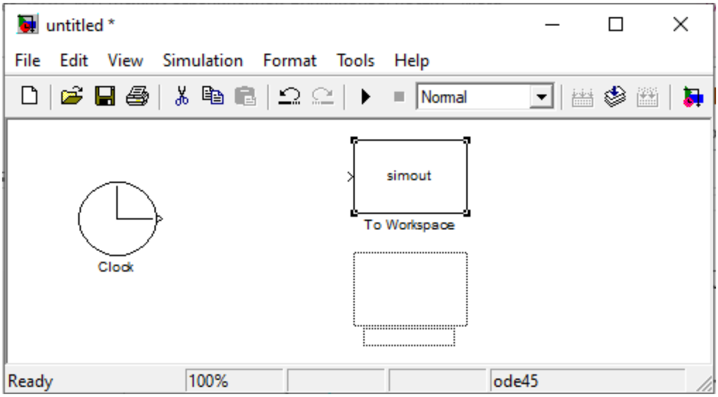


Figure 1.5 – Adding and copying blocks from Simulink Library Browser to Untitled Window

A number is added to the new block name. After copying, all the blocks can be spreading all over the Untitled Window. Therefore, they should be located in such that they can be easily connected, as shown in Figure 1.6.

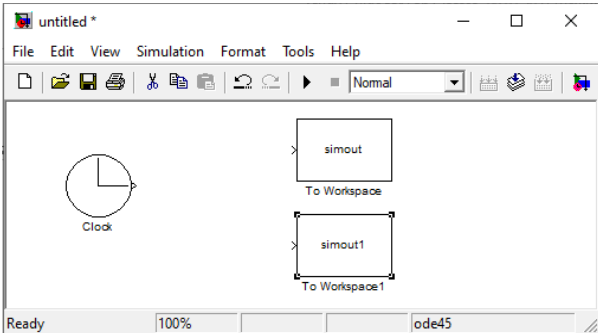


Figure 1.6 – Location of blocks to connect them in an Untitled Window

To connect the blocks together, need to draw connecting lines between the output and input ports. The input and output ports of some blocks are shown in Figure 1.7: the Clock block has an output port and no input port, the To Workspase block has an input port and no output port, the Transfer Fcn block has an output and input ports.

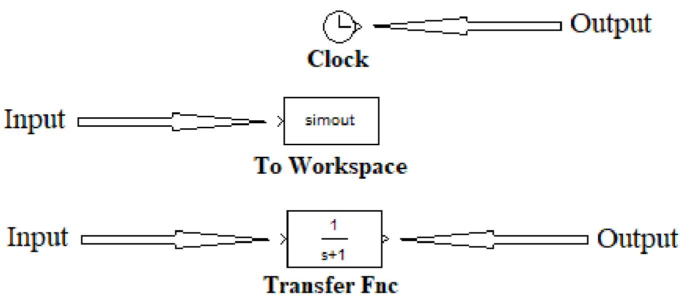
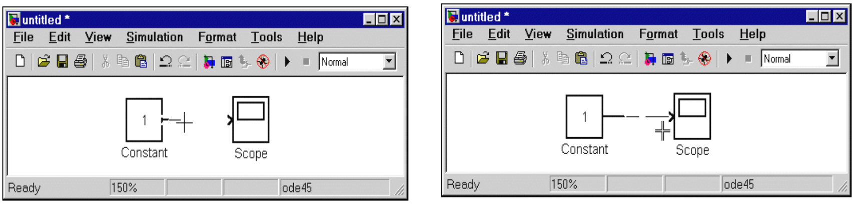


Figure 1.7 – The input/output identification of some blocks

There are two ways to connect blocks:

- manually;
- automatically.

An example of a manual connection of two blocks together is shown in figure 1.8.



a *b*

Figure 1.8 – Connecting the two blocks manually

To do this, place the cursor on the output port of the first block (Constant block). The cursor changes to a cross (+) (Figure 1.8,a). Press and hold the left mouse button and drag the cross from the output port of the Constant block to the input port of the next block (Scope block). While you hold down the mouse button, the connecting line is shown as a red dashed arrow. When the cross is over the output port release the left mouse button (Figure 1.8,b). If the connecting line is a solid black arrow, it means that the blocks are connected. If the connecting line is a dashed red line, it means that the blocks are not connected.

To connect the two blocks together automatically, place the cursor on the first block (Constant block) and right-click. Black markers will appear in the corners of this block. Next, press and hold down the **Ctrl** key, position the cursor on the second block (Scope block) and right-click. A connecting line will draw.

Figure 1.9 shows an example of a manual connection of three blocks. To connect three or more blocks, you must first connect two blocks together.

To connect the input port of the third block (Display block) to existing connecting line, a branch point must be created. To create a branch point in a connecting line (i.e. a take-off point), need to place the cursor where you want to

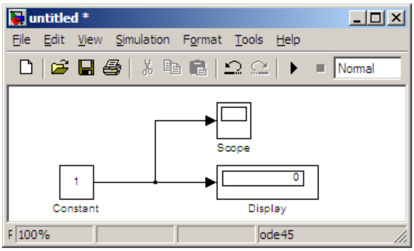


Figure 1.9 – Connecting the three blocks manually

start a branch line. Click, and then drag the cursor away from the line to form a dotted-red line. Next, drag the cursor to the anothe input port (Display block), and then release the mouse button.

So, Simulink connects all the blocks togather with connecting lines. They transmit signals in the direction indicated by the arrow. The lines must always transmit signals from the output port of one block to the input port of another block. On exception to this is a line may branch point from another line, splitting the signal into two blocks.

Setting and changing datas

When the blocks are connected, need to set the data for each block. Double left-click the block icon opens the Block Parameters dialog box (highlighted by the ellipse in Figure 1.10).

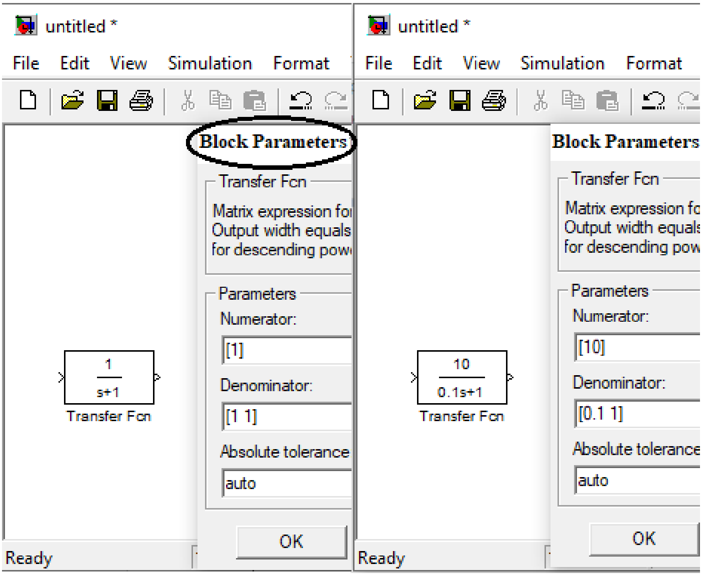


Figure 1.10 – Setting and changing datas in the Block Parameters TransferFcn

If you enter data as a letter, the numeric values need to set in the MATLAB Command Window or in the script before you start the simulation. If you enter the data as a decimal, the whole number part and fractional part are separated by a decimal point, not a comma. Then close the Block Parameters window by pressing **OK** or **Apply**.

Selecting model components

To perform any action with a model components (block, connecting line, inscription), them must be selected. To do this, place the cursor on the model component you want to select and right-click. Black markers will appear in the corners of the model component.

To select some model components, place the cursor near the group of model components. Press and hold down the right mouse button, drag it. A dotted line will appear and resize as you move the mouse. All framed model components become selected. You can also select some model components with the **Edit → Select All** command.

A selected model component can be copied, moved to the clipboard, pasted from the clipboard and deleted.

Copy and move model components to the clipboard

To copy a model component, you need to select it. Then left-click and hold down the left mouse button and drag it. A copy of the model component is created, which you can drag to the place you want.

To copy the block to the clipboard, you need to select it, click the **Edit → Copy** from the menu or by clicking *Copy*  button on the toolbox.

To copy the model to the clipboard, select it and using the **Edit → Copy model to Clipboard** move the model image on the Windows clipboard and make it available to other applications.

Paste blocks and Simulink model from the clipboard

To paste a block or Simulink model from the clipboard, left-click any place in Untitled Wimdow. Click the **Edit → Paste** command from the menu or by clicking *Paste*  button on the toolbox.

Delete model components

To delete model components, you need to select them, click the **Edit →**

Clear command from the menu or by clicking Delete on the keyboard.

Pay attantion that the **Clear** command deletes the model components without moving them to the clipboard.

Resize blocks

To resize a block, you should select it. Place the cursor on one of the black markers in the corners of the block.The cursor changes to a double-sided arrow, left-click and expand (or compress) the block image, as shown in Figure 1.11.

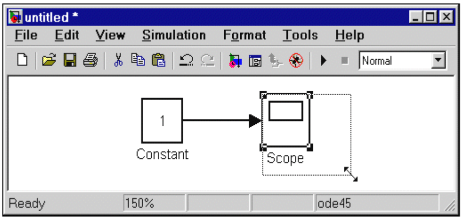


Figure 1.11 – Resizing blocks

Move blocks

To move block, you need to select it, click and hold down the left mouse button. Connecting lines between the input port of one block and the output port of another block can decrease or increase in length, but not break.

It is also can insert any block into a connecting line between already connected blocks. To do this, select the block, click and hold down the right mouse button. Drag this block to the desired location on the connecting line.

Setting Simulation Parameters and run Simulation

Before you run Simulation the behavior of a block diagram, set the simulation parameters. Click the **Simulation** → **Simulation Parameters…** command from the menu Untitled Window. It opens the **Simulation parameters: Untitled** window to the *Solve pane*, as shown in Figure 1.12, where you can set the type of numerical solver, start and stop times, and maximum step size.

The simulation time is set in the *Start time* and *Stop time* fields. In the Start time field, enter zero, and in the Stop time field, enter a user-defined number depending on task conditions.

In the *Type* field need to select: the integration step as **variable-step** and integration method as **ode45 (Dormand-Prince)**. These parametrs use for the continuous-time systems simulation, which are considered in the disciplin «Theory of automatic control. Part 1».

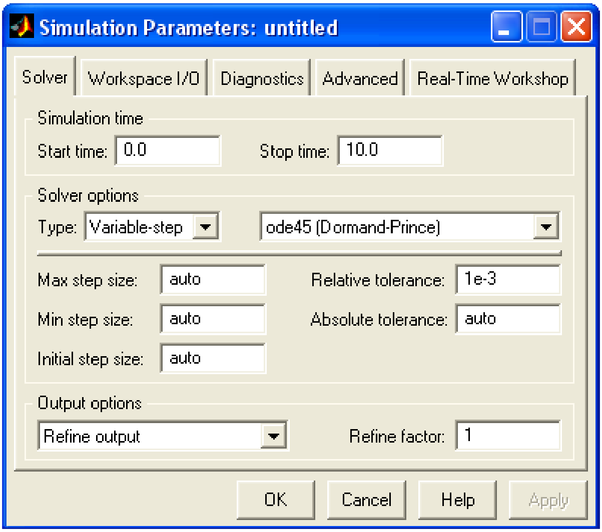


Figure 1.12 – Simulation Parameters Window

In the *Max step size* field, set the maximum integration step Δt_{int} . The default setting is **auto**. If this is the case, the transient responses will be as broken lines. To make the transient responses smooth, the integration step need to calculat by the formula $\Delta t_{\text{int}} \leq \dfrac{T_{\min}}{10}$, where $T_{\min}$ is the minimum time constant of the system. Then close the Simulation Parameters Window dialog by pressing **OK** or **Apply**.

After setting Model Simulation Parameters, you can run the simulation your model and study its behaviour. To do this, click the **Simulation → Simulation Start** command from the menu Untitled Window or by clicking *Start Simuation*  button on the toolbox. The simulation runs, and then stops when it reaches the stop time specified in the Model Simulation Parameters dialog box.

Save the block diadram and script as a file

The created block diagram and script need to save to disk as a file. To do this, click the **File → Save As...** command from the menu. The **Save As...** menu is opened, where you need to select the folder and write the file name. If you want to save the block diagram, the file will have the extension «**.mdl**». If you want to save the script, the file will have the extension «**.m**».

The file name must starts with a letter and cannot exceed 32 symbols. The file name cannot contain Cyrillic and other special symbols.

If you want to save the file again, use the **File → Save** command from the menu.

1.2 Tasks

1. Study Simulink and basic concepts.
2. Run Simulink.
3. Create the Simulink model named Untitled.
4. Open the Simulink Library Browser.
5. Build the block diagram and locate its blocks as shown in Figure 1.13

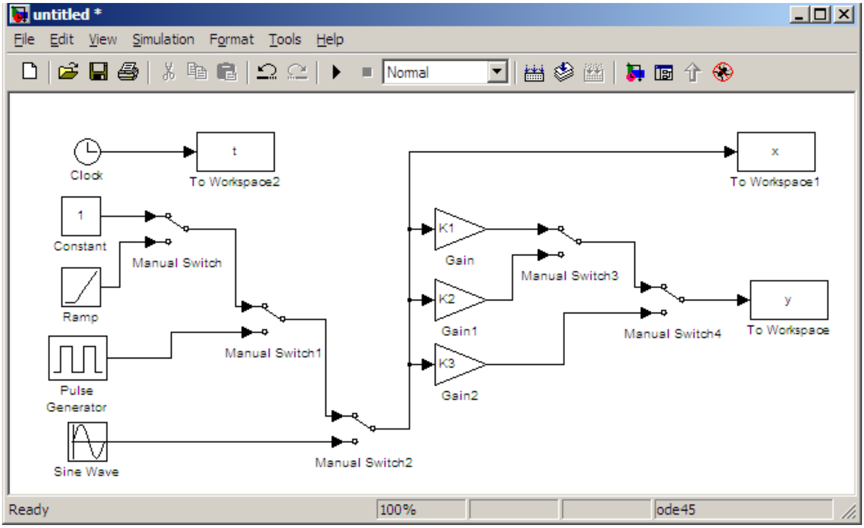


Figure 1.13 – Block diagram for studying the Gain block with different input signals

5.1. To build the block diagram, first open the **Simulink/Sources** library. Find and move the input signal icons from the Simulink Library Browser to the created Untitled Window, as shown in Figure 1.4 and set block data in the

Block Parameters, as shown in Figure 1.10.

The input signals applied to the Gain block to study its behavior from Sources library:

– the **Clock** block (time signal) outputs the current simulation time at each simulation step. The Clock block icon is shown in Figure 1.14;

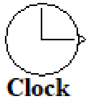
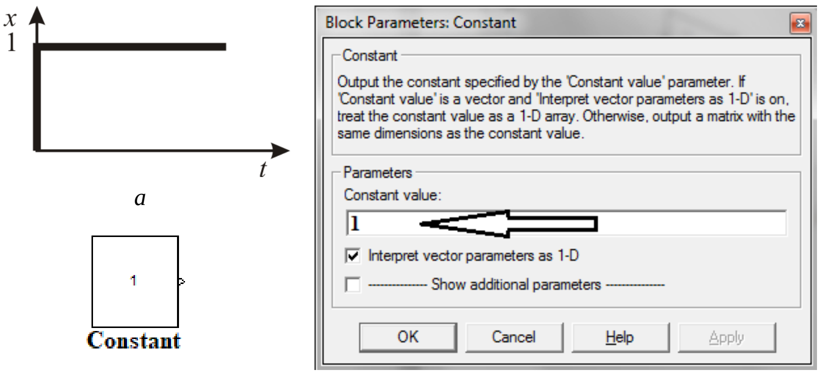


Figure 1.14 – Clock block icon

– the input signal, the **unit step function** increases instantly from zero to one and remains unchanged, as shown in Figure 1.15,*a*.

The unit step function is described by the equation $x(t) = 1(t)$ and is set by the Constant block. The Constant block icon is shown in Figure 1.15,*b*.

Enter 1 in the *Constant value* field of the **Block Parameters: Constant**, as shown in Figure 1.15,*c*:



a *b* *c*

Figure 1.15 – Unit step function, its icon as the Constant block and setting data in the Constant block

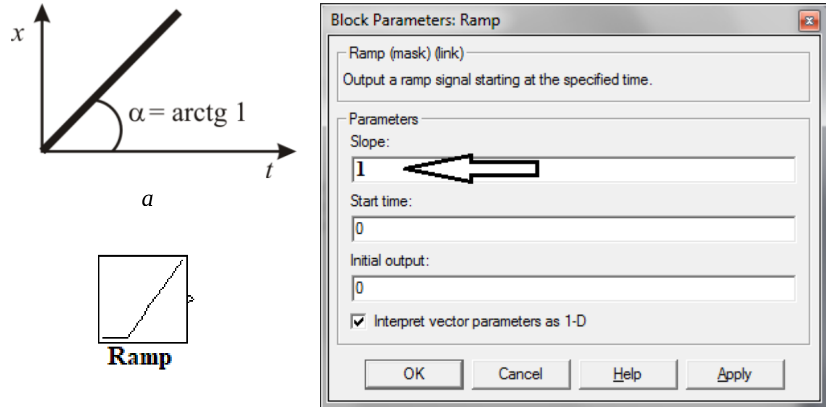
– the input signal, the **linearly increasing signal** starts at a specified time and changes by a specified rate, as shown in Figure 1.16,*a*.

The linearly increasing signal is described by the equation $x(t) = 1 \cdot t$ and is set by the Ramp block. Ramp block icon is shown in Figure 1.16,*b*.

Set the data in the **Block Parameters: Ramp** as shown in Figure 1.16,*c*:

- the rate of change of the output signal in the *Slope* field, enter 1. This corresponds to the angle of the signal to the abscissa axis with an angle of 45°;

- the start time in the *Start time* field, enter 0;
- the initial output signal level in the *Initial output*, enter 0:

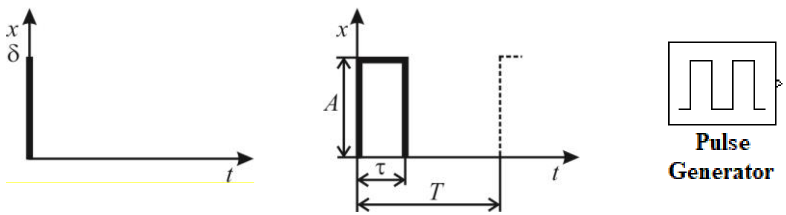


a *b* *c*

Figure 1.16 – Linearly increasing signal, its icon as the Ramp block and setting data in the Ramp block

– the input signal, the **unit impulse function** is a rectangle with an area equal to one, S = 1. The amplitude of the rectangle tends to infinity, $A \to +\infty$, the width its tends to zero, $\tau \to 0$, the period its tends to infinity, $T \to +\infty$, as shown in Figure 1.17,*a*.

The unit impulse function is described by the equation $x(t) = \delta(t)$ and is set by the Pulse Generator block. The Pulse Generator block generates square wave pulses at regular intervals, as shown in Figure 1.17,*b*. Its icon is shown in Figure 1.17,*c*.



a *b* *c*

Figure 1.17 – Unit impulse function and its icon as the Pulse Generator block

Set the data in the **Block Parameters: Pulse Generator**, as shown in Figure 1.18:

- in the *Amplitude* field, enter 100 or another value;
- in the *Period* (secs) field, enter 100 or another value;
- in the *Pulse Width* (% of period) field, enter the number calculated using the formula:

$$\boldsymbol{Pulse\ Width}(\%) = \frac{\tau \cdot 100\%}{T},$$

where $\tau = \frac{1}{A}$ – is pulse width,

T – is period.

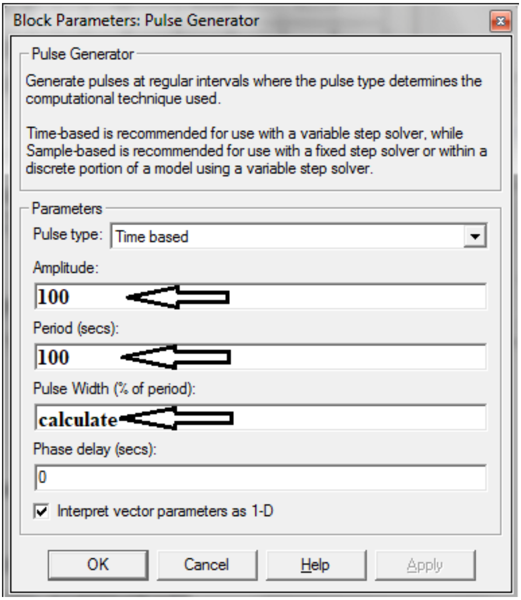
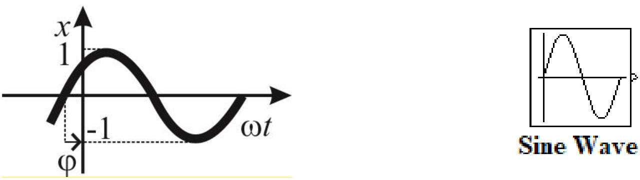


Figure 1.18 – Setting data in the Pulse Generator block

- *Phase delay* ***(secs)*** is set to 0 in this laboratory work.

– the input signal, the **sinusoidal signal**, whose wave amplitude is one, $A = 1$, angular frequency is one, $\omega = 1$, initial phase is zero, $\varphi_0 = 0$, and wave offset is zero, $\varphi = 0$, as shown in Figure 1.19,*a*.

The sinusoidal signal is described by the equation $x(t) = 1 \cdot \sin(\omega t + \varphi)$ and is set by the Sine Wave block. Sine Wave block icon is shown in Figure 1.19,*b*.



a *b*

Figure 1.19 – Sinusoidal signal and its icon as the Sine Wave block

Set the data in the **Block Parameters: Sine Wave**, as shown in Figure 1.20:

- in the *Amplitude* field, enter 1;
- in the *Bias* (wave offset φ) field, enter 0;
- in the *Frequency* (rad/s) (angular frequency ω), enter 1;
- in the *Phase* (rad) (initial phase φ_0), enter 0.

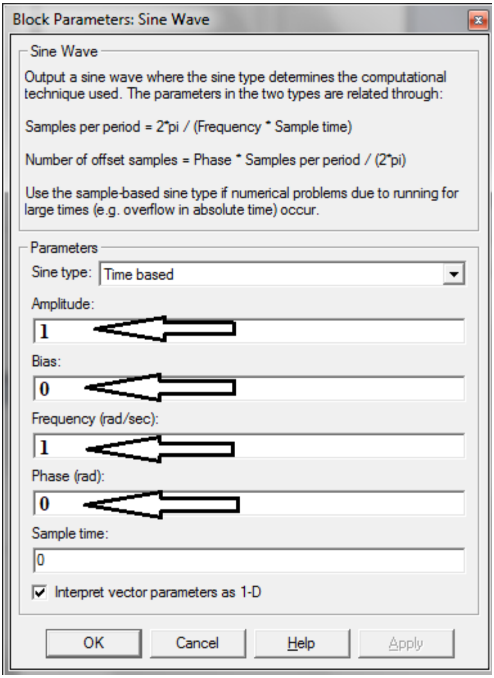


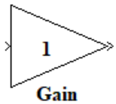
Figure 1.20 – Setting data in the Ramp block

5.2. The Gain block you are studying is described by the transfer function *K*. It can be found in the **Simulink/Math Operations** library. Gain block icon is shown in Figure 1.21,*a*.

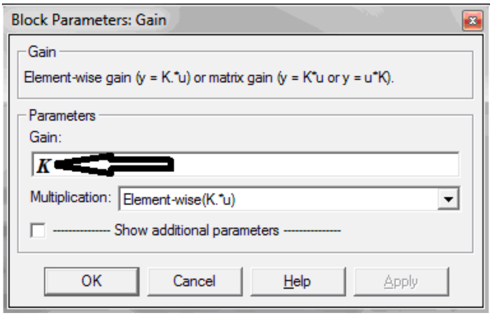
In the lab class, the transfer function *K* is set in three variants:

- $K_1 = \frac{N}{2}$, where *N* is the variant number corresponding to the number in the list of group students;
- $K_2 = N$;
- $K_3 = 2 \cdot N$.

Enter *K* in the *Gain* field of the **Block Parameters: Gain**, as shown in Figure 1.21,*b*, for the corresponding Gain block in Figure 1.13.



a



b

Figure 1.21 – Gain block icon and setting data in it

So, the block diagram you are studying has four input signals and three Gain blocks with different transfer functions *K*. In the lab class you should study the response of each Gain block to the given input signals one after the other. For this, select one of the four input signals and one of the three Gain blocks and connect them together through the Manual Switch block.

5.3. The Manual Switch block can be found in the **Simulink/Signal Routing** library. Manual Switch block icon is shown in Figure 1.22.

Find and move the Manual Switch block icon from the Simulink Library Browser to the created Untitled Window. Connect all the blocks together with connecting lines between the output and input ports, as shown in Figure 1.13.

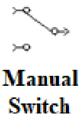


Figure 1.22 – Manual Switch block icon

To toggle between the selected input signals and the selected Gain blocks, double-click on the Manual Switch block.

So, you indicate which input signal feeds the Gain block's input. The study results are written to MATLAB workspace using the To Workspace block.

5.4. The To Workspace block can be found in the **Simulink/Sinks** library. To Workspace block icon is shown in Figure 1.23.

Find and move the To Workspace block icon from the Simulink Library Browser to the created Untitled Window. Connect the To Workspace blocks together with the connection lines between the output ports of the Clock block and the Manual Switch block, as shown in Figure 1.13.

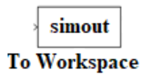


Figure 1.23 – To Workspace block icon

Set the data in the **Block Parameters: To Workspace**, as shown in Figure 1.24:

- in the *Variable name* field, enter the name of the variable whose data is saved to the clipboard. In the lab class, the block diagram in Figure 1.13 has:
  - input signal name is *x*. Therefore, in the *Variable name* field, enter *x*;
  - output signal name is *y*. Therefore, in the *Variable name* field, enter *y*;

– the time signal name is *t*. Therefore, in the *Variable name* field, enter *t*;

- in the *Save format* field, select *Array*. In Array format, the data is saved as an array in which the number of rows is determined by the number of calculated points in time and the number of columns by the dimensionality of the vector that is fed to the block input. If the input is a scalar signal, the array will contain only one column.

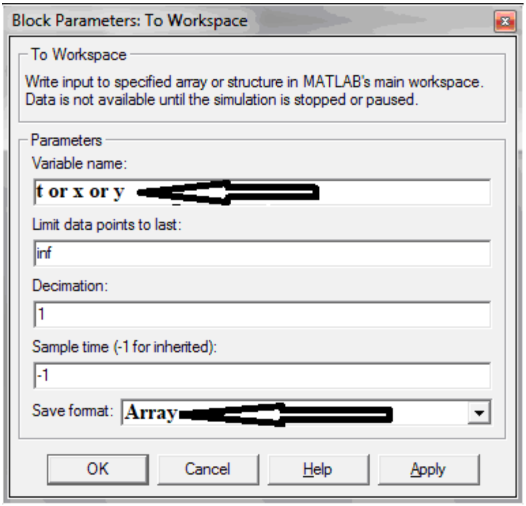


Figure 1.24 – Setting data in the To Workspace Block

So, the block diagram in Figure 1.13 is created.

6. Copy the created block diagram to the clipboard as described in the **Theoretical part: Copy and move model components to the clipboard**.

7. Open a Word Document and paste the created block diagram from the clipboard using the **Edit** → **Paste** command from the menu or by clicking *Paste*

 button on the toolbox in Word Document.

8. Set the simulation parameters as described in the **Theoretical part: Setting Simulation Parameters and running simulation**.

9. Save the created block diagram as described in the **Theoretical part: Save the block diagram and script as a file**.

After saving the block diagram, study the behaviour of the Gain block with different values of transfer ratios *K* to the given input signals $x(t)$:

– the unit step function;

– the linearly increasing signal;

– the unit impulse function;

– the sinusoidal signal.

10. Study the behaviour of the Gain block with transfer ratios K_1, K_2 and K_3 for each input signal type $x(t)$ separately. For this, switch the Gain blocks

one after the other using the Manual Switch block.

Run simulation after each Gain block switch by clicking *Start Simuation* button on the toolbox.

11. Plot the transient responses of the input $x(t)$ and output $y(t)$ signals in the same coordinate axes. For this, in the MATLAB window, type the **plot** and **grid on** commands as shown below:

```
>> plot(t,x,'k-',t,y,'r-');grid on
```

As a result, the plot command creates a 2-D line plot in the input $x(t)$ and output $y(t)$ signals data versus the corresponding values in t, using the specified line style and colour, in MATLAB window called Figure 1.

In example, the command plot(t,x,'k-',t,y,'r-') is set so that the transient response of the input signal $x(t)$ is represented by a solid black line, and the transient response of the output signal $y(t)$ by a dashed black line. You can select the line style and colour yourself, as described in Appendix A.1.

To plot the all transient output responses of the Gain blocks $y(t)$ when studying with the one input signal type $x(t)$ together in Figure 1, you must type the **hold on** command and repeat the **plot** command again:

```
>> hold on
>> plot(t,x,'k-',t,y,'b-');grid on
```

12. Label the transient responses of the input $x(t)$ and output $y(t)$ signals, the X and Y axis using and buttons on the Figure toolbox as shown in Figure A.1 of the Appendix A.2.

For example: for the block diagram in Figure B.1 of the Appendix B shows simulation results (Figure B.2 of the Appendix B) when a unit step function is fed to the input.

By analysing the results, you can see how this model works: an input signal equal to one, $x = 1$, is fed from the output of the Constant block to the input of the Gain block with transfer ratios $K_1 = 25$, $K_2 = 50$ and $K_3 = 100$. The input signal equal to one is multiplied by the corresponding value of transfer ratio K. If $K_2 = 50$, so the output signal will be equal to $y = 1 \cdot 50 = 50$.

Figure B.2 shows the *X* and *Y* axes and the transient responses of the input *x*(*t*) and output *y*(*t*) signals:

– the transient response of the input signal *x*(*t*) is represented by the black solid line and described by the equation *x*(*t*) = 1(*t*);

– the transient responses of the output signals *y*(*t*) are represented by the black dashed lines and described by corresponding equations *y*(*t*) = 25, *y*(*t*) = 50 and *y*(*t*) = 100.

– *X*-axis describes time *t*;

– *Y*-axis describes input *x* and output *y* signals.

13. Copy the Figure to the clipboard by clicking **Edit → Copy Figure**.

14. Open a Word Document again and paste the Figure from the clipboard using the **Edit → Paste** command from the menu or by clicking *Paste*  button on the toolbox in Word Document.

15. Make a report (see Appendix B.1) and save it on disk or USB.

16. Get the mark for your lab class from your teacher by answer-ing his/her control questions.

1.3 Self-control questions

1. What is Simulink?
2. What subparts of the Simulink Library Browser are used in the lab class?
3. What blocks does the block diagram consist in the lab class?
4. How do you set the parameters for the blocks used in the lab class?
5. Describe the operations on the blocks.
6. What are the functions of the blocks in the Simulink/Sources library?
7. How do you set the parameters for the blocks used in the Simulink/Sources library in the lab class?
8. What do the **plot**, **grid on** and **hold on** commands mean?
9. How is the created block diagram and simulation results moved to a Word document?
10. Name and write down the equation of the input signals.

Lab class 2

Study of time responses of typical dynamic control actions

Objective of the work:

– getting skills in creating block diagrams of typical dynamic control actions in Simulink;

– studying the time responses of typical dynamic control actions;

– studying the effect of changing the parameters of typical dynamic control actions transfer functions on the behavior of their time responses in MATLAB/Simulink.

2.1 Theoretical part

Typical dynamic control actions are control actions that are described by linear ordinary differential equations of the first or second order.

The **differential equation of the first-order typical dynamic control actions** looks as:

$$T\frac{dy}{dt}+y(t)=K\cdot\left(T_0\frac{dx}{dt}+x(t)\right), \qquad (2.1)$$

where $x(t)$ – is the input signal;

$y(t)$ – is the output signal;

T – is the derivative coefficient of the output signal $y(t)$;

T_0 – is the derivative coefficient of the input signal $x(t)$;

K – is the transfer ratio of the typical dynamic control action.

The **differential equation of the second-order typical dynamic control actions** looks as:

$$T^2\frac{d^2y}{dt^2}+2\xi T\frac{dy}{dt}+y(t)=K\cdot\left(T_0^2\frac{d^2x}{dt^2}+2\xi_0 T_0\frac{dx}{dt}+x(t)\right), \qquad (2.2)$$

where ξ_0, ξ – are the damping ratio, which shows how quickly the oscillation process damps in the input and output signals respectively.

These linear ordinary differential equations describe the dynamic behaviour of typical dynamic control actions. Linear ordinary differential equations can be solved using the Laplace transform method. To do this, all derivatives of the differential equation are replaced by a differential operator or a Laplace operator s ($s = \frac{d}{dt}$) and the initial conditions are assumed to be zero.

As a result, we obtain the **transfer function** of the typical dynamic control action as the ratio of the Laplace transform of the output signal $y(s)$ to the Laplace transform of the input signal $x(s)$ with the initial conditions as zero: $W(s) = \frac{y(s)}{x(s)}$.

The basic typical dynamic control actions and their transfer functions $W(s)$ are shown in Table 2.1.

Table 2.1 – Transfer functions of typical first- and second-order dynamic control actions

Name the dynamic control action	Transfer function of the dynamic control action
the proportional control action (P)	$W(s) = K$
the integral control action (I)	$W(s) = \frac{K}{s} = \frac{1}{T \cdot s}$
the derivative control action (D)	$W(s) = T \cdot s$
the first-order aperiodic control action (A-1)	$W(s) = \frac{K}{T \cdot s + 1}$
the vibration control action (V)	$W(s) = \frac{K}{T^2 \cdot s^2 + 2 \cdot \xi \cdot T \cdot s + 1}$, where $\xi < 1$
the conservative control action (Con)	$W(s) = \frac{K}{T^2 \cdot s^2 + 1}$, where $\xi = 0$
the second-order aperiodic control action (A-2)	$W(s) = \frac{K}{T^2 \cdot s^2 + 2 \cdot \xi_1 \cdot T \cdot s + 1} =$ $= \frac{K}{(T_1 \cdot s + 1) \cdot (T_2 \cdot s + 1)}$, where $\xi_1 \geq 1$

Using the transfer function, you can get a transient at the output of the typical dynamic control action when input signals are fed to it. This allows you to evaluate the behavior of the typical dynamic control action in the time domain.

In the lab class you will study dynamic properties of the typical dynamic control actions when the following input signals are fed to them:

– the unit step function $x(t) = 1(t)$ (see Figure 1.15);

– the unit impulse function $x(t) = \delta(t)$ (see Figure 1.17).

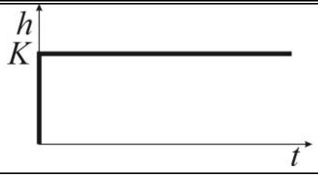
The dependence of the change in the output of the typical dynamic control action on time t, when a unit step function $x(t) = 1(t)$ is fed to its input at zero initial conditions, is called the **transient response** $h(t)$.

The transient response $h(t)$ is defined as the inverse Laplace transform. A function of a variable t is called an «original», and its Laplace transform is called an «image». The image of input signal $L[x(t)] = X(s) = \frac{1}{s}$, the image of output signal $L[y(t)] = L[h(t)] = Y(s) = W(s) \cdot X(s) = \frac{W(s)}{s}$. So, the transient response looks as:

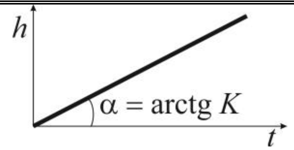
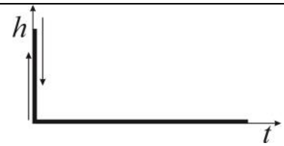
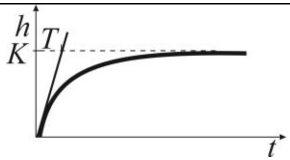
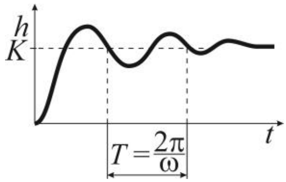
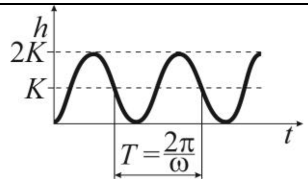
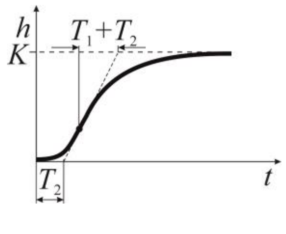
$$h(t) = L^{-1}\left\{\frac{W(s)}{s}\right\}. \qquad (2.3)$$

The transient responses of the basic typical dynamic control actions $h(t)$ are shown in Table 2.2.

Table 2.2 – Transient responses of typical first- and second-order dynamic control actions

Name the dynamic control action and transient response equation	Transient response of the dynamic control action
the proportional control action (P) $h(t) = K \cdot 1(t)$	

Continuation of the Table 2.2

Name the dynamic control action and transient response equation	Transient response of the dynamic control action
the integral control action (I) $h(t) = K \cdot t$	
the derivative control action (D) $h(t) = T \cdot \delta(t)$	
the first-order aperiodic control action (A-1) $h(t) = K \cdot \left(1 - e^{-\frac{t}{T}}\right)$	
the vibration control action (V) $h(t) = K \cdot \left[1 - e^{-\lambda t} \cdot \left(\cos\omega t - \frac{\lambda}{\omega} \cdot \sin\omega t\right)\right]$ wh ere $\omega = \frac{\sqrt{1-\xi^2}}{T}$, $\lambda = -\frac{\xi}{T}$	
the conservative control action (Con) $h(t) = K \cdot \left[1 - \cos\left(\frac{\sqrt{1-\xi^2}}{T} \cdot t\right)\right]$	
the second-order aperiodic control action (A-2) $h(t) = K - \frac{K \cdot T_1}{T_1 - T_2} \cdot e^{-\frac{t}{T_1}} + \frac{K \cdot T_2}{T_1 - T_2} \cdot e^{-\frac{t}{T_2}}$, where $T_1 = \frac{T}{\xi - \sqrt{\xi^2 - 1}}$, $T_2 = \frac{T}{\xi + \sqrt{\xi^2 - 1}}$	

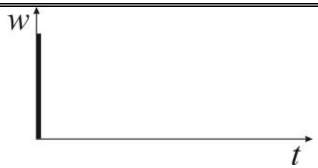
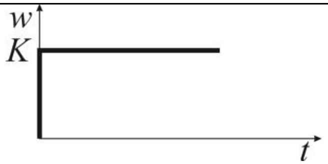
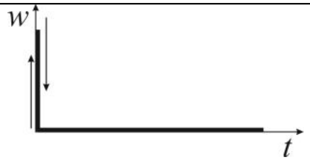
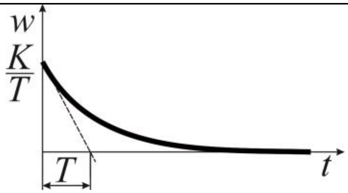
The dependence of the change in the output signal of the typical dynamic control action on time *t*, when a unit impuls function $x(t) = \delta(t)$ is fed to its input, is called the **impulse transient responses** $w(t)$.

The impulse transient response $w(t)$ is defined as the inverse Laplace transform. The image of input signal $L[\delta(t)] = X(s) = 1$, the image of output signal $L[y(t)] = L[w(t)] = Y(s) = W(s) \cdot X(s) = W(s)$. So, the transient response looks as:

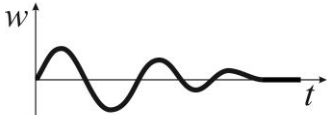
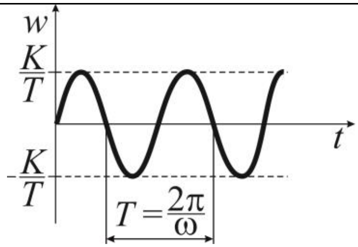
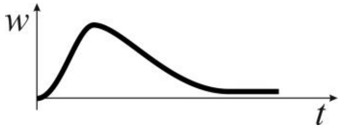
$$w(t) = L^{-1}\{W(s)\}. \qquad (2.4)$$

The impulse transient responses of the basic typical dynamic control actions $w(t)$ are shown in Table 2.3.

Table 2.3 – Impulse transient responses of typical first- and second-order dynamic control actions

Name the dynamic control action and impulse transient response equation	Impulse transient response of the dynamic control action
the proportional control action (P) $w(t) = K \cdot \delta(t)$	
the integral control action (I) $w(t) = K$	
the derivative control action (D) $w(t) = T\dfrac{d\delta(t)}{dt}$	
the first-order aperiodic control action (A-1) $w(t) = \dfrac{K}{T} e^{-\frac{t}{T}}$	

Continuation of the Table 2.3

Name the dynamic control action and impulse transient response equation	Impulse transient response of the dynamic control action
the vibration control action (V) $w(t)=\frac{K}{\omega T^2}\cdot e^{-\lambda t}\cdot \sin \omega t$, where $\omega=\frac{\sqrt{1-\xi^2}}{T}$, $\lambda=-\frac{\xi}{T}$	
the conservative control action (Con) $w(t)=\frac{K}{T}\cdot \sin\left(\frac{\sqrt{1-\xi^2}}{T}\cdot t\right)$	
the second-order aperiodic control action (A-2) $w(t)=\frac{K}{T_1-T_2}\cdot\left(e^{-\frac{t}{T_1}}-e^{-\frac{t}{T_2}}\right)$	

From the definition of a unit impulse function $\delta(t)=\left(1(t)\right)'$, a ratio is established between the transient response $h(t)$ and the impulse transient response $w(t)$, which looks as:

$$w(t)=h'(t) \qquad \text{or} \qquad h(t)=\int_0^t w(t)\cdot dt\,. \qquad (2.5)$$

The transient response $h(t)$ and the impulse transient response $w(t)$ are called **time responses** of typical dynamic control actions.

2.2 Tasks

1. Study the lecture on the topic «Time responses of typical dynamic control actions: transient responses and impulse transient responses».

2. Based on the block diagram (see Figure 1.13), build a block diagram

when an unit step function $x(t) = 1(t)$ and an unit impulse function $x(t) = \delta(t)$ are fed to its in Simulink (see Figure C.2,*a* of the Appendix C).

3. Replace the Gain blocks by blocks of the typical first- and second-order dynamic control actions with their transfer functions $W(s)$ one after the other according to Table 2.1.

3.1. The first typical dynamic control action you study is the integral control action (I). This is described by the transfer function $W(s) = \frac{K}{s} = \frac{1}{T \cdot s}$ (see Table 2.1). It can be found in the **Simulink/Continuous** library. The Integrator block icon is shown in Figure 2.1,*a*. It integrates the input signal $x(t)$ at zero initial conditions.

In the lab class, to realise the transfer function $W(s) = \frac{1}{T \cdot s}$, it is necessary to locate the Integrator block after the Gain block and connect it together as shown in Figure 2.1,*b*.

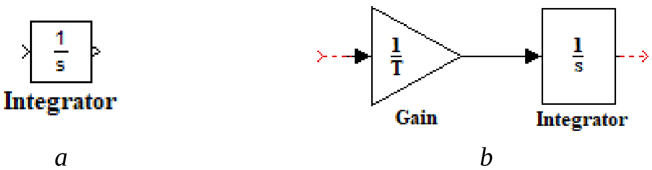


Figure 2.1 – Integrator block icon and
block diagram for realising the Integral control action

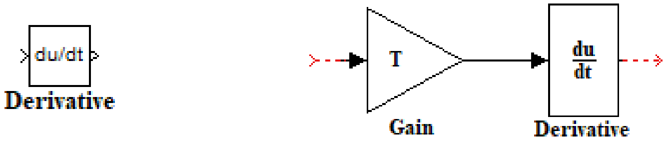
The time constant T of the transfer function $W(s) = \frac{1}{T \cdot s}$ is set in three variants: T_1, T_2 and T_3. You can take the time constants T_1, T_2 and T_3 from Table 2.4 for your variant N, where N is the variant number corresponding to the number in the student group list.

Enter time coefficients T_1, T_2 and T_3 in the *Gain* field of **Block Parameters: Gain** as shown in Figure 2.1,*b*, for the corresponding Gain block.

3.2. The second typical dynamic control action you study is the derivative control action (D). This is described by the transfer function $W(s) = T \cdot s$ (see

Table 2.1). It can be found in the **Simulink/Continuous** library. The Derivative block icon is shown in Figure 2.2,*a*. It differentiates the input signal $x(t)$ by the current time t.

In the lab class, to realise the transfer function $W(s) = T \cdot s$, it is necessary to locate the Derivative block after the Gain block and connect it together as shown in Figure 2.2,*b*.



a *b*

Figure 2.2 – Derivative block icon and
block diagram for realising the Derivative control action

The time constant T of the transfer function $W(s) = T \cdot s$ is set in three variants: T_1, T_2 and T_3. Take the time constants T_1, T_2 and T_3 from Table 2.4 for your variant N.

Enter time coefficients T_1, T_2 and T_3 in the *Gain* field of **Block Parameters: Gain** for the corresponding Gain block.

3.3. The next typical dynamic control actions you study are:

– the first-order aperiodic control action (A-1);

– second-order aperiodic control action (A-2);

– vibrational control action (V);

– conservative control action (Con).

These are described by transfer functions $W(s)$ as shown in Table 2.1. In the general form their transfer functions are the ratio of the polynomials of the numerator of the m-th degree and the denominator of the n-th degree:

$$W(s) = \frac{Y(s)}{X(s)} = \frac{b_m \cdot s^m + b_{m-1} \cdot s^{m-1} + ... + b_1 \cdot s + b_0}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + ... + a_1 \cdot s + a_0}. \qquad (2.6)$$

In the lab class, you can realise transfer functions of typical first- and second-order dynamic control actions using the Transfer Fcn block. It can be found in the **Simulink/Continuous** library. The Transfer Fcn block icon is shown in Figure 2.3.

In the Transfer Fcn block, the numerator and denominator polynomials are ***only*** set by coefficients, which are entered ***through a space*** in decreasing degrees of the Laplace operator *s*, as shown in Figure 1.10.

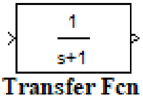


Figure 2.3 – Transfer Fcn block icon

Pay attention! The order of the numerator polynomial degree *m* must be less or equal to the order of the denominator polynomial degree *n* ($m \le n$).

For example, to realise a transfer function for a second-order aperiodic control action $W(s) = \dfrac{K}{T^2 \cdot s^2 + 2 \cdot T \cdot \xi \cdot s + 1} = \dfrac{10}{0{,}1^2 \cdot s^2 + 2 \cdot 0{,}1 \cdot 3 \cdot s + 1}$, set the data in the **Block Parameters: Transfer Fcn**, as shown in Figure 2.4:

- in the *Numerator* field, enter 10 in square brackets, [10], because *K*=10;
- in the *Denominator* field, enter 0.1^2, $2 \cdot 0.1 \cdot 3$ and 1 through a space in square brackets, [0.1^2 $2 \cdot 0.1 \cdot 3$ 1], because: $T^2 = 0{,}1^2$, $2 \cdot T \cdot \xi = 2 \cdot 0{,}1 \cdot 3$.

According to your variant *N*, take the transfer ratio *K*, time constants T_1, T_2 and T_3, damping ratios ζ_1 (for vibration control action), ζ_2 (for second-order aperiodic control action) and ζ_3 (for conservative control action) from Table 2.4 and set them in the **Block Parameter: Transfer Fcn** for the corresponding Transfer Fcn block, as shown in the example.

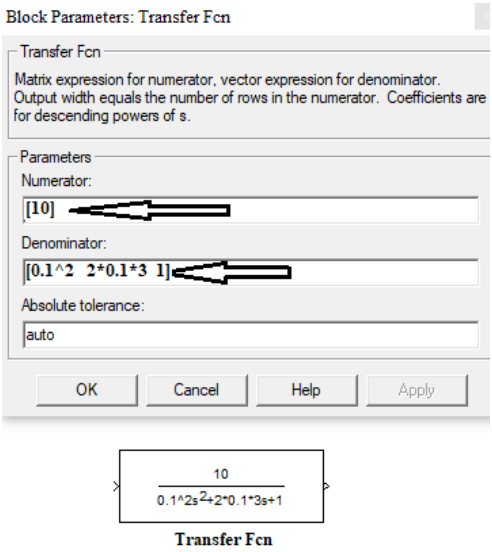


Figure 2.4 – Setting data in the Transfer Fcn Block

So, the block diagram in Figure C.2,*a* of the Appendix C is created.

4. Copy the created block diagram to the clipboard as described in the

Theoretical part: Copy and move model components to the clipboard.

5. Open a Word Document and paste the created block diagram from the clipboard using the **Edit** → **Paste** command from the menu or by clicking *Paste*  button on the toolbox in Word Document.

6. Set the simulation parameters as described in the **Theoretical part: Setting Simulation Parameters and running simulation**.

7. Save the created block diagram as described in the **Theoretical part: Save the block diagram and script as a file**.

8. Obtain the transfer functions of the typical first- and second-order dynamic control actions from Table 2.1 using the code in Appendix C.2.

9. Study the behaviour of the following typical dynamic control actions one after the other:

– the integral control action (I);

– the derivative control action (D);

– the first-order aperiodic control action (A-1);

– the second-order aperiodic control action (A-2);

– vibrational control action (V);

– conservative control action (Con)

for given input signals $x(t)$:

– the unit step function $x(t) = 1(t)$;

– the unit impulse function $x(t) = \delta(t)$.

The behaviour of the typical dynamic control actions listed above is studied at different values of time constants T_1, T_2, T_3 and damping ratios ζ_1, ζ_2, ζ_3. For this, switch the Transfer Fcn blocks one after the other using the Manual Switch block.

Run simulation after each Transfer Fcn block switch by clicking *Start Simuation*  button on the toolbox.

10. Plot the transient responses of the input $x(t)$ and output $y(t)$ signals in the same coordinate axes. For this, in the MATLAB window, type the **plot** and **grid on** commands as shown below:

```
>> plot(t,x,'k-',t,y,'r-');grid on
```

To plot the all transient output responses of the Transfer Fcn blocks $y(t)$ when studying with the one input signal type $x(t)$ in the same coordinate axes, you must type the **hold on** command and repeat the **plot** command again:

```
>> hold on
>> plot(t,x,'k-',t,y,'b-');grid on
```

11. Label the transient responses of the input $x(t)$ and output $y(t)$ signals, the X and Y axis using  and  buttons on the Figure toolbox as shown in Figure A.1 of the Appendix A.2.

12. Copy the Figure to the clipboard by clicking **Edit → Copy Figure**.

13. Open a Word Document again and paste the Figure from the clipboard using the **Edit → Paste** command from the menu or by clicking *Paste*  button on the toolbox in Word Document.

14. Make a report (see Appendix C) and save it on disk or USB.

15. Get the mark for your lab class from your teacher by answer-ing his/her control questions.

2.3 Self-control questions

1. What is called a typical dynamic control action? Name the typical first- and second-order dynamic control actions?

2. What responses are called time responses? What is called a transient response $h(t)$ and an impulse transient response $w(t)$.

3. How to determine the transient response $h(t)$ on the transfer function?

4. How to determine the impulse transient response $w(t)$ on the transfer function?

5. How to determine the impulse transient response $w(t)$ of a dynamic control action by the transient response $h(t)$?

6. Write down the transfer function and plot the time responses of the first-order typical dynamic control actions.

7. Write down the transfer function and plot the time responses of the second-order typical dynamic control actions.

8. Assess the effect of changing the parameters of the dynamic control actions on their time responses.

Table 2.4 – Data of the typical dynamic control actions

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
K	10	15	20	25	30	35	40	45	50	10
T_1	0,1	0,1	0,1	0,1	0,2	0,2	0,2	0,2	0,3	0,3
T_2	1	1	1	1	1	1	1	1	1	1
T_3	1,5	2	2,5	3	3,5	4	4,5	5	1,5	2
ζ_1	0,3	0,4	0,5	0,6	0,7	0,3	0,4	0,5	0,6	0,7
ζ_2	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,0
ζ_3	0	0	0	0	0	0	0	0	0	0

N	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
K	15	20	25	30	35	40	45	50	10	15
T_1	0,3	0,3	0,4	0,4	0,4	0,4	0,5	0,5	0,5	0,5
T_2	1	1	1	1	1	1	1	1	1	1
T_3	2,5	3	3,5	4	4,5	5	2	2,5	3	3,5
ζ_1	0,3	0,4	0,5	0,6	0,7	0,3	0,4	0,5	0,6	0,7
ζ_2	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,0	1,1
ζ_3	0	0	0	0	0	0	0	0	0	0

N	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
K	20	25	30	35	40	45	50	10	15	10
T_1	0,6	0,6	0,6	0,6	0,7	0,7	0,7	0,7	0,8	0,8
T_2	1	1	1	1	1	1	1	1	1	1
T_3	4	4,5	5	2	2,5	3	3,5	4	4,5	5
ζ_1	0,3	0,4	0,5	0,6	0,7	0,3	0,4	0,5	0,6	0,7
ζ_2	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,0	1,1	1,2
ζ_3	0	0	0	0	0	0	0	0	0	0

Lab class 3

Study of the frequency response of typical dynamic control actions

Objective of the work:

– getting skills in writing MATLAB script code that generates the transfer functions of the typical dynamic control actions;

– studying the frequency response of typical dynamic control actions.

3.1 Theoretical part

The **frequency response** is a function that relates the output response to a sinusoidal input $x(t) = A_{\text{input}} \sin(\omega \cdot t)$ at frequency ω.

If a sine wave of a certain frequency ω is fed to the dynamic control action, the output signal is also sine wave and has the same frequency ω, but with a different amplitude $A_{\text{output}}(t)$ and phase shift $\varphi = \frac{\Delta t}{T} \cdot 360^0$:

$$y(t) = A_{\text{output}} \cdot \sin(\omega \cdot t + \varphi), \qquad (3.1)$$

where T – is period, sec.

Figure 3.1 shows the sinusoidal input $x(t)$ and output $y(t)$ signals of a dynamic control action in steady-state. Here the output wave lags behind the input wave by an offset angle.

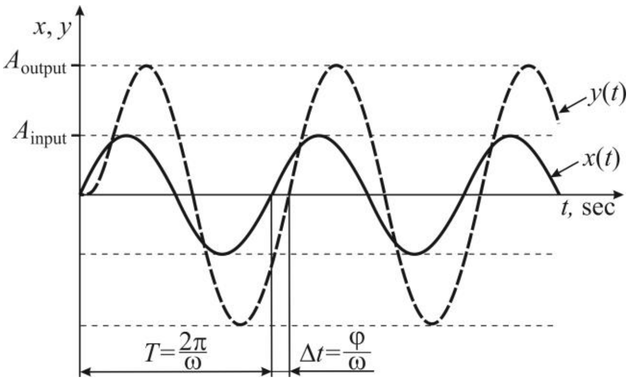


Figure 3.1 – The sinusoidal input $x(t)$ and output $y(t)$ signals of the dynamic control action in steady-state

For any dynamic control action, the ratio of the output wave amplitude to the input wave amplitude $\frac{A_{\text{output}}}{A_{\text{input}}}$ and the phase shift between the waves of the output signal and the input signal φ depend ***only*** on the frequency ω. The ratio of the output wave amplitude and the input wave amplitude is described by the frequency transfer function $\frac{A_{\text{output}}}{A_{\text{input}}} = H(j\omega)$.

The **frequency transfer function** $H(j\omega)$ is the ratio of the Fourier transform of the sinusoidal output signal to the Fourier transform of the sinusoidal input signal at zero initial conditions.

The Fourier transform can be obtained from the Laplace transform by substitute the Laplace operator s in the transfer function $W(s)$ with the complex number $j\omega$:

$$H(j\omega) = W(s)\big|_{s=j\omega}. \quad (3.2)$$

The basic typical dynamic control actions and their frequency transfer functions $H(j\omega)$ are shown in Table 3.1.

Table 3.1 – Frequency transfer functions of typical first- and second-order dynamic control actions

Name the dynamic control action	Frequency transfer functions of the dynamic control action
the integral control action (I)	$H(j\omega) = -j\frac{K}{\omega}$
the derivative control action (D)	$H(j\omega) = j \cdot K \cdot \omega$
the first-order aperiodic control action (A-1)	$H(j\omega) = \frac{K}{(T\cdot\omega)^2+1} - j\frac{K\cdot T\cdot\omega}{(T\cdot\omega)^2+1}$
the vibration control action (V) and the second-order aperiodic control action (A-2)	$H(j\omega) = \frac{K\cdot(1-T^2\cdot\omega^2)}{(1-T^2\cdot\omega^2)^2+(2\cdot\xi\cdot T\cdot\omega)^2} - j\frac{K\cdot 2\cdot\xi\cdot T\cdot\omega}{(1-T^2\cdot\omega^2)^2+(2\cdot\xi\cdot T\cdot\omega)^2}$
the conservative control action (Con)	$H(j\omega) = \frac{K}{1-(T\cdot\omega)^2}$

The frequency transfer function $H(j\omega)$ is a complex number. Therefore, it will have real $U(\omega)$ and imaginary $V(\omega)$ parts:

$$H(j\omega) = U(\omega) + jV(\omega) = H(\omega) \cdot e^{j\varphi(\omega)}, \qquad (3.3)$$

where $U(\omega) = \operatorname{Re} H(j\omega) = \dfrac{A_{\text{output}} \sin(\omega t + \varphi)}{A_{\text{input}} \sin \omega t}$ – is real part of the frequency transfer function $H(j\omega)$;

$V(\omega) = \operatorname{Im} H(j\omega) = \arg H(j\omega)$ – is imaginary part of of the frequency transfer function $H(j\omega)$;

$H(\omega) = |H(j\omega)|$ – is the modulus of the vector $H(j\omega)$ on the complex plane or the *amplitude frequency response*;

$\varphi(\omega) = \angle H(j\omega)$ – is the angle (argument) or phase of the vector $H(j\omega)$ or the *phase frequency response*.

The **amplitude frequency response** is the square root of the sum of the squares of the real and imaginary parts:

$$|H(j\omega)| = \sqrt{U(\omega)^2 + V(\omega)^2}\ . \qquad (3.4)$$

The amplitude frequency response is built on the complex plane. The *X*-axis indicates the frequency changes from zero to infinity ($0 < \omega < \infty$), the *Y*-axis the modulus (length) of the vector $H(j\omega)$;

The **phase frequency response** is the arctangent of the ratio of the imaginary part of the frequency response to the real part of the frequency response. The phase frequency response is built on the complex plane. The *X*-axis indicates the frequency changes from zero to infinity ($0 < \omega < \infty$), the *Y*-axis the angle $\varphi(\omega)$ (argument) or phase of the vector $H(j\omega)$.

Pay attention! The phase frequency response of the **first-order** dynamic control actions tends to $-\dfrac{\pi}{2}$ $\left(\varphi \to -\dfrac{\pi}{2}\right)$ and is defined by the formula:

$$\varphi(\omega) = arctg \frac{V(\omega)}{U(\omega)}. \qquad (3.5)$$

The phase frequency response of the **second-order** dynamic control actions tends to $-\pi$ $(\varphi \to -\pi)$ and is defined by the formula:

$$\varphi(\omega) = -\pi + arctg \frac{V(\omega)}{U(\omega)}\ . \qquad (3.6)$$

When the frequency ω changes from 0 to infinity, the vector of the frequency transfer function $\left|H(j\omega)\right|$ draws around a curve called *amplitude-phase frequency response*.

The **amplitude-phase frequency response** represents is the geometrical location of the ends of the vectors or hodograph corresponding to the frequency response $H(j\omega)$ when the frequency changes from zero to infinity ($0 < \omega < \infty$). The amplitude-phase frequency response is built on the complex plane. The *X*-axis indicates the real part of the transfer function $U(\omega)$, the *Y*-axis the imaginary part of the transfer function $V(\omega)$. In essence, such a graph combines in one plane the amplitude frequency $H(\omega)$ and phase frequency $\varphi(\omega)$ responses.

Pay attention! The amplitude-phase frequency response of the **first-order** dynamic control actions is located in *the fourth quadrant* of the complex plane.

The amplitude-phase frequency response of the **second-order** dynamic control actions is located in *the third and fourth quadrants* of the complex plane.

The **real frequency response** $U(\omega)$ is the even frequency transfer function $H(j\omega)$. The *X*-axis indicates the frequency changes from zero to infinity ($0 < \omega < \infty$), the *Y*-axis indicates the real part of the frequency transfer function $U(\omega)$.

The **imaginary frequency response** $V(\omega)$ is an uneven frequency transfer function $H(j\omega)$. The *X*-axis indicates the frequency changes from zero to infinity ($0 < \omega < \infty$), the *Y*-axis indicates the imaginary part of the frequency transfer function $V(\omega)$.

In practice, the amplitude frequency $H(\omega)$ and phase frequency $\varphi(\omega)$ responses are often represented on a logarithmic scale:

– the **logarithmic amplitude frequency response** $L(\omega)$ represent is the amplitude frequency response in which the amplitude of the frequency response $H(j\omega)$ is expressed in decibels, dB, and the frequency is expressed in logarithmic scale $\lg(\omega)$;

$$L(\omega) = 20\lg H(\omega); \tag{3.7}$$

– the **logarithmic phase frequency response** $\varphi(\omega)$ represent is the phase frequency response in which the phase is expressed in degrees and the frequency is expressed in logarithmic scale $\lg(\omega)$.

In the lab class, to build the frequency responses for typical dynamic control action from Table 3.1, it is more convenient to use the **M-files Editor/MATLAB (script-files/MATLAB)**. An M-file or script file, is a simple text file, in which MATLAB commands are placed. When the file is run, MATLAB reads the commands all at once or in parts. All M-file names must end with the extension «name.m».

3.2 Tasks

1. Study the lecture on the topic «Frequency responses of typical dynamic control actions».

2. Run MATLAB.

3. To write a new script in MATLAB you need to open M-file Editor window. For this, select **File** → **New** → **M-file** command from the menu or click **New M-file**  button on the MATLAB toolbox. It opens a new Untitled window, as shown in Figure 3.2.

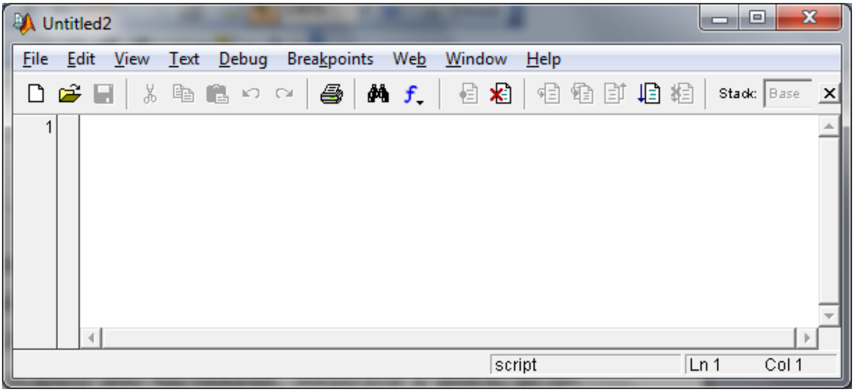


Figure 3.2 – M-file Editor Window

4. After opening the Untitled window in the M-file Editor window, code can be written to build the frequency responses of typical dynamic control actions with the transfer functions $W(s)$ from Table 2.1.

Pay attention! The code below are **only** for building the frequency responses of the **first-order** dynamic control actions:

clc;	Clear the Command Window
clear all;	Removal of variables and functions from clipboard
w=[0:0.1:100];	Setting the frequency range ω from 0 to 100 in step Δt = 0,1. **Pay attention!** The frequency range can be **any**
s=j.*w;	Substitute the Laplace operator s with the complex number $j\omega$: $s = j\omega$
K = your variant data;	Set the transfer ratio data of the typical dynamic control action K from Table 2.4, according to your variant N
T = your variant data;	Set the time constant data of typical dynamic control action T_1 from Table 2.4, according to your variant N
W = the transfer function of the typical dynamic control action you are studying *For example*, the transfer function of a **first order** ape-riodic control action (A-1) in the general form is written as follows: W = K./(T.*s+1)	Write down the transfer function in the general form of a typical dynamic control action $W(s)$ you are studying. As a result of the substitute $s = j\omega$, the transfer function in the Laplace transforms $W(s)$ is convert into the frequency transfer function $H(j\omega)$. **Pay attention!** The transfer functions of the typical dynamic control actions can be found in Table 2.1. According to MATLAB syntax, put a dot before the operations of division (/), multi-plication (*) and exponentiation (^).
U = real(W); V = imag(W);	Select the real $U(\omega)$ and imaginary $V(\omega)$ parts from the frequency transfer function $H(j\omega)$. See formula 3.3
H = sqrt(U.^2+V.^2);	Amplitude calculation of frequency transfer function $H(j\omega)$. See formula 3.4

Continuation of the code

fi = atan(V./U).*(180./pi)	Phase calculation of frequency transfer function *H*(*j*ω). See formula 3.5
figure;plot(U,V);grid on	Building the amplitude-phase frequency response in a separate figure window. **Pay attantion!** The amplitude-phase frequency response of a first order dynamic control actions is located in the *fourth* quadrant of the complex plane
figure;plot(w,H);grid on	Building the amplitude frequency response *H*(ω) in a separate figure window
figure;plot(w,fi);grid on	Building of the phase frequency response φ(ω) in a separate figure window
figure; bode([K],[denominator coefficients **only**]); grid on *For example*, the **bode** command for the transfer function of the **first order** aperiodic control action (A-1) $W(s)=\dfrac{K}{T\cdot s+1}$ is written as follows: bode([K],[T 1]), where *T* – coefficient before Laplace operator *s* of the first order, 1 – free coefficient.	Building the logarithmic amplitude frequency response *L*(ω) and the logarithmic phase frequency response φ(ω) under each other in a separate figure window. **Pay attantion!** Enter denominator coefficients ordered by decreasing Laplace operator *s* (see Figure 2.4). By default, MATLAB clears the figure before each **plot** command. Using the **figure** command to open a new figure window. Command **grid on** displays the major grid lines for the current axes

Pay attention! The code for building the frequency responses of the **second-order** dynamic control actions are shown in Appendix C.

5. Save the written code as described in the **Theoretical part: Save the block diagram and script as a file**.

6. Run the M-file Editor. Click the *Run*  button on the toolbox and build the frequency responses.

7. Label the *X* and *Y* axis using  and  buttons on the Figure toolbox as shown in Figure C.1 of the Appendix C.

8. Copy the Figure to the clipboard by clicking **Edit** → **Copy Figure**.

9. Open a Word Document and paste the Figure from the clipboard using the **Edit** → **Paste** command from the menu or by clicking *Paste*  button on the toolbox in Word Document.

10. Make a report (see Appendix C) and save it on disk or USB.

11. Get the mark for your lab class from your teacher by answering his/her control questions.

3.3 Self-control questions

1. What are the frequency responses?

2. How to determine the frequency responses having a control action transfer function?

3. What response is called Amplitude-phase frequency response and how to build it?

4. What responses are called Real $U(\omega)$ and Imaginary $V(\omega)$ frequency responses and how to build them?

5. How to determine the amplitude and argument from the amplitude-phase frequency response?

6. What responses are called Logarithmic amplitude-frequency response $L(\omega)$ and the Logarithmic phase-frequency response $\varphi(\omega)$ and how to build them?

7. What are the units of measurement taken on the coordinate axes of Logarithmic amplitude-frequency response $L(\omega)$ and the Logarithmic phase-frequency response $\varphi(\omega)$?

8. How will the logarithmic amplitude-frequency and logarithmic phase-frequency response control actions change if the transfer ratio increases 100 times?

9. Which typical control action does not change the phase of a harmonic signal of any frequency?

REFERENCES

1. Richard C. Dorf and Robert H. Bishop Modern control systems, 12t edition, PrenticeHall, 2011, 1071 p.
2. Karl Johan Astrom and Richard M. Murray Feedback Systems an Introduction for Scientists and Engineers, DRAFT v2.7a (17 July 2007), 412 p.
3. Simulink Getting Started Guide – MathWorks – MATLAB and Simulink, 2016, 92 p.
4. Никульчев Е. В. Control System Toolbox, [online] Available: http:// www:matlab.exponenta.ru.
5. Дьяконов В. П. MATLAB 6.5 SP1/7 + Simulink 5/6 в математике и моделировании. Серия «Библиотека профессионала», Москва : СОЛОН-Пресс, 2005, 576 с.
6. Черных И. В. Моделирование электротехнических устройств в MATLAB, SimPowerSystems и Simulink, Саратов : Профобразование, 2017, 288 с.
7. Бутова О. А. Методические указания к выполнению лабораторной работы «Исследование типовых динамических звеньев в пакете MATLAB» по дисциплине «Теория автоматического регулирования» : для студ. всех форм обучения спец. «Электроника», в том числе для иностранных студ. / О. А. Бутова, Л. В. Фетюхина, М. А. Шишкин, Л. В. Асмолова, В. Н. Шамардина, Харьков : НТУ «ХПИ», 2018, 72 с.
8. Герман-Галкин С. Г. Matlab & Simulink. Проектирование мехатронных систем на ПК. ,СПб.: КОРОНА-Век, 2008, 368 с.
9. Асмолова Л. В. Основи дослідження лінійних динамічних систем у середовищі пакету MATLAB : лаб. практикум з курсу «Теорія автоматичного керування». Ч. 1 : для студентів усіх форм навчання за спец. 141 Електроенергетика, електротехніка та електромеханіка / Л. В. Асмолова, В. М. Шамардіна, Харків : НТУ «ХПІ», 2020, 64 с.
10. Harold Klee and Randal Allen Simulation of Dynamic Systems with MATLAB and Simulink, Third Edition, CRC Press Taylor & Francis Group, 2018, 853 p.

APPENDIXES

APPENDIX A
Formatting, adding labels, adjusting colours on plots for report

A.1 Create 2-D line plot

The functions $y = f(x)$ are used in mathematical calculations as well as in computer simulations and are plotted as a graph in a coordinate system: *X*-axis and *Y*-axis. The resulting set of points are connected by straight line sections. In MATLAB they create a one-dimensional (1-D) array matrix.

To create 2-D line plot, use the plot command:

- **plot (X,Y)** – creates a 2-D line plot of the data in *Y* versus the corresponding values in *X*.

If *X* and *Y* are both vectors, then they must have equal length. The plot function plots *Y* versus *X*.

If *X* and *Y* are both matrices, then they must have equal size. The plot function plots columns of *Y* versus columns of *X*.

If one of *X* or *Y* is a vector and the other is a matrix, then the matrix must have dimensions such that one of its dimensions equals the vector length. If the number of matrix rows equals the vector length, then the plot function plots each matrix column versus the vector. If the number of matrix columns equals the vector length, then the function plots each matrix row versus the vector. If the matrix is square, then the function plots each column versus the vector.

- **plot(X1,Y1,'color_type'...,Xn,Yn,'color_type')** plots more than one pair of *X*, *Y* using the same axes for all plots. Each plots is given a colour and line style, which you can select from Table A.1.

Table A.1 – Colour and line style to identify the plot

Line color				***Line style***	
'r'	red	'g'	green	'–'	solid line
'b'	blue	'm'	purple	':'	double dotted line
'k'	black	'y'	yellow	'-.'	dotted line
'c'	cyan	'w'	white	'--'	dashed

The end of the APPENDIX A

A.2 Adding labels and axes to a figure

To pass on important information about the plot you need to make labels.

Click **Insert Text**  button on the **Figure/Simulink** toolbox.

Left-click on the desired location in the figure and a text label cursor will appear. The text labels are written in English.

As an example, Figure A.1 shows a block diagram and the transient responses of the input *x*(*t*) and output *y*(*t*) signals using the plot command:

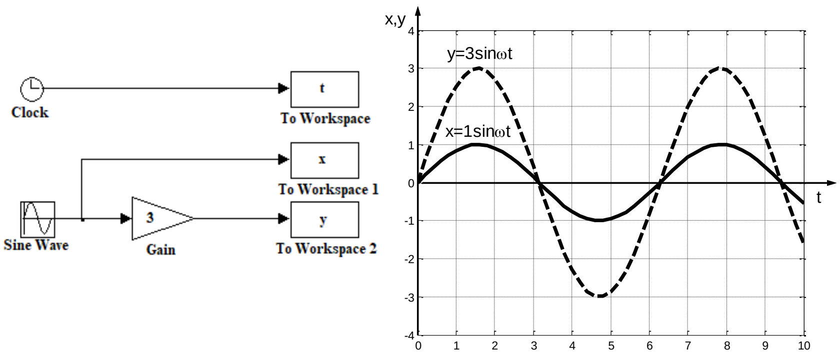
```
>> plot(t,x,'k-',t,y,'k--.');grid on
```

The **plot** command describes that the plot of the input signal *x*(*t*) is indicated by a **solid black** line and the plot of the output signal *y*(*t*) is indicated by a **dashed black** line.

The **grid on** command displays the axis grid lines.

Axis labels have been added to the Figure: the *X*-axis describes the time *t*, the *Y*-axis describes the input *x* and output *y* signals.

To draw arrows showing the direction of the *X*- and *Y*- axes in Figure, click **Insert Arrow**  button on the **Figure/Simulink** toolbox. The cursor changes to a cross (+). Press and hold the left mouse button and drag the cross from start to end limits of the *X*-axis or *Y*-axis, as shown in Figure A.1,*b*.



a *b*

Figure A.1 – Block diagram and transient responses of input *x*(*t*) and output *y*(*t*) signals in the same axes

APPENDIX B

Report fragment of the lab class 1

Basic knowledge of MATLAB and Simulink. Building simple models

B.1 Report content

1. Title page (see Appendix E)
2. Objective of the work
3. Results:

3.1 Block diagram for studying the Gain block with different input signals

3.2 Transient responses of the input $x(t)$ and output $y(t)$ signals in the same coordinate axes in study the behaviour of the Gain block with different values of transfer ratios K to the given input signals $x(t)$:

– the unit step function;
– the linearly increasing signal;
– the unit impulse function;
– the sinusoidal signal.

4. Conclusion

B.2 Example of report design

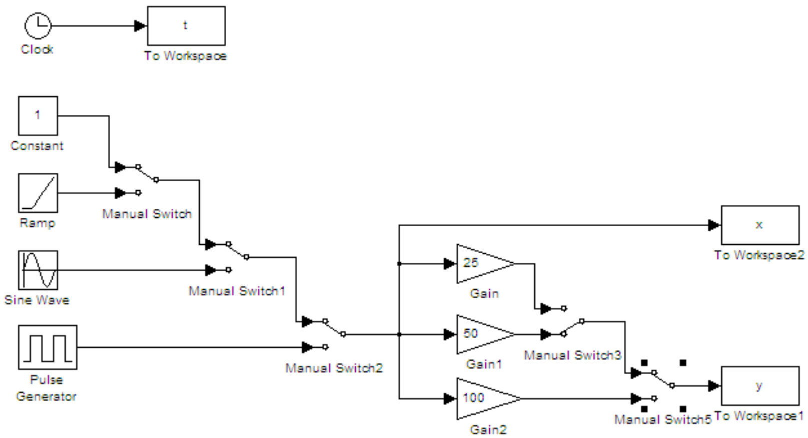


Figure B.1 – Block diagram for studying the Gain block with different input signals

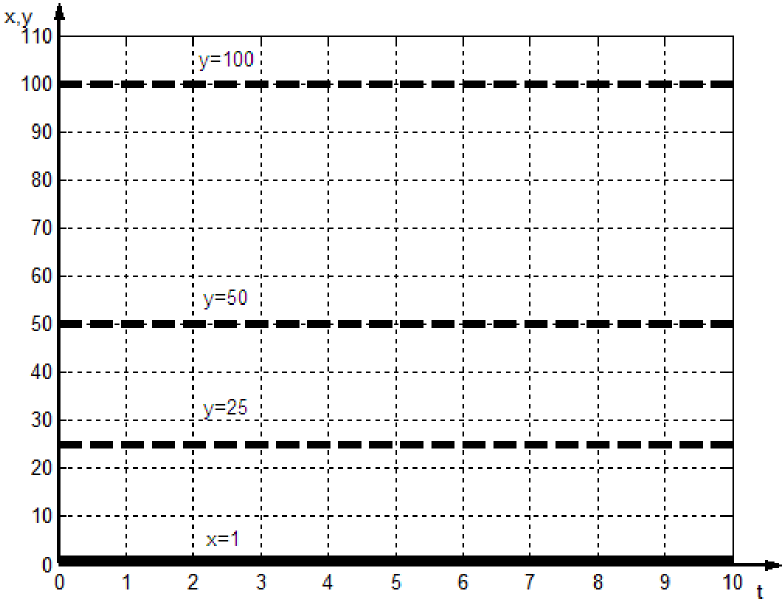


Figure B.2 – Transient responses of the ***Gain*** block at different values of the transfer ratio ***K*** when a unit step function $x(t)=1(t)$ is fed to the input

APPENDIX C
Report fragment of the lab class 2
Study of time responses of typical dynamic control actions

C.1 Report content

1. Title page (see Appendix E)

2. Objective of the work

3. Results of studying the following typical dynamic control actions:

– the integral control action (I);

– the derivative control action (D);

– the first-order aperiodic control action (A-1);

– the second-order aperiodic control action (A-2);

– vibrational control action (V);

– conservative control action (Con):

3.1 Block diagram for studying the behaviour of typical first- and second-order dynamic control actions for given input signals $x(t)$:

– the unit step function $x(t) = 1(t)$;

– the unit impulse function $x(t) = \delta(t)$.

3.2 Transfer functions of the typical first- and second-order dynamic control actions.

3.3 Transient responses of the input $x(t)$ and output $y(t)$ signals in the same axes in study the behaviour of the typical first- and second-order dynamic control actions.

4. Conclusion

C.2 Code for transfer functions of typical first- and second-order dynamic control actions

num=[K]	– this is the numerator of the transfer function in the Laplace transform $W(s)$ from Table 2.1. Take numerical data from Table 2.4
den=[a_2 a_1 a_0]	– this is the denominator of the transfer function in the Laplace transform $W(s)$ from Table 2.1. Take numerical data from Table 2.4
sys=tf(num,den)	– is the command to obtain the transfer function $W(s)$ as the ratio of the numerator to the denominator

C.3 Example of report design

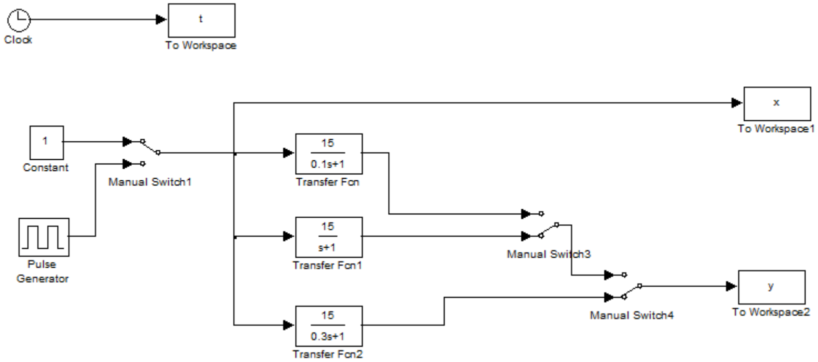


Figure C.1 – Block diagram for studying the Gain block with different input signals

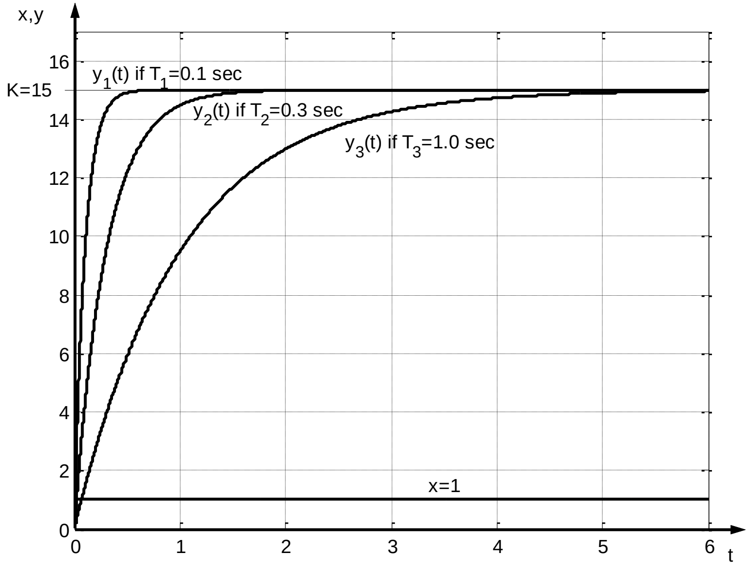


Figure C.2 – Transient responses of the first order aperiodic control action at different values of the time constant *T* when a unit step function $x(t)=1(t)$ is fed to the input

APPENDIX D
Report fragment of the lab class 3
Study of frequency responses of typical dynamic control actions

D.1 Report content

1. Title page (see Appendix E)
2. Objective of the work
3. Results of studying the following typical dynamic control actions:

– the integral control action (I);
– the derivative control action (D);
– the first-order aperiodic control action (A-1);
– the second-order aperiodic control action (A-2);
– vibrational control action (V);
– conservative control action (Con):

3.1 Code for building the frequency responses of typical first- and second-order dynamic control actions.

3.2 Transfer functions of the typical first- and second-order dynamic control actions.

3.3 Frequency responses of the typical first- and second-order dynamic control actions.

4. Conclusion

D.2 M-file to build frequency responses of the typical second-order dynamic control actions

As an example, here are the code for building the frequency responses of the vibrational control action described by a transfer function:

$$W(s) = \frac{K}{T^2 \cdot s^2 + 2 \cdot \xi \cdot T \cdot s + 1} = \frac{100}{0{,}01s^2 + 0{,}1s + 1}$$

clc;	Clear the Command Window
clear all;	Removal of variables and functions from clipboard
K = 100;	Enter the transfer ratio data K = 100 from Table 2.1
T = 0.1;	Enter the time constant data from Table 2.1 or calculate using the formula: $T = \sqrt{T^2} = \sqrt{0{,}01} = 0{,}1$

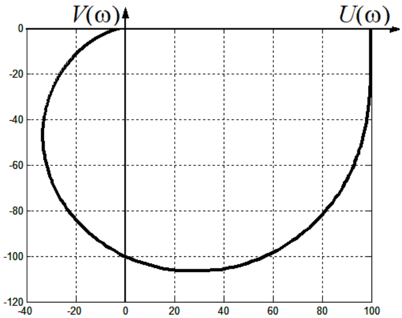
Continuation

ksi=0.5;	Enter the damping ratio data from Table 2.1 or calculate using the formula: $2 \cdot \xi \cdot T = 0{,}1$ $2 \cdot \xi \cdot 0{,}1 = 0{,}1$ $0{,}2 \cdot \xi = 0{,}1$ $\xi = \frac{0{,}1}{0{,}2} = 0{,}5$
w1=[0:0.1:(1/T-0.1)]; w2=[1/T:0.1:100];	Setting the frequency range ω from 0 to 100 in step Δt = 0.1. **Pay attention!** The frequency range ω can be any. The frequency range ω for the second order control actions is set before $\omega = \frac{1}{T}$ and after it, as the amplitude-phase frequency response is located in the third and fourth quadrants of the complex plane. The frequency between the third and fourth quadrant is $\omega = \frac{1}{T}$. Therefore, frequency range w1, if $\omega \le \frac{1}{T}$; frequency range w2, if $\omega > \frac{1}{T}$
s1=j*w1; s2=j*w2;	Substitute the Laplace operator *s* with the complex number $j\omega$: for w1 will be s1; for w2 will be s2
W1=K./(T.^2.*s1.^2+2.*ksi.*T.*s1+1); W2=K./(T.^2.*s2.^2+2.*ksi.*T.*s2+1);	Write down the transfer function in the general form of the vibrational control action: $W(s) = \frac{K}{T^2 \cdot s^2 + 2 \cdot \xi \cdot T \cdot s + 1}$ for s1 will be W1; for s2 will be W2

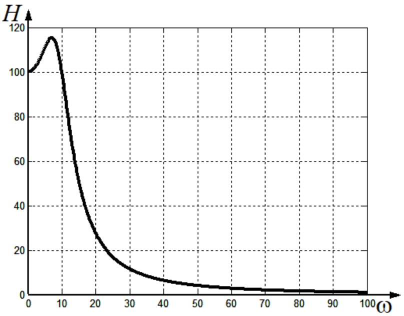
Continuation

U1=real(W1); U2=real(W2);	Select the real $U(\omega)$ part from the frequency transfer function $H(j\omega)$, according to formula 3.3: for W1 will be U1; for W2 will be U2
V1=imag(W1); V2=imag(W2);	Select the imaginary $V(\omega)$ part from the frequency transfer function $H(j\omega)$, according to formula 3.3: for W1 will be V1; for W2 will be V2
H1=sqrt(U1.^2+V1.^2); H2=sqrt(U2.^2+V2.^2);	Amplitude calculation of frequency transfer function $H(j\omega)$, accoding to formula 3.4: for U1 and V1 will be H1; for U2 and V2 will be H2
fi1=atan(V1./U1)*(180/pi); fi2=(-pi+atan(V2./U2))*(180/pi);	Phase calculation of frequency transfer function $H(j\omega)$, accoding to formula 3.6: for U1 and V1 will be fi1 if $\omega \le \frac{1}{T}$; for U2 and V2 will be fi2 if $\omega > \frac{1}{T}$
figure;plot(U1,V1,’r’); hold on; plot(U2,V2,'b'); grid on;	Building the amplitude-phase frequency response in a separate figure window **Pay attention!** Amplitude-phase frequency response of second order dynamic control actions is located in the third and fourth quadrants of the complex plane
figure;plot(w1,H1,’r’); hold on; plot(w2,H2,'b') ;grid on	Building the amplitude frequency response $H(\omega)$ in a separate figure window
figure;plot(w1,fi1,'r'); hold on; plot(w2,fi2,'b'); grid on;	Building of the phase frequency response $\varphi(\omega)$ in a separate figure window
figure;bode([K],[T^2 2*ksi*T 1]); grid on	Building the logarithmic amplitude frequency response $L(\omega)$ and the logarithmic phase frequency response $\varphi(\omega)$ under each other in a separate figure window

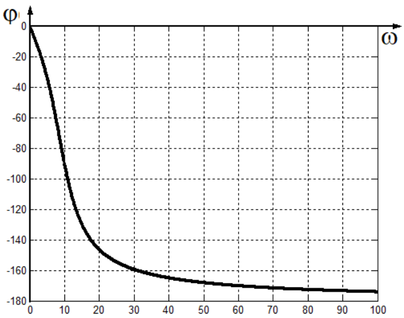
D.3 Simulation results



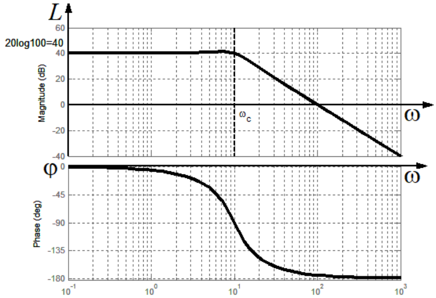
a) Amplitude-phase frequency response



b) Amplitude frequency response



c) Phase frequency response



d) Logarithmic amplitude-frequency response $L(\omega)$ and the Logarithmic phase-frequency response $\varphi(\omega)$

Figure D.1 – Frequency responses of the Vibrational control action

Навчальне видання

АСМОЛОВА Лариса Валеріївна
ШАМАРДІНА Віра Миколаївна
ЧУДОВСЬКА Тетяна Сергіївна

ОСНОВИ ДОСЛІДЖЕННЯ ТИПОВИХ ДИНАМІЧНИХ ЛАНОК У СЕРЕДОВИЩІ ПАКЕТУ MATLAB

Навчальний посібник
для студентів спеціальності
141 – Електроенергетика, електротехніка та електромеханіка
(англійською мовою)

Роботу до друку рекомендував доц. Олександр ЧЕПЕЛЮК

В авторській редакції

План 2021 р., поз. 86

Підп. до друку 21.10.2021 р. Формат 60×84 1/16. Папір офсет.
Riso – друк. Гарнітура Times New Roman. Ум. друк. арк. 3.26.
Наклад 50 прим. Зам. № 0016. Ціна договірна.

Видавничий центр НТУ “ХПІ”
Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.
61002, Харків, вул. Кирпичова, 2

ТОВ «ПромАрт»
61023, Харків, вул. Весніна, 12
Свідоцтво суб’єкта видавничої справи серії ДК № 5748 від 06.11.2017.
тел. (057) 717–28–80
www.promart.in.ua
e-mail: promartizdat@gmail.com