Andrius PETROVAS
Roma RINKEVIČIENĖ

AUTOMATIC CONTROL THEORY I, II

Project No
VP1-2.2-ŠMM-07-K-01-047

The Essential Renewal of Undergraduates Study Programs of VGTU Electronics Faculty

Vilnius "Technika" 2012

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Andrius PETROVAS
Roma RINKEVIČIENĖ

AUTOMATIC CONTROL THEORY I, II

A Laboratory Manual

Vilnius "Technika" 2012

A. Petrovas, R. Rinkevičienė. Automatic Control Theory I, II: A Laboratory Manual. Vilnius: Technika, 2012. 98 p. [4,09 author's sheets. 2012 06 04].

Descriptions of laboratory works are for students of technology sciences study field, electrical engineering study branch, automation study programme, taking course in automatic control theory. Descriptions of laboratory works will also be useful for incoming students in the frame of Erasmus program or other students of Exchange programs in energetic systems and mechatronics study branches. Every description of the laboratory work includes the aim of the work, a wide theoretical part, tasks, content of the report and control questions. Basic knowledge of MATLAB application is presented and specialized commands for this purpose are considered.

The publication has been recommended by the Study Committee of
VGTU Electronics Faculty.

Reviewed by:
Assoc. Prof PhD Saulius Lisauskas, VGTU Department of Automation
Assoc. Prof PhD Zita Savickienė, VGTU Department of Automation

This publication has been produced with the financial assistance of Europe Social Fund and VGTU (Project No VP1-2.2-ŠMM-07-K-01-047). The book is a part of the project "The Essential Renewal of Undergraduates Study Programs of VGTU Electronics Faculty".

This is an educational methodology book, No 1335-S, issued by VGTU Press TECHNIKA http://leidykla.vgtu.lt

Language editor *Vida Bėkštienė*
Typesetter *Donaldas Petrauskas*

eISBN 978-609-457-162-6
doi:10.3846/1335-S

Contents

INTRODUCTION

The descriptions of Automatic control laboratory works are designed for students, taking course in automatic control theory for automation study programme or students of other specialities, taking the same course. The laboratories are devoted to get acquainted with synthesis and analysis of automatic control systems with application of MATLAB and Simulink software. The main knowledge about MATLAB is presented here, the main tools for system analysis in time and frequency domain are considered. The main commands and the main principles of system modeling are overwieved. Application of Simulink facilitates analysis of system responses. Freguency response analysis indicates system stability and stability margins as well as design of controllers. Simulink allows modeling and simulation of linear, non-linear and discrete systems.

Description of laboratory works is for use in the laboratory. The book allows to get deeper knowledge of automatic control theory, to understand processes of the system using modern and attractive software.

The main knowledge about MATLAB and Simulink

MATLAB high-performance language for technical computing integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation and includes:

- mathematics and computation;
- algorithm development;
- data acquisition;
- modeling, simulation, and prototyping;
- data analysis, exploration, and visualization;
- scientific and engineering graphics;

- application development, including graphical user interface building.

MATLAB is an interactive system the basic data element of which is an array that does not require dimensioning.

The name MATLAB stands for *matrix laboratory*. MATLAB was originally written to provide easy access to matrix software developed by LINPACK and EISPACK projects. Today MATLAB engines incorporate LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called *toolboxes*. Toolboxes are comprehensive collections of MATLAB functions that extend the MATLAB environment to solve particular classes of problems. You can add on toolboxes for signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many other areas.

MATRIXES

In MATLAB environment, a matrix is a rectangular array of numbers. Special meaning is sometimes attached to 1-by-1 matrices, which are scalars, and to matrices with only one row or column, which are vectors. MATLAB has other ways of storing both numeric and nonnumeric data, but in the beginning, it is usually best to think of everything as a matrix.

Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

Matrices can be entered into MATLAB in several different ways:

1. Enter an explicit list of elements.
2. Load matrices from external data files.
3. Generate matrices using built-in functions.
4. Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You only have to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix:

```
A =
    16     3     2    13
```

```
5    10    11     8
9     6     7    12
4    15    14     1
```

This matrix matches the numbers in the engraving. Once the matrix is entered, it is automatically remembered in MATLAB workspace. It can be referred to simply as `A`.

Commands sum, transpose, and diag

You are probably already aware that the special properties of a magic square have to do with the various ways of summing its elements. If you take the sum along any row or column, or along either of the two main diagonals, you will always get the same number. Let us verify that using MATLAB. The first statement to try is:

```
sum(A)
```

MATLAB gives

```
ans =
    34    34    34    34
```

When you do not specify an output variable, MATLAB uses the variable ans, short for *answer*, to store the results of a calculation. You have computed a row vector containing the sums of the columns of A. Each of the columns has the same sum, the *magic* sum, 34.

The sum of the elements on the main diagonal is obtained with the `sum` and the `diag` functions:

diag(A)

produces

ans =
  16
  10

```
     7
     1
```

and

sum(diag(A))

produces

```
ans =
    34
```

MATLAB software provides four functions that generate basic matrices:

`zeros`	All zeros
`ones`	All ones
`rand`	Uniformly distributed random elements
`randn`	Normally distributed random elements

Here are some examples:

```
Z = zeros(2,4)
Z =
     0     0     0     0
     0     0     0     0

F = 5*ones(3,3)
F =
     5     5     5
     5     5     5
     5     5     5

N = fix(10*rand(1,10))
N =
     9     2     6     4     8     7     4     0     8     4

R = randn(4,4)
R =
```

0.6353	0.0860	-0.3210	-1.2316
-0.6014	-2.0046	1.2366	1.0556
0.5512	-0.4931	-0.6313	-0.1132
-1.0998	0.4620	-2.3252	0.3792

Subscripts

The element in row `i` and column `j` of `A` is denoted by `A(i,j)`. For example, A(4,2) is the number in the fourth row and the second column. For the magic square, A(4,2) is 15. So to compute the sum of the elements in the fourth column of A, type
A(1,4) + A(2,4) + A(3,4) + A(4,4)

This subscript produces

```
ans =
    34
```

but is not the most elegant way of summing a single column.

It is also possible to refer to the elements of a matrix with a single subscript, A(k). A single subscript is the usual way of reference row and column vectors.

The Colon Operator

The colon, :, is one of the most important MATLAB operators. It occurs in several different forms. The expression

```
1:10
```

is a row vector containing the integers from 1 to 10:

1 2 3 4 5 6 7 8 9 10

To obtain nonunit spacing, specify an increment. For example,

```
100:-7:50
```

is

100 93 86 79 72 65 58 51

and

```
0:pi/4:pi
```

is

0 0.7854 1.5708 2.3562 3.1416

Subscript expressions involving colons refer to portions of a matrix:

```
A(1:k,j)
```

is the first `k` elements of the `j`th column of `A`. Thus:

```
sum(A(1:4,4))
```

computes the sum of the fourth column and gives

```
ans =
     34
```

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

```
num_students = 25
```

creates a 1-by-1 matrix named num_students and stores value 25 in its single element. To view the matrix assigned to any variable, simply enter the variable name.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are *not* the same variable.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the *long* format specified by the IEEE® floating-point standard.

MATLAB software stores the real and imaginary parts of a complex number. It handles the magnitude of the parts in different ways depending on the context. For instance, the `sort` function sorts based on magnitude and resolves ties by phase angle.

```
sort([3+4i, 4+3i])
ans =
   4.0000 + 3.0000i   3.0000 + 4.0000i
```

The phase angle is calculated as:

```
angle(3+4i)
ans =
    0.9273
angle(4+3i)
ans =
    0.6435
```

The magnitude is calculated as:

```
abs(3+4i)
ans =
       5
```

Functions

MATLAB provides a large number of standard elementary mathematical functions, including `abs`, `sqrt`, `exp` and `sin`. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type

```
help elfun
```

For a list of more advanced mathematical and matrix functions, type

```
help specfun
help elmat
```

Some of the functions, like `sqrt` and `sin`, are *built in*. Built-in functions are part of MATLAB core, so they are very efficient. Other functions, like `gamma` and `sinh`, are implemented in M-files.

There are some differences between built-in functions and other functions. For example, for built-in functions, you cannot see the code. For other functions, you can see the code and even modify it if you want.

Several special functions provide values of useful constants.

`pi`	3.14159265...
`i`	Imaginary unit,
`j`	Same as `i`

The Colon Operator

The colon, :, is one of the most important MATLAB operators. It occurs in several different forms. The expression

```
1:10
```

is a row vector containing the integers from 1 to 10:

1 2 3 4 5 6 7 8 9 10

To obtain nonunit spacing, specify an increment. For example,

```
100:-7:50
```

is

```
100    93    86    79    72    65    58    51
```

and

```
0:pi/4:pi
```

is

```
0    0.7854    1.5708    2.3562    3.1416
```

Subscript expressions involving colons refer to portions of matrix:

```
A(1:k,j)
```

is the first k elements of the jth column of A. Thus:

```
sum(A(1:4,4))
```

computes the sum of the fourth column. However, there is a better way to perform this computation. The colon by itself refers to *all* the elements in a row or column of a matrix and the keyword `end` refers to the *last* row or column. Thus:

```
sum(A(:,end))
```

computes the sum of the elements in the last column of A:

```
ans =
     34
```

Building Tables

Array operations are useful for building tables. Suppose n is the column vector

```
n = (0:9)';
```

Then

```
pows = [n  n.^2  2.^n]
```

builds a table of squares and powers of 2:

```
pows =
   0    0    1
   1    1    2
   2    4    4
   3    9    8
   4   16   16
   5   25   32
   6   36   64
   7   49  128
   8   64  256
   9   81  512
```

The elementary math functions operate on arrays element by element. So format

```
short g
x = (1:0.1:2)';
logs = [x log10(x)]
```

builds a table of logarithms.

```
logs =
    1.0         0
    1.1   0.04139
    1.2   0.07918
    1.3   0.11394
```

1.4	0.14613
1.5	0.17609
1.6	0.20412
1.7	0.23045
1.8	0.25527
1.9	0.27875
2.0	0.30103

Representing Polynomials

MATLAB software represents polynomials as row vectors containing coefficients ordered by descending powers. For example, consider the equation

$$p(x) = x^3 - 2 \cdot x - 5$$

To enter this polynomial into MATLAB, use the row of descending coefficients

```
p = [1 0 -2 -5];
```

Evaluating Polynomials

The `polyval` function evaluates a polynomial at a specified value. To evaluate p at $x = 5$, use

```
polyval(p,5)

ans =
   110
```

Roots

The roots function calculates the roots of a polynomial:

```
r = roots(p)
```

```
r =
      2.0946
     -1.0473 +      1.1359i
     -1.0473 -      1.1359i
```

By convention, the MATLAB software stores roots in column vectors. The function `poly` returns to the polynomial coefficients:

```
p2 = poly(r)

p2 =
     1   8.8818e-16   -2   -5
```

poly and roots are inverse functions, up to ordering, scaling, and roundoff error.

MATLAB Commands List

The following list of commands can be very useful for future reference. Use `help` in MATLAB for more information on how to use the commands.
Usualy commands both from MATLAB and from Control Systems Toolbox, as well as some commands/functions which we wrote ourselves are used. For those commands/functions which are not standard in MATLAB, we give links to their descriptions. For more information on writing MATLAB functions, see the `function` page.

Table 1. Command list

Command	**Description**
Abs	Absolute value
axis	Set the scale of the current plot, see also plot, figure
bode	Draw the Bode plot, see also logspace, margin, nyquist1

conv	Convolution (useful for multiplying polynomials), see also deconv
det	Find the determinant of a matrix
eig	Compute the eigenvalues of a matrix
eps	MATLAB's numerical tolerance
figure	Create a new figure or redefine the current figure, see also subplot, axis
for	For, next loop
format	Number format (significant digits, exponents)
function	Creates function m-files
grid	Draw the grid lines on the current plot
help	HELP!
hold	Hold the current graph, see also figure
if	Conditionally execute statements
imag	Returns the imaginary part of a complex number, see also real
impulse	Impulse response of linear systems, see also step, lsim
input	Prompt for user input
inv	Find the inverse of a matrix
legend	Graph legend
length	Length of a vector, see also size
linspace	Returns a linearly spaced vector
lnyquist	Produce a Nyquist plot on a logarithmic scale, see also nyquist1
log	Natural logarithm, also log10: common logarithm
lsim	Simulate a linear system, see also step, impulse
margin	Returns the gain margin, phase margin, and crossover frequencies, see also bode
nyquist1	Draw the Nyquist plot, see also lnyquist. Note this command was written to replace MATLAB standard command nyquist to get more accurate Nyquist plots

ones	Returns a vector or matrix of ones, see also zeros
plot	Draw a plot, see also figure, axis, subplot.
poly	Returns the characteristic polynomial
polyval	Polynomial evaluation
print	Print the current plot (to a printer or postscript file)
pzmap	Pole-zero map of linear systems
rank	Find the number of linearly independent rows or columns of a matrix
real	Returns the real part of a complex number, see also imag
rlocus	Draw the root locus
roots	Find the roots of a polynomial
set	Set(gca,'Xtick',xticks,'Ytick',yticks) to control the number and spacing of tick marks on the axes
sqrt	Square root
Ss	Create state-space models or convert LTI model to state space, see also tf
ssdata	Access to state-space data. See also tfdata
step	Plot the step response, see also impulse, lsim
text	Add a piece of text to the current plot, see also title, xlabel, ylabel, gtext
Tf	Creation of transfer functions or conversion to transfer function, see also ss
tfdata	Access to transfer function data, see also ssdata
title	Add a title to the current plot
xlabel/ylabel	Add a label to the horizontal/vertical axis of the current plot, see also title, text, gtext
zeros	Returns a vector or matrix of zeros

Introduction to MATLAB Functions

When entering a command such as `roots`, `plot`, or `step` into MATLAB what you are really doing is running an *m-file* with inputs and outputs that has been written to accomplish a specific task. These types of m-files are similar to subroutines in programming languages in that they have inputs (parameters which are passed to the m-file), outputs (values which are returned from the m-file), and a body of commands which can contain local variab. MATLAB calls these m-files **functions**. You can write your own functions using `function` command.

A new function must be given a filename with a '.m' extension. This file should be saved in the same directory as MATLAB software, or in a directory which is contained in MATLAB's search path. The first line of the file should contain the syntax for this function in the form:

Function [output1, output2] = filename (input1, input2, input3)

A function can input or output as many variables as are needed. The next few lines contain the text that will appear when the `help filename` command is evoked. These lines are optional, but must be entered using `%` in front of each line in the same way that you include comments in an ordinary m-file.

Below is a simple example of what the function, `add.m`, might look like.

```
function [var3] = add(var1,var2)
% add is a function that adds two numbers
var3 = var1+var2
```

If you save these three lines in a file called "add.m" in MATLAB directory, then you can use it by typing at the command line:

```
y = add(3,8)
```

Obviously, most functions will be more complex than the one demonstrated here. This example just shows what the basic form looks like. Look at the functions created for this tutorial listed below, or at the functions in the toolbox folder in MATLAB software (for more sophisticated examples), or try `help function` for more information.

COMPOSITION OF DYNAMIC SYSTEMS IN MATLAB

Laboratory work No. 1

Objective: Learn to describe in MATLAB system transfer function and calculate responses to typical input signals.

Tasks of the work:

1. Make a model of a system, indicated by the teacher.
2. Derive transfer function of the system, indicated by the teacher.
3. Apply MATLAB to describe transfer function of item 2 by ratio of polynomials and zeros–poles form.
4. Find system response to input signal, indicated by the teacher.

Theoretical part

The first step of automatic control system analysis is development of system dynamic model. Dynamic model here is assumed as model, describing dynamics of the system. This Automatic system theory course deals with linear time invariant systems, whose models are described by linear differential equations.

Development of dynamic model allows simulation of system, i.e obtain solutions of system differential equations.

Differential equations can be solved by MATLAB in some ways:

1. Equations should be rearranged expressing those in form $\dot{x} = f(x,t)$ and then solving by numerical methods by integrating.
2. Use specialized tools for analysis of automatic control analysis: ***ltiview*** ir ***Control system toolbox***.
3. Use special package Simulink developed for simulation of dynamic systems.

The first way is most complex, but it is most general. It can be applied for linear and non-linear systems and is suitable not only for MATLAB, but also for similar programs, for example, ***Scilab***. It has only one shortcoming: its application requires good skills in programming, therefore it will not be considered here. We will deal with the last ones.

Control system toolbox is collection of MATLAB programs for analysis and synthesis of automatic control systems. The mentioned ***ltiview*** also enters as part MATLAB. ***Control system toolbox*** is also widely used for linear systems.

Simulink is visual environment for simulation of dynamic systems. It is applied for simulation of linear, nonlinear and discrete systems. Its application for linear systems and non-linear systems will be described.

The linear differential equation looks like this:

$$\sum_{k=0}^{n} a_k \frac{d^k y}{dt^k} = \sum_{l=0}^{m} b_l \frac{d^l u}{dt^l}; \text{ kur } \quad n \geq m. \tag{1}$$

where: a_n ir b_n – constants; y – system output; u – system input.

Expression (1) can be rewritten in operational form:

$$Y \sum_{k=0}^{n} a_k s^k = U \sum_{l=0}^{m} b_l s^l; \quad \text{kur} \quad n \geq m. \tag{2}$$

Although linear differential equations can be solved analytically, automatic control theory uses definition of transfer function. Transfer function is defined as ratio of output and input signals in operational form:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{l=0}^{m} b_l s^l}{\sum_{k=0}^{n} a_k s^k}. \tag{3}$$

Transfer function does not depend on output and input signals, therefore it is a very convenient tool to describe any control system. Using ***Control system toolbox*** tool transfer function is developed with **`tf`** command. Its generalized form looks like that:

```
G_plant = tf(sk_koef_vect, var_ koef_vect, T_s)
```

where variables *`sk_koef_vect`* and *`var_ koef_vect`* – are vectors of transfer function numerator and denominator, *`T_s`* – sampling time of discrete system, which can be missed for continuous time system transfer function.

Denominator of expression (3) set equal to zero, i.e. $\sum_{k=0}^{n} a_k s^k = 0$, is called characteristic equation of the system. Let $p_1, p_2, ..., p_n$ be roots of characteristic equation, named as poles of transfer function. It is known from mathematics course, any n-th order characteristic equation can be expanded to the product of first order polynomials:

$$\sum_{k=0}^{n} a_k s^k = \prod_{k=0}^{n} (s - p_k) \tag{4}$$

The numerator of transfer function (3) can be expressed in the same way. Its roots are called transfer function zeros and denoted $n_1, n_2, ..., n_m$. Then

$$\sum_{l=0}^{m} b_l s^l = \prod_{l=0}^{m} (s - n_l) \tag{5}$$

Expression of transfer function by product of first order polynomials has its own advantages and shortcomings, for example, facilitates application of root locus method, therefore ***Control system toolbox*** allows to express system transfer function by zero-pole form:

$$W = K \frac{\prod_{l=0}^{m} (s - n_l)}{\prod_{k=0}^{n} (s - p_k)}. \tag{6}$$

For this purpose command `zpk` is used, having general expression as:

```
G_plant = zpk(zero_vect, poles_vect, K, T_s)
```

where variables *zero_vect* ir *poles_vect* are vectors of transfer function zeros and poles; *k* – system gain.

Any system can be expressed also in the third way: by state space variables. In general, any system can be described by state space variables in this way:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}; \\ \mathbf{y} = \mathbf{Cx} + \mathbf{Du}; \end{cases} \tag{7}$$

where: **A**, **B**, **C**, **D** are system inputs, outputs and transfer matrixes correspondingly; **x** is vector of state variables; **u** and **y** are system input and output signal vectors correspondingly. The system (5) described by state space variables can be converted to transfer function by command `ss`:

```
G_plant = ss(A, B, C, D, T_s)
```

It is important to note, that commands of system creation can be already constructed system. These commands are used to express already created systems or plants in other form. For example, if it is given some plant or system *W_1*, which can be described by expression (6) using `zpk` command, and it is desired to get its description by transfer function form (3), the command `tf(W_1)` can be used and MATLAB will calculate corresponding transfer function. The model allows analysis of the system. In general, the response to unit step, impulse function and ramp signal are considered. The most popular input signal is unit step **1**(t) and Dirac impulse. Step response is calculated by `step` command, having these forms:

```
Step (G_plant)
Step (G_plant, t_vect)
[y_vect, t_vect] = step (G_plant)
y_vect = step (G_plant, t_vect)
```

The first two commands draw system step response in separate graphical window. The first one chooses time interval automaticaly, the second one uses time interval *t_vect* indicated by user as variable. The last two commands return values, therefore they are used in creating of programs, where the numerical values are required. In the same way `impulse` command is used, which calculates impulse response of the system.

If the system response to unconventional signal is required, `lsim` command I is applied. Its form looks like this:

```
lsim(G_plant, u_vect, t_vect)
y_vect = lsim(G_plant, u_vect, t_vect)
```

where variables *u_vect* ir *t_vect* are input signal and time values vectors correspondingly. The first command presents output in graphical window, the second – by numerical values.

For generation of input signal vector *u_vect* values `gensig` command can be used. It is designated for generation of sine, rectangular or repeating pulses. Command form is:

```
[u_vect, t_vect] = gensig (type, period)
```

Variable *type* is text variable, indicating type of selected to general signal. It can be expressed as: 'sin' for generation of sinusoidal signal; 'square' for rectangular impulses; 'pulse' for generation of repeating Dirac impulses. As all generated signals are periodical, repeating period is designated by variable *period*.

Examples of dynamic system modeling and analysis

1. Construct mathematical model of the electric circuit shown in Fig. 1 and get its time response. Voltage u is system input signal and voltage u_2 is output signal.

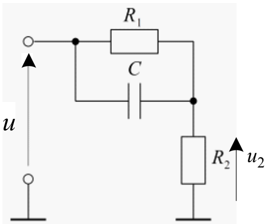

Fig. 1. Analyzed circuit

In general case mathematical model of the electrical system is constructed using the main laws of electrical engineering: Ohm's law and Kirchoff's laws. Application of Kirchoff's laws for this circuit requires solving of 3 equations. In a more simple way this circuit can be considered as voltage divider and for voltage u and u_2 we can apply voltage division rule. Resistance R_1 is connected in parallel with capacitor C, therefore their equivalent impedance in frequency domain can be calculated as:

$$Z_1(s) = \frac{\frac{R_1}{Cs}}{R_1 + \frac{1}{Cs}} = \frac{R_1}{R_1 Cs + 1}. \tag{8}$$

System transfer function is equal to division ratio and is calculated as:

$$G(s) = \frac{U_2(s)}{U(s)} = \frac{R_2}{Z_1(s) + R_2} = \frac{R_2}{\frac{R_1}{R_1 Cs + 1} + R_2} = \frac{R_2\left(R_1 Cs + 1\right)}{R_2 R_1 Cs + R_2 + R_1}. \tag{9}$$

System model in MATLAB is constructed by `tf` command. Let R_1 = 1 kΩ, R_2 = 5 kΩ and C = 1 μF. Then MATLAB command, describing the circuit in Fig. 1 looks like this:

```
>> R_1 = 1e3; R_2 = 5e3; C = 0.5e-6;
>> W = tf(R_2*[R_1*C, 1],[R_2*R_1*C, R_1+R_
2])

Transfer function:
2.5 s + 5000
------------
2.5 s + 6000
```

Expression of this function in zero-pole form is given as:

```
>> zpk(G)

Zero/pole/gain:
(s+2000)
--------
(s+2400)
```

System unit step response is calculated by `step` command. Unit step response is given in Fig. 2. It can be seen from Fig.2, that the system is composed from real differentiating and first order (exponential) elements.

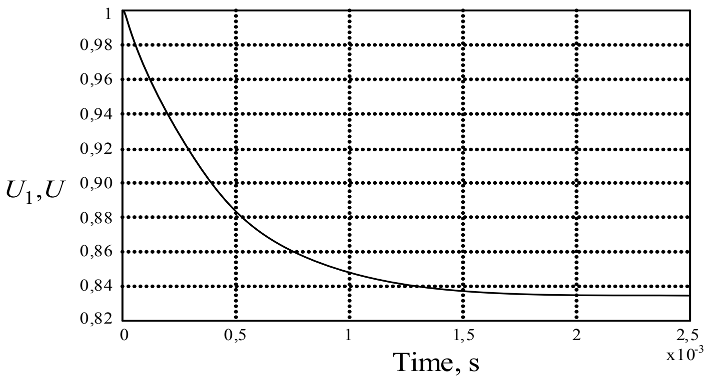

Fig. 2. Unit step response of the system

In the same way system response to impulse or other shape signals can be obtained.

Example **2.** Develop mathematical model of the circuit, shown in Fig. 3 in the form of transfer function, when input signal is voltage u_1 and output is voltage u_2. Assume operational amplifier being ideal.

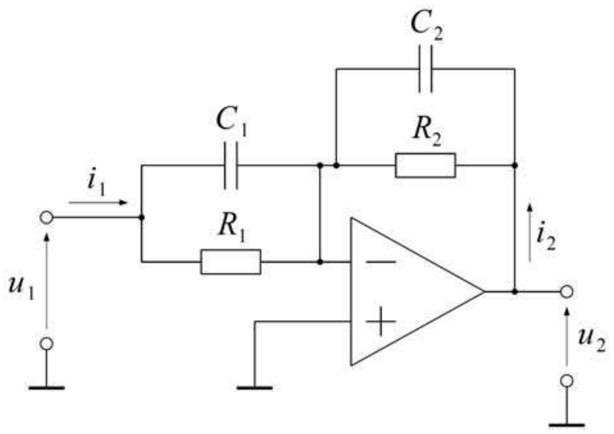

Fig.3. Analyzed circuit

Assumption of ideal operational amplifier facilitates analysis of the system. At first, the gain of ideal op-amp does not depend on frequency and approaches to infinity: $k_{oa}\to\infty$. Else, input impedance of op-amp is equal to infinity, threrefore currents do not flow for direct either inverse inputs. With these assumptions analysis of circuit simplifies and we can apply the so called "virtual ground" principle. With $k_{oa}\to\infty$ we can assume $u_+\approx u_-$, where u_+ voltage of is direct and u_- is inverse input voltage. From Fig. 3 it can be seen, that $u_+=0$, therefore we can state, that $u_-\approx 0$ V, and all circuits, connected to that inverse input, can be considered as virtually connected to the ground.

Parrallel connected impedances R_1C_1 ir R_2C_2 can be written in operational form as:

$$Z_x(s)=\frac{R_x\dfrac{1}{C_xs}}{R_x+\dfrac{1}{C_xs}}=\frac{R_x}{R_xC_xs+1};\tag{11}$$

where x is element index: $x=\{1,2\}$.

While current does not flow to inverse input, application of the first Kirchhoff's law for virtual ground node, i. e. for node to which inverse input is connected, can be written as:

$$I_1=-I_2\leftrightarrow U_1\frac{R_1C_1s+1}{R_1}=-U_2\frac{R_2C_2s+1}{R_2}\tag{12}$$

Then system transfer function is:

$$W(s)=\frac{U_2}{U_1}=-\frac{R_1C_1s+1}{R_2C_2s+1}\cdot\frac{R_2}{R_1}.\tag{13}$$

Content of the report:

1. Develop mathematical model, indicated by the teacher.
2. Commands, creating simulation model *MATLAB* software, required to create simulation model in transfer function and zero-pole form.
3. Commands, giving response to the input signals, indicated by the teacher.
4. Plots of system response to the indicated test signals.

Control questions:

1. What are the means mostly used for analysis of dynamic systems?
2. What are system zeros and poles?
3. What is the difference between `tf` and `zpk` commands?
4. What are typical input signals of the analysed system?
5. What response does command `impulse` generate?

FREQUENCY RESPONSE OF DYNAMIC SYSTEMS

Laboratory work No. 2

Objective: to get acquainted with definition of frequency response, its calculation and visualization methods**.**

Tasks of the laboratory work:

1. Derive transfer function, indicated by the teacher.
2. Plot system frequency response and Bode plot of this system.
3. Define system gain and phase angle at frequency, indicated by the teacher.

Teoretical part

Linear dynamic systems in general case are described by differential equations in this way:

$$\sum_{k=0}^{n} a_k \frac{d^k y}{dt^k} = \sum_{l=0}^{m} b_l \frac{d^l u}{dt^l}; \tag{1}$$

where: $n \geq m$, a_n ir b_n are constants; y is system output; u is system input.

At known input signal *u,* describing function in time domain $u = f_u(t)$ Eq. 1 can be solved in analytical way to get analythical function, describing output signal. This analysis method, operating with functions, depending on time, is called time domain analysis. Solution of high order differential equations even n ≥ 3 is rather complicated problem, therefore analysis in time domain spread only after fast and powerful computers have spread.

Together with systems analysis in time domain, the other method, analysis in frequency domain, is widely used. This metod assumes that system input is harmonic signal, changing by sine law:

$$u = U \sin \omega t \qquad (2)$$

where: U is input signal amplitude, ω is signal frequency in radians per second.

The following frequency-domain plots of G (jω) versus are often used in the analysis and design of linear control systems in the frequency domain.

1. Polar plot. A plot of the magnitude versus phase in the polar coordinates is varied from zero to infinity.

2. Bode plot. A plot of the magnitude in decibels versus (or log10) in semilog (or rectangular) coordinates.

3. Magnitude-phase plot. A plot of the magnitude (in decibels) versus the phase on rectangular coordinates, with a variable parameter on the curve.

The polar plot of a function of the complex variable *s*, *G* (*s*), is a plot of the magnitude of *G* (*jω*) versus the phase of *G* (*jω*) on polar coordinates as ω is varied from zero to infinity.

To illustrate the construction of the polar plot of a function G (s), consider the function

$$G(s) = \frac{1}{Ts + 1} \qquad (3)$$

where T is positive constant.

Substuting s = $j\omega$ yields:

$$G(j\omega) = \frac{1}{Tj\omega + 1} \qquad (4)$$

Eq. 4 can be rewritten in terms of magnitude and angle as:

$$G(j\omega) = \frac{1}{\sqrt{T^2\omega^2 + 1}} \angle - \tan^{-1} \omega T \qquad (5)$$

When ω is zero, the magnitude of G ($j\omega$) is unity, and the phase of G ($j\omega$) is at 0°. Thus, at ω = 0, G ($j\omega$) is represented by a vector of unit length directed in the 0° direction. As ω increases, the magnitude of G (jω) decreases, and the phase becomes more negative. As ω increases, the length of the vector in the polar coordinates decreases, and the vector rotates in the clockwise (negative) direction. When ω approaches infinity, the magnitude of G (jω) becomes zero, (can be assumed of very small magnitude) and the phase reaches –90°. By substituting other values of ω into Eq. (4), the exact plot of G (jω) seems as semicircle, as shown in Fig. 2.

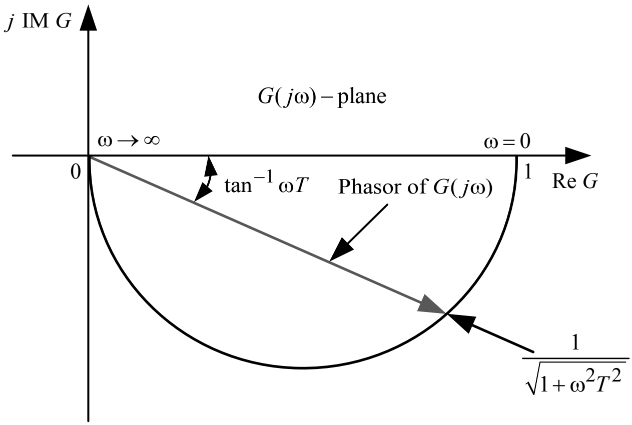

Fig.2. Polar plot of function $G(j\omega) = \dfrac{1}{Tj\omega + 1}$

Consider the function:

$$G(j\omega) = \frac{1}{Tj\omega + 1} \tag{6}$$

where T_1 and T_2 are positive real constants. Equation (6) can be rewritten as this:

$$G(j\omega) = \frac{1+\omega^2 T_2^2}{1+\omega^2 T_1^2} \angle(\tan^{-1} \omega T_2 - \tan^{-1} \omega T_1) \tag{7}$$

The polar plot of G (jω), in this case, depends on the relative magnitudes of T_1 and T_2. If T_2 is greater than T_1, the magnitude of G (jω) is always greater than unity as ω is varied from zero to infinity, and the phase of G (jω) is always positive. If T_2 is less than T_1, the magnitude of G (jω) is always less than unity, and the phase is always negative. The polar plots of G (jω) of Eq. (7) corresponding to both two mentioned conditions are shown in Fig. 3.

Application of software MATLAB arba *Octave* allows calculation of polar plots directly replacing Laplace operator *s* by jω. The specialized commands can be used to calculate that if MATLAB ***Control system toolbox*** is used. Frequency response is calculated by command `freqresp`, which has the form:

```
H = freqresp(G_plant,w_vect)
```

where variable `G_plant` is model of analyzed plant, described by transfer function, zero-pole-gain or state space variables, `w_vect` is vector of frequency values, at which the function is calculated, `H` is threedimentional vector for saving of polar plot values.

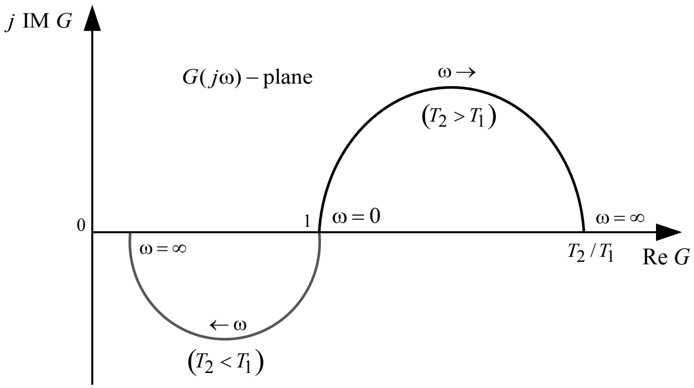

Fig. 3. Polar plots of $G(j\omega) = \dfrac{1 + j\omega T_2}{1 + j\omega T_1}$

In general way `H = freqresp(sys,w)` computes the frequency response H of the linear time invariant (LTI) model sys at the frequencies specified by the vector w.

These frequencies should be real and expressed in radians/second.

Polar plots, shown in Fig. 5 and Fig.6 usually are called hodograph and are used mainly for analysis of system stability.

The other possible analysis method calculates dependence of magnitude |G (jω)| and phase angle $\angle G(j\omega)$ on frequency. Boths plots can be represented separately.

If magnitude of the system is expressed by decibels

$$L(\omega) = 20\log_{10}\left[|G(j\omega)|\right] \tag{12}$$

and plotted versus $\log_{10}\omega$ or ω, and phase also is plotted versus $\log_{10}\omega$ or ω, we will have two plots, called as Bode plots.

If the logarithmic scale is used for frequency, then plotted dependence of magnitude versus frequency or angle versus frequency is called Bode plots.

Bode plot is also known as a corner plot or an asymptotic plot of G (jω). These names came from the possibility to construct the Bode plot by using straight-line approximations that are asymptotic to the actual plot.

The Bode plot has the following features:

1. Since the magnitude of G (jω) in the Bode plot is expressed in dB, product and division factors in G (jω) became additions and subtractions, respectively. The phase relations are also added and subtracted from each other algebraically.
2. The magnitude plot of the Bode plot of G (jω) can be approximated by straightline segments, which allow the simple sketching of the plot without detailed computation.

Application of MATLAB for construction of Bode plots allows using standard graphs plotting commands. Command `plot` can be used for the calculated dependence of magnitude versus ω.

Before command `plot` commands `abs` and `angle` are used. Then magnitude is expressed by decibels according to Eq. (12) and using command `semilogx` the magnitude and angle versus ω are plotted in semilogarithmic scale.

Example of commands application:

```
plot(abs(G_jw))
plot(angle(G_jw))
```

Sometimes the mentioned command `angle` meets difficulties at calculation of phase angle of higher order transfer functions.

In this case it is usefull to apply command `Bode` from ***Control system toolbox*** to get Bode plots:

```
bode(G_plant)
bode(G_plant, w_vect)
```

Table 1. MATLAB commands

N	MATLAB (*Octave*) commands
8.	`omega_vect = logspace(-2,6,1000)`
2	`s_vect = 1j*omega_vect`
3	`G_jw = 1./(0.2*s_vect+1)`
	`plot(G_jw)`

`Logspace` plots logarithmically spaced vector. Command `logspace(x1, x2)` generates a row vector of 50 logarithmically equally spaced points between decades 10^X1 and 10^X2. If X2 is pi, then the points are between 10^X1 and pi. Command `logspace(x1, x2, n)` generates N points. For `n` < 2, `logspace` returns to 10^X2. Command is used in this way:

```
y_vect = logspace(a,b,n)
```

Command in the second row of Table 1 executes replacement s = jω described in expression (8). At least the expression of frequency response according to (15) is calculated in the third row. The obtained numerical values of frequency response are designated in the complex plane. For this MATLAB command is used: `plot(G_jw)`. Command `plot` gives graph, shown in Fig. 1. It is called polar plot.

Table 2. MATLAB commands

N	MATLAB (Octave) commands
1	`L = 20*log10(abs(G_jw))`
2	`fi = angle(G_jw)*180/pi`
3	`semilogx(omega_vect, L, omega_vect, fi)`

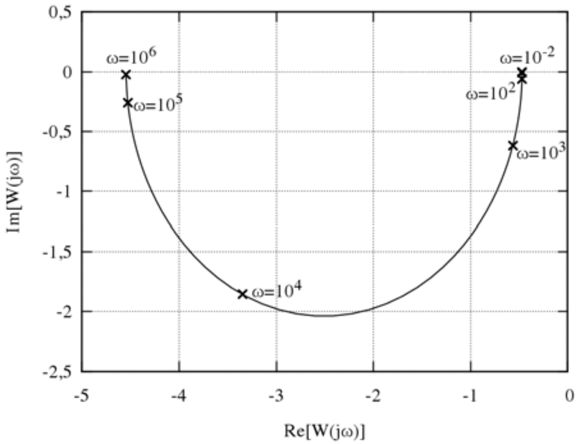

Fig.1 . Polar plot of frequency response

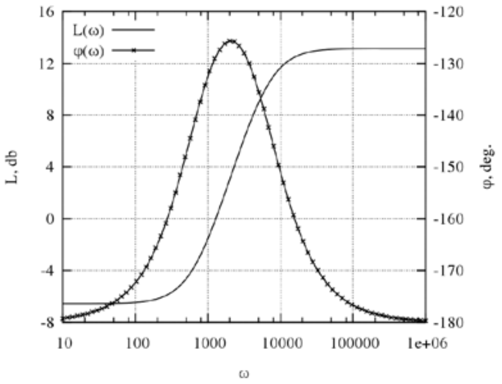

Fig. 2. Bode plot

There are some other commands to plot frequency response. Command `bode(sys)` draws the Bode plot of the LTI model sys (created with either tf, zpk, ss, or frd). The frequency range and number of points are chosen automatically.

`bode(sys,{wmin,wmax})` draws the Bode plot for frequencies between `wmin` and `wmax` (in radians / second).

`bode(sys,w)` uses the user-supplied vector `w` of frequencies, in radian / second, at which the Bode response is to be evaluated.

`bode(sys1,sys2,...,w)` graphs the Bode response of multiple LTI models `sys1,sys2,...` on a single plot. The frequency vector `w` is optional. You can specify a color, line style, and marker for each model, as in

`bode(sys1,'r',sys2,'y--',sys3,'gx').`

Content of the report

1. Elaboration of model, indicated by teacher.
2. Commands for calculation of frequency response and plots of frequency response.
3. Bode plots.

4. Results of calculation of gain and phase angle for the indicated frequency.

Control questions

1. How are transfer function and analytical expression of frequency response related?
2. What are the main ways to depict frequency response?
3. What is the physical meaning of frequency reponse magnitude and phase angle?
4. What does the Bode plot represent?

THE FIRST ORDER DYNAMIC SYSTEMS

Laboratory work No. 3

Objectives: to get acquainted with the first order dynamic systems and learn to analyze their characteristics in time and frequency domain.

Tasks:

1. Elaborate mathematical modul of the system, indicated by the teacher.
2. Expand the elaborated model to series connection of the first order systems, if it is necessary.
3. According to differential equations plot step responses and frequency responses.
4. Check results with Matlab software.

Theoretical part

The simplest transfer function is described by equation:

$$y = Ku; \tag{1}$$

where: K is gain of the system.

Time domain analysis of Eq. (1) shows, that unit step response will have the same shape as unit step $1(t)$, but its amplitude will increase by K times.

$$y_p = K1(t). \tag{2}$$

Plot of unit step $1(t)$ and response y is shown in Fig. 4.

Transfer function of this system is:

$$G_P = K. \tag{3}$$

Frequency response of the system $G(j\omega) = K + 0j$.

Plot $G(j\omega)$ is given in Fig. 5.

Bode plot of this function is calculated as:

$$L(\omega) = 20\lg\left|G(j\omega)\right| = 20\lg K; \tag{4}$$

and does not depend on frequency. The Bode plot $L(\omega)$ is shown in Fig. 6. The phase angle of proportional system:

$$\varphi = \tan^{-1}\frac{\operatorname{Im} G(j\omega)}{\operatorname{Re} G(j\omega)} = \tan^{-1} 0 = 0; \tag{5}$$

is horizontal straight, shown in Fig. 7.

The simplest electrical examples of propotional system are shown in Fig. 1 a – voltage divider and Fig. 1 b – amplifier.

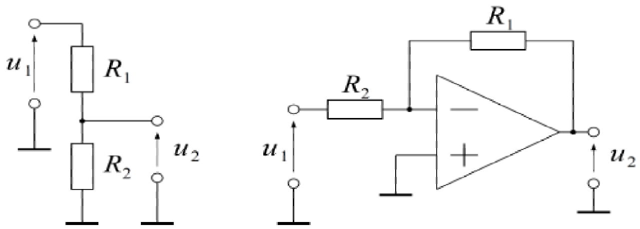

Fig.1. Example of projectional system

Output voltage U_2 of voltage divider shown in Fig. 1 is calculated.

$$u_2 = \frac{R_2}{R_1 + R_2} u_1. \tag{6}$$

Gain of the system depends on resistances R_1 and R_2 and changes in the range $0 \le K \le 1$.

The system with operational amplifier gives possibility to change output voltage. For the ideal operational amplifier principle of virtual ground can be applied which gives:

$$\frac{u_1}{R_2} = -\frac{u_2}{R_1}; \rightarrow u_2 = -\frac{R_1}{R_2} u_1. \tag{7}$$

Therefore, varying ratio of R_1/R_2 gives possibility to change gain in wide range.

The propotional system has no inertic, nevertheless all real systems have that. Therefore propotional system is idealized real system. As example the real oscillograme of the operational amplifier, shown in Fig. 2, corresponding to the step response of amplifier, is shown in Fig. 1 a.

In the circuit operational amplifier LM 741 and resistances $R_1 = R_2 = 10$ kΩ. were used.

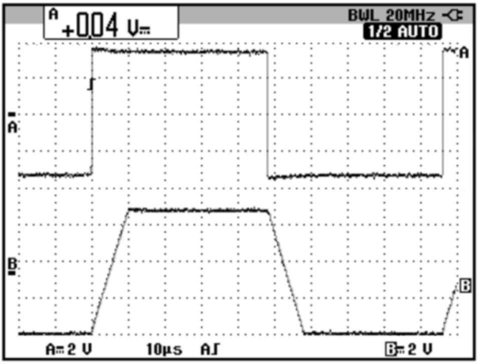

Fig. 2. Impulse response of amplifier

The upper curve in Fig. 2 corresponds to input signal and the lower curve corresponds to impulse response of amplifier. According to expression (7), gain $K = -1$, therefore the inverted output signal is shown in Fig. 2 to facilitate comparison of output and input signals. The amplifier was supplied by ≈ 10 kHz rectangular impulses. The output signal has the front shape with slope settling time of impulse response is about $10\ \mu s$, which depends on inertic of operational amplifier.

The integrating circuit is described as:

$$y = K_I \int_0^t u dt + y_0; \quad \to \quad \frac{dy}{dt} = K_I u; \tag{8}$$

where: K_I – coefficient, y_0 – initial value of output signal.

Step response of integrating systems is:

$$y_I = K_I \int_0^t 1(0) dt + y_0 = K_I t + y_0. \tag{9}$$

Expression (9) describes straight line, which in Fig. 4 is marked by y_1.

Assuming $y_0 = 0$ and applying for Eq. (8) Laplace transform, the transfer function of integrating system is:

$$Y(s) = \frac{K_I}{s} U(s); \quad \to \quad G_I = \frac{K_I}{s}. \tag{10}$$

Frequency response of integrator is:

$$G_I(j\omega) = \frac{K_I}{j\omega} = -j\frac{K_I}{\omega}. \tag{11}$$

Expression (11) indicates the linear frequency response. Straight line begins at point $G_I(0j) = -j\infty$ and ends at point $G_I(j\infty) = 0$. Polar plot of integrating system is shown in Fig. 5. The Bode plot of the integrating circuit is calculated as:

$$L_I(\omega) = 20\lg\left|G_I(j\omega)\right| = 20\lg\frac{K_I}{\omega} = 20\lg K_I - 20\lg\omega. \tag{12}$$

Expression (12) is description of straight line. Increasing the frequency 10 times yields:

$$\begin{aligned} L_I(10\omega) &= 20\lg K_I - 20\lg 10\omega = 20\lg K_I - 20\lg\omega - 20 = \\ &L_I(\omega) - 20. \end{aligned} \tag{13}$$

Comparison of (12) and (13) expressions shows the reduction of the integrating system magnitude by 20 dB/dec.

Frequency ω_0, at which magnitude $L_I(\omega_0) = 0$, is called intersection frequency and calculated as:

$$20\lg K_I - 20\lg \omega_0 = 0; \quad \rightarrow \quad \omega_0 = K_I. \qquad (14)$$

Bode plot of integrating system is shown in Fig. 6. Intersection frequency is $\omega_0 = K_I$, and the slope of characteristic is equal to -20 dB/dec.

Semilog dependence phase angle versus frequency is calculated as:

$$\varphi_I(\omega) = \arctan\frac{\operatorname{Im} G_I(j\omega)}{\operatorname{Re} G_I(j\omega)} = -\arctan\frac{K_I/\omega}{0} = -90^\circ. \qquad (15)$$

(15) shows that output signal phase angle does not depend on frequency and is equal to –90 °. Graphically $\varphi_I(\omega)$ is shown in Fig. 7.

The practical examples of integrating circuits are shown in Fig. 3.

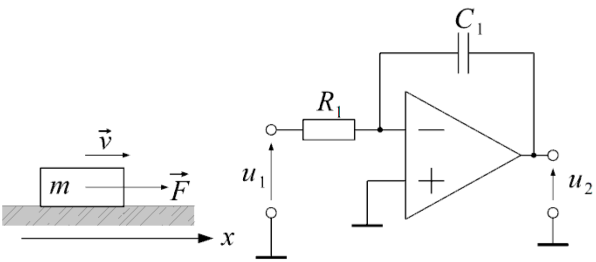

Fig. 3. Examples of integrating systems

Mechanical system, shown in Fig. 3 a, is composed by the body of mass *m*, lying on the support. If the friction force between the body and surface is not considered, the dynamics of the system is described by the second Newton‘s Law:

$$m\frac{d\vec{v}}{dt}=\vec{F}; \tag{16}$$

where: $\vec{F}$ is force, acting the body, $\vec{v}$ is speed of the body.

If the movement is one dimensional, i.e. along x axis, vector equation (16) can be replaced by projections of vectors:

$$m\frac{dv_x}{dt}=F_x; \quad \to \quad \frac{dv_x}{dt}=\frac{1}{m}F_x \tag{17}$$

Assuming system input and output signals correspondingly $u=F_x$; $y=v_x$ and $K_I=m^{-1}$, the expression (17) becomes similar to integrating system equation (8).

At analysis of circuit in Fig. 3 b it is convenient to apply virtual ground principle:

$$\frac{u_1}{R_1}=-i_2; \quad \to \quad \frac{u_1}{R_1}=-C_1\frac{du_2}{dt}; \quad \to \quad \frac{du_2}{dt}=-\frac{u_1}{C_1R_1}. \tag{18}$$

Again, if the system output and input signals correspondingly are $u=u_1$, $y=u_e$ and $K_I=-\dfrac{1}{C_1R_1}$, equation (18), describing the system, coincides with that of integrating system (8).

Many systems, having inertia, are described in time domain by the first order differential equation:

$$a_1\frac{dy}{dt}+a_0y=b_0u. \tag{19}$$

Denoting $K_A=\dfrac{b_0}{a_0}$ and $T=\dfrac{a_1}{a_0}$, the first order system is rewritten in this way:

$$T\frac{dy}{dt}+y=K_A u; \tag{20}$$

where K_A is the first order system gain and T is time constant.

The general solution of homogeneous system, in engineering called as free movement, has the form:

$$y_h = C_1 \exp\left(-\frac{t}{T}\right); \tag{21}$$

where C_1 is integrating constant, calculated from the initial conditions, t is time.

Solution of Eq. 20 is equal to the sum of solutions of general solution and solution of non-homogenic solution, which depends on input signal *u.* If the input signal is unit step and boundary conditions are assumed $y(0)=0$ and $y(\infty)=K_A$, then solution of Eq. (20) is:

$$y_A = K_A\left[1-\exp\left(-\frac{t}{T}\right)\right]. \tag{22}$$

Unit step response of the first order system is plotted in Fig. 4 according to expression (22).

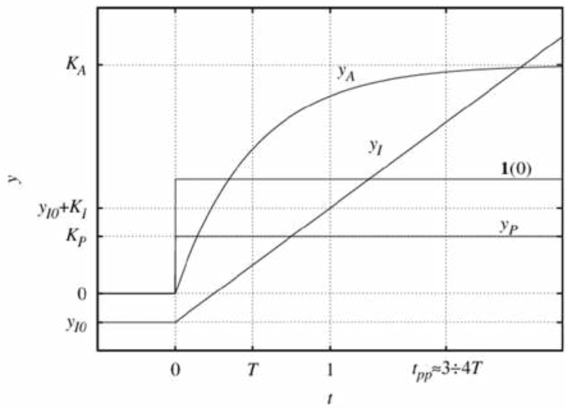

Fig. 4. Unit step responses of proportional (P), integrating (I) and first order system (A)

Figure 4 shows that unit response of the first order system exponentially approaches to steady-state value K_A. Settling time t_s is called time, after which the response differs from steady-state value by assumed ε value. If it is assumed that $\varepsilon = 5\,\%$, then $t_s \approx 3T$. For $\varepsilon = 2\,\%$, settling time $t_s \approx 4T$.

The first order system transfer function is:

$$G_A(s) = \frac{K_A}{Ts+1}. \tag{23}$$

Frequency response is calculated by replacing s with jω in transfer function (23):

$$G_A(j\omega) = \frac{K_A}{Tj\omega+1} = K_A\frac{1-Tj\omega}{1+(T\omega)^2}. \tag{24}$$

Designating $a = T\omega$ gives real and imaginary parts of frequency response in the form:

$$x \equiv \mathrm{Re}\left[G(j\omega)\right] = \frac{K_A}{1+a^2};\; y \equiv \mathrm{Im}\left[G(j\omega)\right] = -K_A\frac{a}{1+a^2}. \tag{25}$$

From Eq. (25) is evident that:

$$a^2 = \frac{K_A}{x} - 1; \tag{26}$$

then:

$$y = -K_A\frac{\sqrt{\frac{K_A}{x}-1}}{\frac{K_A}{x}} = -x\sqrt{\frac{K_A}{x}-1}. \tag{27}$$

Raising both sides of Eq. (27) in square and assuming that $\forall x \geq 0 : y \leq 0$, yields:

$$\begin{cases} y^2 = K_A x - x^2; \\ y \le 0; \end{cases} \to \begin{cases} y^2 + x^2 - 2\frac{K_A}{2}x + \left(\frac{K_A}{2}\right)^2 = \left(\frac{K_A}{2}\right)^2; \\ y \le 0. \end{cases} \quad (28)$$

Finally, Eq. (28) can be rewritten as:

$$\begin{cases} y^2 + \left(x - \frac{K_A}{2}\right)^2 = \left(\frac{K_A}{2}\right)^2 \\ y \le 0. \end{cases} \quad (29)$$

The first equation of the set (29) describes the circle with center coordinates $x_0 = (0, K_A/2)$ and radius $R = K_A/2$. The second unequality shows, that the characteristic is situated in below horizontal axis. Therefore in general way polar plot of the first order system is halfcircle, shown in Fig. 5.

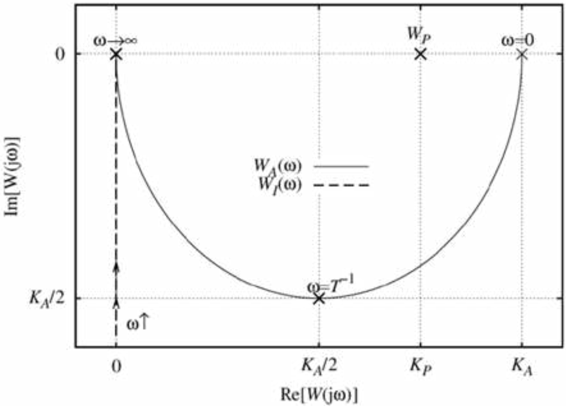

Fig. 5. Polar plots of proportional (P), integrating (I) and first order system (A)

Figure 5 shows that the first order system frequency response $G_A(\omega)$ begins and ends on the real axis. Polar plot initial coordinates are $x_0 = (K_A, 0)$ and finishing coordinates $x_\infty = (0,0)$. The plot has the third characteristic point with coordinates $x = \left(K_A/2, -K_A/2\right)$. It corresponds to frequency $\omega = 1/T$.

Magnitude of frequency response is calculated from expression (24):

$$\left|G_A(j\omega)\right| = \frac{K_A\sqrt{1+(T\omega)^2}}{1+(T\omega)^2}. \tag{30}$$

It is expressed in decibels:

$$L_A(\omega) = 20\lg\left|G_A(j\omega)\right| = 20\lg\left[\frac{K_A\sqrt{1+(T\omega)^2}}{1+(T\omega)^2}\right]. \tag{31}$$

Simple program can be made to plot the expression (31), but design of controllers uses asymptotic diagrams, made from the straight lines, asymptotically approaching to actual diagram. Asymptotic, or corner Bode, plot is analysed in the range of low and high frequency. In the low frequency range at $T\omega \ll 1$, expression (31) becomes:

$$L_A(\omega)\Big|_{\omega T<<1} \approx 20\lg K_A. \tag{32}$$

Therefore, in the low frequency range Bode diagram can be plotted by horizontal straight line.

In the high frequency range, $T\omega \gg 1$, expression (31) is rewritten as:

$$L_A(\omega)\Big|_{\omega T>>1} \approx 20\lg\left(\frac{K_A}{T\omega}\right) = 20\lg K_A - 20\lg T\omega \tag{33}$$

changing the frequency 10 times yields:

$$\begin{aligned} L_A(10\omega)\Big|_{\omega T>>1} &\approx 20\lg K - 20\lg 10T\omega = \\ &= 20\lg K - 20\lg T\omega - 20 = L_A(\omega)\Big|_{\omega T>>1} - 20. \end{aligned} \tag{34}$$

Therefore in the range of high frequency it is represented by straight line with slope of -20 dB/dec. So in the low and high frequency ranges asymptotic Bode diagram is constructed in straight lines. They intersect at a corner frequency ω_S which is calculated comparing (32) and (33) expressions:

$$20\lg K = 20\lg K - 20\lg T\omega_s; \quad \rightarrow \quad 20\lg T\omega_s = 0;$$
$$\rightarrow \quad T\omega_s = 1; \quad \rightarrow \quad \omega_s = \frac{1}{T}. \tag{35}$$

The actual $L_A(\omega)$ and the asymptotic $L_{Aa}(\omega)$ Bode diagrams, plotted correspondingly according to (31) and (32)–(33) expressions, are shown in Fig. 6.

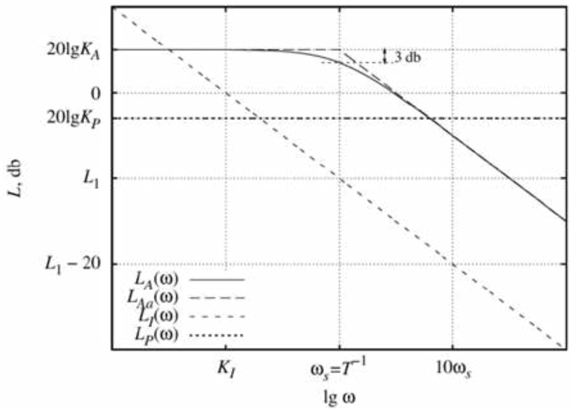

Fig. 6. Bode plots of proportional (P), integrating (I) and first order system (L_A, L_{Aa})

Fig. 6 shows that the Bode plot of the first order system is horizontal line at low frequencies with magnitude of $20\lg(K_A)$ dB. In the middle range of frequencies at frequency $\omega_s = \frac{1}{T}$, the difference between magnitudes of actual and asymptotic diagrams reaches about 3 dB. Finally in the high frequency range actual and asymptotic diagrams coincide and slope of the diagram is -20 dB/dec. The phase angle dependence versus frequency also is plotted in semilog coordinates. The phase angle is calculated from expression:

$$\varphi_A(\omega) = \arctan\frac{\operatorname{Im} G_A(j\omega)}{\operatorname{Re} G_A(j\omega)}. \qquad 6)$$

Dependence of phase angle versus frequency of the first order system is shown in Fig. 7.

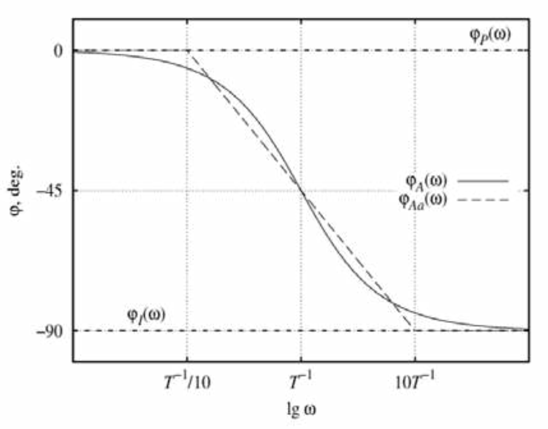

Fig. 7. Dependence of phase angle versus frequency of proportional (P), integrating (I) and first order system (L_A, L_{Aa})

Figure 7 shows that the phase angle of the first order system $\varphi(\omega)$ in the low frequency range is close to zero, i. e. is similar to that proportional system. In the middle frequency range phase it reaches – 45° at frequency $\omega = T^{-1}$. Finally, the phase angle approaches – 90° in the high frequency range. The asymptotic phase angle characteristic φ_{Aa} is also presented in Fig. 7. According to asymptotic characteristic it is possible to separate three ranges of frequency: low frequency range, while $\omega < \omega_s/10$; $\omega_s/10 < \omega < 10\omega$ is middle frequency range and $\omega > 10\omega_s$ is high frequency range.

Examples of the first order system are given in Fig. 8.

The system shown in Fig. 8a differs from the system shown in Fig. 3a just with assumption non-zero friction force $\vec{F}_T$. For sliding movement:

$$\vec{F}_T = -k_T \vec{v}; \tag{37}$$

where k_T is friction coefficient. The second Newton's Law can be written in this way:

$$m\frac{d\vec{v}}{dt} = \vec{F} + \vec{F}_T; \quad \rightarrow \quad m\frac{d\vec{v}}{dt} = \vec{F} - k_T \vec{v}. \tag{38}$$

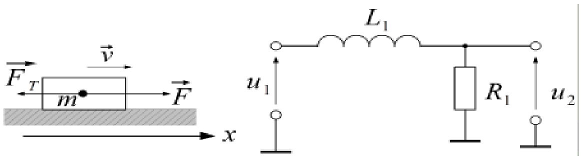

Fig. 8. Examples of the first order systems

If the vector projections in axis x are considered, then expression (38) is rewritten in scalar form as:

$$m\frac{dv}{dt} + k_T v = F; \quad \rightarrow \quad \frac{m}{k_T}\frac{dv}{dt} + v = \frac{1}{k_T}F. \tag{39}$$

If the input and output signals correspondingly are $u = F_x$, $y = v_x$, and $K_A = \dfrac{1}{k_T}$ and $T = \dfrac{m}{k_T}$, then equation (39) becomes the same as the differential equation of the first order system (Eq. (20)).

The electrical circuit shown in Fig. 8 b is described by differential equation:

$$L\frac{di}{dt}+Ri=u_1; \tag{40}$$

where i is current.

Output voltage according to Ohm‘s low is calculated as:

$$u_2=iR; \quad \to \quad i=\frac{u_2}{R}. \tag{41}$$

Substituting (41) to (40) yields:

$$\frac{L}{R}\frac{du_2}{dt}+u_2=u_1. \tag{42}$$

From Eq. (41) it is evident, that the gain of the system $K_A=1$ and time constant $T={L}/{R}$.

Ideal and real differentiating systems also depend on the first order systems. Ideal differentiating system is described as:

$$a_0y=b_1\frac{du}{dt}; \to y=\frac{b_1}{a_0}\frac{du}{dt}=K_D\frac{du}{dt}; \tag{43}$$

where K_D is gain of the system.

Step response of ideal differentiating system is Dirac impulse $\delta(t)$:

$$\delta(t)=\begin{cases}\infty, & t=0;\\ 0, & t\neq 0.\end{cases} \tag{44}$$

Dirac function is impulse function with limited signal power:

$$\int_{-\infty}^{\infty}\delta(t)dt=1. \tag{45}$$

Step response of ideal differentiating system is shown in Fig. 9, where it corresponds to Dirac impulse $\delta(t)$.

Transfer function of ideal differentiating system is:

$$a_0 Y(s) = b_1 U(s) s \quad \rightarrow \quad G_{DI}(s) \equiv \frac{Y(s)}{U(s)} = K_{DI} s. \qquad (46)$$

From expression (46) frequency response and the Bode plot can be found. Frequency response is calculated as:

$$G_{DI}(j\omega) = K_{DI} j\omega. \qquad (47)$$

Polar plot of expression (47) coincides with imaginary axis and begins at the origin at $\omega = 0$ as well as approaches to infinity at $\omega = \infty$. Polar plot of $G_{D1}(j\omega)$ is shown in Fig. 10. Magnitude of Bode diagram is calculated as:

$$L_{DI}(\omega) = 20\lg(K_{DI}\omega) = 20\lg(K_{DI}) + 20\lg(\omega). \qquad (48)$$

Expression (48) is straight line with –20 dB/dec slope. Frequency, at which amplitude of ideal differentiating system is equal to zero, is obtained by setting to zero the left hand side of expression (48):

$$20\lg(K_{DI}) = -20\lg(\omega_0), \quad \rightarrow \quad \omega_0 = K_{DI}^{-1}. \qquad (49)$$

$L_{DI}(\omega)$ is shown in Fig. 11.

Phase dependence versus frequency is calculated as:

$$\varphi_{DI}(\omega) = \arctan \frac{\operatorname{Im} G_{DI}(j\omega)}{\operatorname{Re} G_{DI}(j\omega)} = 90^{\circ}. \qquad (50)$$

Thus $\varphi_{DI}(\omega)$ is horizontal straight with ordinate, equal to 90º. It is shown in Fig. 12. Ideal differentiating system has no prototypes in real techniques. Instead of that real differentiating system comprised from ideal differentiating system and first order system is used. It is described as:

$$a_1 \frac{dy}{dt} + a_0 y = b_1 \frac{du}{dt}. \tag{51}$$

Using notations $K_{DR} = b_1 / a_0$ and $T = a_1 / a_0$, differential equation can be rewritten in this way:

$$T \frac{dy}{dt} + y = K_{DR} \frac{du}{dt}; \tag{52}$$

where: K_{DR} is gain of real differentiating system and T is time constant.

Left sides of Eqs. (51) and (20) are the same, thus the solution of homogenous equation will have the same form as (21). Analytical expression of general solution at $u = 1(t)$ is:

$$y = \frac{K_{DR}}{T} \exp\left(-\frac{t}{T}\right). \tag{53}$$

Step response of real differentiating system is shown in Fig. 9. As the solution of homogenous system is the same as that of the first order system, thus step response is characterized by the same settling time $t_{pp} = 3 \div 4T$.

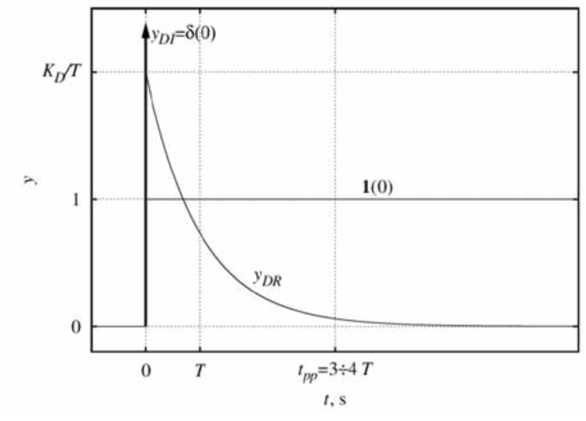

Fig. 9. Step responses of ideal (DI) and real (DR) differentating systems

According to (52) the transfer function of real differentiating system is:

$$G_{DR} = \frac{K_{DR} s}{Ts + 1}. \tag{54}$$

Transfer function of real differentiating system in frequency domain is expressed as:

$$G_{DR}\left(j\omega\right) = \frac{K_{DR} j\omega}{Tj\omega + 1} = K_{DR} \frac{T\omega^2 + j\omega}{1 + \left(T\omega\right)^2}. \tag{55}$$

Real and imaginary part of expression (55) has the form:

$$x \equiv \mathrm{Re}\left[G\left(j\omega\right)\right] = \frac{K_{DR} T\omega^2}{1 + \left(T\omega\right)^2};\ y \equiv \mathrm{Im}\left[G\left(j\omega\right)\right] = \frac{K_{DR}\omega}{1 + \left(T\omega\right)^2}. \tag{56}$$

he real part of $G_{DR}\left(j\omega\right)$ can be presented in this way:

$$\omega^2 = \frac{x}{T\left(K_{DR} - Tx\right)}. \tag{57}$$

Substituting ω to expression (56) yields:

$$y = \frac{K_{DR}\sqrt{\dfrac{x}{T\left(K_D - Tx\right)}}}{1 + T^2 \dfrac{x}{T\left(K_D - Tx\right)}} = \left(K_{DR} - Tx\right)\sqrt{\frac{x}{T\left(K_{DR} - Tx\right)}}. \tag{58}$$

Dividing both sides of Eq. (58) by $\left(K_D - T_x\right)$ and taking square of those, also considering $\forall\omega \geq 0 : y \geq 0$ gives:

$$\begin{cases} \left(\dfrac{y}{K_{DR} - Tx}\right)^2 = \dfrac{x}{T\left(K_{DR} - Tx\right)}; \\ y \geq 0; \end{cases} \rightarrow \begin{cases} y^2 + x^2 - x\dfrac{K_{DR}}{T} = 0; \\ y \geq 0. \end{cases} \tag{59}$$

Parametric equation (59) is rewritten in this way:

$$\begin{cases} y^2 + x^2 - 2\dfrac{K_{DR}}{2T}x + \left(\dfrac{K_{DR}}{2T}\right)^2 = \left(\dfrac{K_{DR}}{2T}\right)^2; \\ y \geq 0; \end{cases}$$
$$\rightarrow \begin{cases} y^2 + \left(x - \dfrac{K_{DR}}{2T}\right)^2 = \left(\dfrac{K_{DR}}{2T}\right)^2. \\ y \geq 0. \end{cases} \tag{60}$$

The first equation of the set (60) describes circle with centre coordinates $x_0 = \left[0, K_{DR}/2T\right]$ and radius $R = K_{DR}/(2T)$. The second equation states the polar plot being over the real axis as half cycle. Polar plot of real differentiating system is shown in Fig. 10.

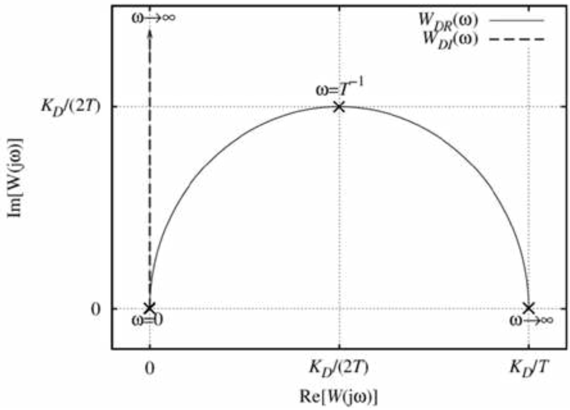

Fig. 10. Polar plot of ideal (DI) and real (DR) differentiating systems

Magnitude of real differentiating system is calculated from expression (55) as:

$$\left|G_{DR}(j\omega)\right| = \frac{K_{DR}\omega}{1+(T\omega)^2}\left|T\omega + j\right| = \frac{K_{DR}\omega\sqrt{(T\omega)^2+1}}{1+(T\omega)^2}. \tag{61}$$

Bode plot uses magnitudes, expressed by decibels, thus:

$$L_{DR}(\omega) = 20\lg\left|G_{DR}(j\omega)\right| = 20\lg\frac{K_{DR}\omega\sqrt{(T\omega)^2+1}}{1+(T\omega)^2}. \tag{62}$$

Expression (62) can be directly used to plot Bode diagram by computer, but in many cases the asymptotic Bode plots are constructed. For this at first two frequency ranges are analysed: low frequency range at $T\omega \ll 1$ and high frequency range at $T\omega \gg 1$. At low frequency range expression (62) becomes:

$$L_{DR}(\omega)\Big|_{T\omega<<1} = 20\lg(K_{DR}\omega). \tag{63}$$

Expression (63) is the same as (47) for ideal differentiating system. It describes the straight line with slope +20 dB/dec. At high frequency zone expression (62) is rewritten in this way:

$$L_{DR}(\omega)\Big|_{T\omega>>1} = 20\lg\frac{K_{DR}}{T}. \tag{64}$$

Expression (64) corresponds to horizontal line with ordinate, equal to K_{DR}/T decibels. Finally, the corner frequency ω_s can be found, where both straights intersect. Set equal expression (63) to (64) and get:

$$20\lg(K_{DR}\omega_s) = 20\lg\frac{K_{DR}}{T}; \quad \rightarrow \quad \omega_s = \frac{1}{T}. \tag{65}$$

Thus, the asymptotic Bode diagram of real differentiating system is composed of two segments of straight lines.

At $\omega < \omega_s$, it is straight line with positive slope +20 dB/dec, and at $\omega > \omega_s$ it is horizontal line with ordinate of $20\lg(K_{DR}/T)$ decibels. Asymptotic Bode plot of real differentiating circuit $L_{DRa}(\omega)$ is shown in Fig. 11.

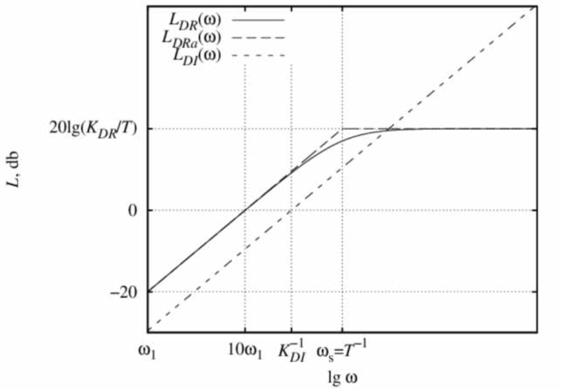

Fig. 11. Bode plots of ideal (DI) and real (DR) differentiating systems

Dependence of output signal phase angle versus frequency is calculated as:

$$\varphi_{DR}(\omega) = \arctan\frac{\operatorname{Im} G_{DR}(j\omega)}{\operatorname{Re} G_{DR}(j\omega)}. \tag{66}$$

Graph of $\varphi_{DR}(\omega)$ is shown in Fig. 12.

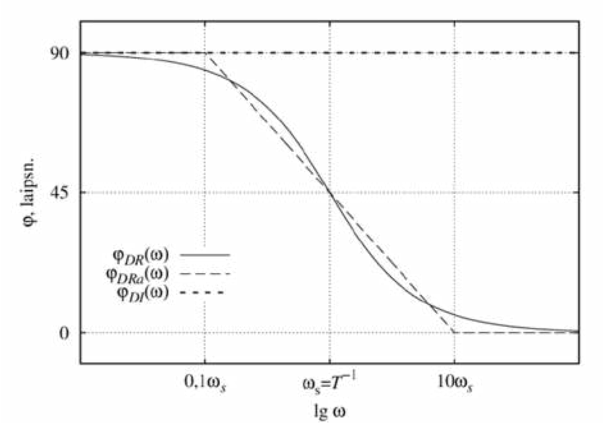

Fig. 12. Phase characteristic of the frequency response of ideal (DI) and real (RD) differentiating system in semi-log scale

In the same Fig. 12 the asymptotic phase angle curve $\varphi_{DRa}(\omega)$ is presented. It approximates real phase angle characteristic and is convenient for design of the desired controller characteristics.

The simple circuit, realizing real differentiating system is shown in Fig. 13.

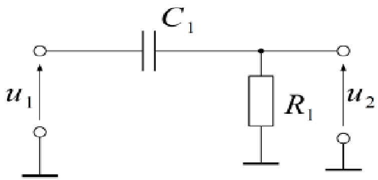

Fig. 13. Real differentiating circuit

Differential equation describing the circuit in Fig. 13 is:

$$\frac{1}{C_1}\int i dt + iR_1 = u_1. \tag{67}$$

System output signal is:

$$u_2 = iR_1. \tag{68}$$

Substituting (68) to (67) yields:

$$\frac{1}{C_1 R_1}\int u_2 dt + u_2 = u_1; \quad \rightarrow \quad C_1 R_1 \frac{du_2}{dt} + u_2 = C_1 R_1 \frac{du_1}{dt}. \tag{69}$$

Denoting $T = C_1 R_1$; $K_{DR} = C_1 R_1$; $y = u_2$ and $u = u_1$ gives the differential equation (52) of real differentiating system.

Content of the report

1. Elaboration of mathematical model of the system indicated by the teacher.
2. Step responses and asymptotic Bode plots.
3. Actual frequency response magnitude and phase angle characteristics calculated and plotted by Matlab.

Control questions

1. What are the first order systems?
2. Why proportional system is not ideal system?
3. What slope has the magnitude straight of integrating system in Bode plot?
4. What is asymptotic Bode diagram of the first order system? How exactly does it approximate the actual?
5. What systems comprise the real differentiating system?
6. How is the Bode plot of real differentiating system constructed?
7. How do the frequency response magnitude and phase characteristics of ideal differentiating system look like?

THE SECOND ORDER SYSTEMS

Laboratory work No. 4

Objectives: to get acquainted with the second order system and to learn to analyze its characteristics in time domain and frequency domain.

Tasks of the work:

1. Elaborate mathematical model of the system, indicated by the teacher.
2. Construct simulation model of the system.
3. Use MATLAB to plot step response, frequency response and the Bode plot.

Theoretical part

In general case the second order system is described by differential equation:

$$a_2 \frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_0 y = b_2 \frac{d^2 u}{dt^2} + b_1 \frac{du}{dt} + b_0 u \qquad (1)$$

where: u and y are system input and output signals, a_0, a_1, a_2, b_0 and b_1 are constants. For practical cases it is convenient to assume $b_1 = b_0 = 0$ and Eq. 1 becomes:

$$a_2 \frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_0 y = b_0 u \qquad (2)$$

Demoting $b_0 / a_0 = K$, $a_1 / a_0 = 2\rho / \omega_0$ ir $a_2 / a_0 = 1 / (\omega_0)^2$, Eq. 2 is rewritten in this way:

$$\frac{1}{\omega_0^2} \frac{d^2 y}{dt^2} + \frac{2\rho}{\omega_0} \frac{dy}{dt} + y = Ku. \qquad (3)$$

Notations, chosen in this way, take the physical meaning: ω_0 is natural frequency of the system, ρ is damping ratio and K is gain.

Solution of Eq. (3) is equal to the sum of solutions of homogenous equation (if the right hand side of Eq. 3 is equal to zero), the so called free movement of the system and solution of non-homogenous system (the so called forced movement). General solution of homogeneous system depends on roots of characteristic equation. It is obtained replacing $\frac{d}{dt} \to \lambda$ in Eq. 3:

$$\frac{1}{\omega_0^2}\lambda^2 + \frac{2\rho}{\omega_0}\lambda + 1 = 0. \tag{4}$$

Characteristic equation (4) is the same polynomial of transfer function denominator therefore its roots in automatic control theory are called system poles:

$$p_{1,2} = \frac{-\frac{2\rho}{\omega_0} \pm \sqrt{\frac{4\rho^2}{\omega_0^2} - \frac{4}{\omega_0^2}}}{\frac{2}{\omega_0^2}} = -\rho\omega_0 \pm \omega_0\sqrt{\rho^2 - 1}. \tag{5}$$

If $\rho^2 - 1 > 0; \to |\rho| > 1$ and $p_{1,2} \in R$, the general solution of homogeneous equation has the form:

$$y_h = C_1 \exp\left(p_1 t\right) + C_2 \exp\left(p_2 t\right) \tag{6}$$

Mathematical analysis also considers cases when $\rho < -1$, but then $-\rho\omega_0 = |\rho|\omega_0$ and $p_1 = |\rho|\omega_0 + \omega_0\sqrt{\rho^2 - 1} > 0$, therefore the first term of Eq. 6 increases to infinity with rising of t. The automatic control theory discusses just the systems with $\rho \geq 0$.

The particular solution depends on system input u. If $u = 1(0)$, then solution of non-homogeneous system has the form:

$$y_{nh} = A \tag{7}$$

Substituting (7) and its derivatives, that are equal to zero, into Eq. 3 and assuming $u = 1(0)$ yields:

$$y_{nh} \equiv A \equiv K \tag{8}$$

Summing Eq. (6) and Eq. (8), the general solution of differential equation (3) at unit step input is:

$$y = y_h + y_{nh} = C_1 \exp(p_1 t) + C_2 \exp(p_2 t) + K. \tag{9}$$

Constants C_1 and C_2 can be calculated from initial conditions, i. e. $y = 0$ and $y' = 0$ at $t = 0$, then the set of equations is obtained:

$$\begin{cases} C_1 + C_2 + K = 0; \\ C_1 p_1 + C_2 p_2 = 0. \end{cases} \tag{10}$$

Solution of Eq. 10 is obtained as:

$$\begin{cases} C_1 = -\dfrac{K p_2}{p_2 - p_1}; \\ C_2 = \dfrac{K p_1}{p_2 - p_1}. \end{cases} \tag{11}$$

Finally, if $\delta > 1$ and the system has zero initial conditions, step response is described as:

$$y = -\frac{K p_2 \exp(p_1 t)}{p_2 - p_1} + \frac{K p_1 \exp(p_2 t)}{p_2 - p_1} + K; \tag{12}$$

where: $p_{1,2}$ are calculated from Eq. (5).

While $p_{1,2} < 0$, the solution is composed from the sum of two terms, decaying by exponential low and gain K. At zero initial conditions there is no any overshoot in this case, therefore y has deadbeat shape and with time approaches to gain value K.

The set of second order system responses with $\omega_0 = 5$ and $K = 2{,}5$ at $\rho = [0{,}5;\ 0{,}75;\ 1;\ 1{,}25;\ 1{,}5]$. Two graphs are presented for $\rho = 1{,}25$ and $\rho = 1{,}5$. Increase damping ratio increases settling time. The shape of step response remains the same. If $\rho^2 - 1 = 0; \rightarrow \rho = 0$, then $p \equiv p_1 = p_2$ and $p \in R$. Then the general solution of homogenous system is described as:

$$y_h = C_1 \exp(pt) + C_2 \exp(pt) \tag{13}$$

Particular solution of non-homogeneous equation depends only on the right hand side of Eq. 3, and is equal to the same expression (8), therefore the general solution of Eq. (3) at δ = 1 is:

$$y = C_1 \exp(pt) + C_2 \exp(pt) + K \tag{14}$$

Constants C_1 and C_2 can be calculated from initial conditions. If the zero initial conditions are assumed, i. e. $y = 0$ and $y' = 0$ at $t = 0$, the obtained set has a form:

$$\begin{cases} C_1 + K = 0; \\ C_1 p + C_2 = 0; \end{cases} \rightarrow \begin{cases} C_1 = -K; \\ C_2 = Kp. \end{cases} \tag{15}$$

Finally, if δ = 0, step response at initial zero conditions is expressed as:

$$y = -K \exp(pt)(1 - pt) + K. \tag{16}$$

There is no overshoot in this case also and step response is deadbeat.

Step response of the second order system at $\omega_0 = 5$; $K = 2{,}5$ and $\rho = 1$ is shown in Fig. 1. The settling time is shorter, than for system with $\rho > 1$.

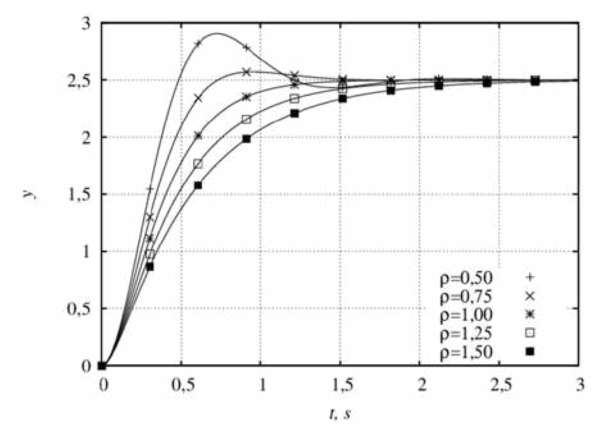

Fig. 1. Set of the second order system step responses at different ρ values

The next case to be considered is matching condition $\rho^2 - 1 < 0; \rightarrow \rho < 1$ and $p_{1,2} \in C$ and roots are equal:

$$p_{1,2} = -\rho\omega_0 \pm \omega_0 j\sqrt{1-\rho^2}. \tag{17}$$

In this case the general solution of homogeneous equation has the form:

$$\begin{aligned} y_h &= C_1 \exp(-\rho\omega_0 t)\sin\left(\omega_0\sqrt{1-\rho^2}\ t\right)+ \\ &+C_2 \exp(-\rho\omega_0 t)\cos\left(\omega_0\sqrt{1-\rho^2}\ t\right). \end{aligned} \tag{18}$$

If the input is unit step, then the particular solution of non-homogenous system again will have the form (7). While $y_h \rightarrow 0$ and $t \rightarrow \infty$, then expression (8) is valid also and the general solution is:

$$\begin{aligned} y &= C_1 \exp(-\rho\omega_0 t)\sin\left(\omega_0\sqrt{1-\rho^2}\ t\right)+ \\ &+C_2 \exp(-\rho\omega_0 t)\cos\left(\omega_0\sqrt{1-\rho^2}\ t\right)+K. \end{aligned} \tag{19}$$

Constants C_1 and C_2 are calculated from initial conditions. If zero initial conditions are assumed , i. e. $y = 0$ and $y' = 0$ at $t = 0$, then the set of equations for calculation constants looks like this:

$$\begin{cases} C_2 + K = 0; \\ C_1 \operatorname{Im} p_{1,2} + C_2 \operatorname{Re} p_{1,2} = 0; \end{cases} \quad \rightarrow \quad \begin{cases} C_2 + K = 0; \\ C_1 \omega_0 \sqrt{1-\rho^2} - C_2 \omega_0 \rho = 0. \end{cases} \tag{20}$$

Solutions of Eq. (18) yields:

$$\begin{cases} C_1 = -\dfrac{K\rho}{\sqrt{1-\rho^2}}. \\ C_2 = -K. \end{cases} \tag{21}$$

Thus, finally, solution of differential equation (3) at $\rho < 1$ is expresses as:

$$\begin{aligned} y = & -\frac{K\rho}{\sqrt{1-\rho^2}} \exp(-\rho\omega_0 t) \sin\left(\omega_0 \sqrt{1-\rho^2}\, t\right) - \\ & -K \exp(-\rho\omega_0 t) \cos\left(\omega_0 \sqrt{1-\rho^2}\, t\right) + K. \end{aligned} \tag{22}$$

Expession (22) causes difficulties in its application therefore it can be presented in a simpler form. The first two terms of equation, describing the free response is solution of homogenic equation at zero initial conditions and can be rewritten in this way:

$$\begin{aligned} y_h = & -\frac{K\rho}{\sqrt{1-\rho^2}} \exp(-\rho\omega_0 t) \sin\left(\omega_0 \sqrt{1-\rho^2}\, t\right) - \\ & -K \exp(-\rho\omega_0 t) \cos\left(\omega_0 \sqrt{1-\rho^2}\, t\right) = \\ & = -K \exp(-\rho\omega_0 t) \left[\begin{array}{l} \dfrac{\rho}{\sqrt{1-\rho^2}} \sin\left(\omega_0 \sqrt{1-\rho^2}\, t\right) + \\ \cos\left(\omega_0 \sqrt{1-\rho^2}\, t\right) \end{array} \right]. \end{aligned} \tag{23}$$

Demote a = tanφ and apply formula:

$$\cos\omega t + a\sin\omega t = \cos\omega t - \frac{\sin\varphi}{\cos\varphi}\sin\omega t = \frac{\cos(\omega t+\varphi)}{\cos\varphi}. \quad (24)$$

If $\omega = \omega_0\sqrt{1-\rho^2}$, $a = \rho\big/\sqrt{1-\rho^2}$ and $\varphi = -\arctan a$, then expression (24) can be applied for portion of Eq. (23):

$$\frac{\rho}{\sqrt{1-\rho^2}}\sin\left(\omega_0\sqrt{1-\rho^2}\,t\right) + \cos\left(\omega_0\sqrt{1-\rho^2}\,t\right) = \\ = \frac{\cos\left(\omega_0\sqrt{1-\rho^2}\,t+\varphi\right)}{\cos\varphi}. \quad (25)$$

In order to express cos φ by parameters of Eq. (3), the set equations must be solved:

$$\begin{cases}\sin^2\varphi+\cos^2\varphi=1;\\ -\tan\varphi=\dfrac{\rho}{\sqrt{1-\rho^2}};\end{cases} \to \begin{cases}\sin^2\varphi+\cos^2\varphi=1;\\ \sin\varphi=-\dfrac{\rho}{\sqrt{1-\rho^2}}\cos\varphi;\end{cases} \\ \to \cos\varphi=\sqrt{1-\rho^2}. \quad (26)$$

Application of express (25) and (26) for (23) gives:

$$y_h = -\frac{K}{\sqrt{1-\rho^2}}\exp(-\rho\omega_0 t)\cos\left(\omega_0\sqrt{1-\rho^2}\,t+\varphi\right); \quad (28)$$

where $\varphi = -\arctan\dfrac{\rho}{\sqrt{1-\rho^2}}$.

Expressions (27) and (28) are more suitable for analysis. They show exponentially decaying oscillations. Frequency of oscillations:

$$\omega_s = \omega_0\sqrt{1-\rho^2} = \operatorname{Im} p_{1,2} \tag{29}$$

is called frequency of selfoscillations.

It is evident from Eq. (28), that at ρ = 0, the system's output is described in this way:

$$y = -K\cos\left(\omega_0 t\right) + K. \tag{30}$$

In general magnitude of oscillation is found from expression:

$$Y = -\frac{K}{\sqrt{1-\rho^2}}\exp\left(-\omega_0\rho t\right). \tag{31}$$

Expression (31) shows that decaying speed is proportional to ρ, therefore ρ is called damping ratio, and if ρ = const, amplitude also remains constant.

Two step responses, shown in Fig. 1 correspond to systems with ρ < 1. The overshoot appears and its magnitude and frequency increases with reducing ρ.

Generalizing analysis of the second order system in time domain can be concluded:

at $\rho \geq 1$ the system has deadbed step response without overshoot and oscillations;

at $0 \leq \rho < 1$ the step response is oscillating with oscillations frequency $\omega_0\sqrt{1-\rho^2}$; the magnitude decays, if ρ > 0 or remains constant, if ρ = 0.

In most cases for synthesis of controller (oscillating) systems with $0 \leq \rho < 1$ are considered.

In Eq. 3 replacing $\frac{d}{dt} \to s$, the transfer function of the system gets the form:

$$G(s) \equiv \frac{Y(s)}{U(s)} = \frac{K}{\frac{s^2}{\omega_0^2} + \frac{2\rho}{\omega_0} s + 1}. \tag{32}$$

Substituting $s = j\omega$ yields the frequency response:

$$G(j\omega) = \frac{K}{-\left(\frac{\omega}{\omega_0}\right)^2 + 1 + \frac{2\rho}{\omega_0} j}. \tag{33}$$

The polar plot of the first order system has two characteristic points. From Eq. (33) at $\omega = 0$ the initial point is calculated as:

$$G(j0) = K; \tag{34}$$

and the final point at $t \to \infty$:

$$G(j\infty) = \lim_{\omega \to \infty} \frac{K}{-\left(\frac{\omega}{\omega_0}\right)^2 + 1 + \frac{2\rho}{\omega_0} j} = 0. \tag{35}$$

Denoting real and imaginary part of Eq. (3) as:

$$\begin{aligned} \alpha(\omega) &\equiv \Re\left[-(\omega/\omega_0)^2 + 2\rho\, j/\omega_0 + 1\right] = 1 - (\omega/\omega_0)^2; \\ \beta(\omega) &\equiv \Im\left[-(\omega/\omega_0)^2 + 2\rho\, j/\omega_0 + 1\right] = 2\rho / \omega_0; \end{aligned} \tag{36}$$

Eq. (33) can be rearranged to this form:

$$G(j\omega) = \frac{K}{\alpha(\omega) + \beta(\omega) j} = \frac{K\left[\alpha(\omega) - \beta(\omega) j\right]}{\alpha^2(\omega) + \beta^2(\omega)}. \tag{37}$$

The real part of complex transfer function is:

$$\Re G(j\omega) = \frac{K\alpha(\omega)}{\alpha^2(\omega) + \beta^2(\omega)}. \tag{38}$$

Numerator of the real parts in Eq. (38) will be equal to zero only if $K\alpha(\omega) = 0$ and then:

$$K\alpha(\omega) = 0; \quad \to \quad \alpha(\omega) = 0; \quad \to \quad 1 - \left(\frac{\omega}{\omega_0}\right)^2 = 0; \\ \to \quad \omega = \omega_0. \tag{39}$$

Imaginary part of Eq. (38) for the frequency ω_0 becomes:

$$\Im G(j\omega_0) = -\frac{K\beta(\omega_0)}{\alpha(\omega_0) + \beta(\omega_0)} = -\frac{2\rho K}{4\rho^2} = -\frac{K}{2\rho}. \tag{40}$$

Thus, the polar plot of the second order system intersects imaginary axis at the point x = $(0, -K/2\rho)$ where the frequency is equal to ω_0. Polar plot of the system with $\omega_0 = 5$, $K = 2{,}5$ and $\rho = [0{,}5;\ 0{,}75;\ 1;\ 1{,}25;\ 1{,}5]$ is presented in Fig. 2.

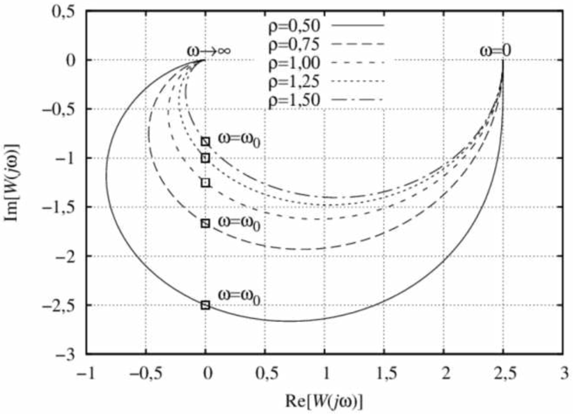

Fig. 2. Polar plot of the second order system with different ρ values

It is evident, that all characteristics begin at point $x(0) = (K; 0) = (2,5; 0)$ and end at point $x(\infty) = (0; 0)$. In this figure frequencies ω_0 are marked by small squares.

Expression (37) is convenient for analysis of magnitude and phase Bode plots. Frequency response magnitude characteristic is:

$$|G(j\omega)| = \sqrt{\left[\frac{K\alpha(\omega)}{\alpha^2(\omega)+\beta^2(\omega)}\right]^2 + \left[\frac{K\beta(\omega)}{\alpha^2(\omega)+\beta^2(\omega)}\right]^2} = \\ = \frac{K}{\sqrt{\alpha^2(\omega)+\beta^2(\omega)}}. \tag{41}$$

Magnitude (41) is expressed in decibels:

$$L(\omega) \equiv 20\lg|G(j\omega)| = 20\lg\left[\frac{K}{\sqrt{\alpha^2(\omega)+\beta^2(\omega)}}\right]. \tag{42}$$

Expression (42) describes the actual characteristic, but it can be plotted just using computer. Asymptotic characteristics have more attractive form and advantages at design of regulators.

Asymptotic magnitude characteristic is constructed of straight lines, intersecting at corner frequency. In low frequency range, when $\omega \ll \omega_0$, then $\alpha(\omega) \approx 1$, $\beta(\omega) \approx 0$ and expression (42) gives magnitude:

$$|G(j\omega)|\Big|_{\omega T<<1} = K; \quad \rightarrow \quad L(\omega)\Big|_{\omega T<<1} \equiv 20\lg\left|G(j\omega)\right|\Big|_{\omega T<<1} \\ = 20\lg K. \tag{43}$$

Thus, in the low frequency range the magnitude is constant and represented by horizontal line parallel to frequency axis with ordinate K dB.

In the high frequency range, at $\omega >> \omega_0$, then $\alpha(\omega) \approx -(\omega/\omega_0)^2$, therefore magnitude is:

$$\left|G(j\omega)\right|\Big|_{\omega T>>1} = \frac{K}{\sqrt{\left(\frac{\omega}{\omega_0}\right)^4 + 4\rho^2\left(\frac{\omega}{\omega_0}\right)^2}} = \frac{K\omega_0}{\omega\sqrt{\left(\frac{\omega}{\omega_0}\right)^2 + 4\rho^2}}. \quad (44)$$

Damping ratio ρ is not limited, but usually it reaches order of units, therefore, if $\omega >> \omega_0$, then inequality $(\omega/\omega_0)^2 >> 4\rho^2$ and (44) is rewritten in this form:

$$\left|G(j\omega)\right|\Big|_{\omega T>>1} = \frac{K\omega_0}{\omega\sqrt{\left(\frac{\omega}{\omega_0}\right)^2}} = K\left(\frac{\omega_0}{\omega}\right)^2. \quad (45)$$

Magnitude characteristic (45) is expressed in decibels:

$$L(\omega)\Big|_{\omega T>>1} = 20\lg\left[K\left(\frac{\omega_0}{\omega}\right)^2\right] = 20\lg K - 40\lg\frac{\omega}{\omega_0}. \quad (46)$$

Using semi-log scale for frequency, expression (46) is represented by straight line. If the frequency is increased 10 times, then:

$$\begin{aligned} L(10\omega)\Big|_{\omega T>>1} &= 20\lg K - 40\lg\frac{10\omega}{\omega_0} = 20\lg K - 40\lg\frac{\omega}{\omega_0} - 40 = \\ &= L(\omega)\Big|_{\omega T>>1} - 40. \end{aligned} \quad (47)$$

The magnitude smaller by 40 decibels is obtained. Thus in high frequency range the second order system Bode plot is expressed by straigh line with slope of –40 dB/dec. The Bode plot in the low and high frequencies is constructed of two lines correspondingly according to (43) and (46) expressions intersecting at corner frequency ω_s:

$$K = K\left(\frac{\omega_0}{\omega_s}\right)^2; \quad \to \quad \omega_s = \omega_0. \tag{48}$$

Corner plot ω_s is equal to system resonant frequency ω_0.

The Bode plot of the second order system with $\omega_0 = 5$, $K = 2{,}5$ and $\rho = [0{,}25; 0{,}5; 0{,}75; 1{,}0; 1{,}25; 1{,}5]$ are presented in Fig. 3.

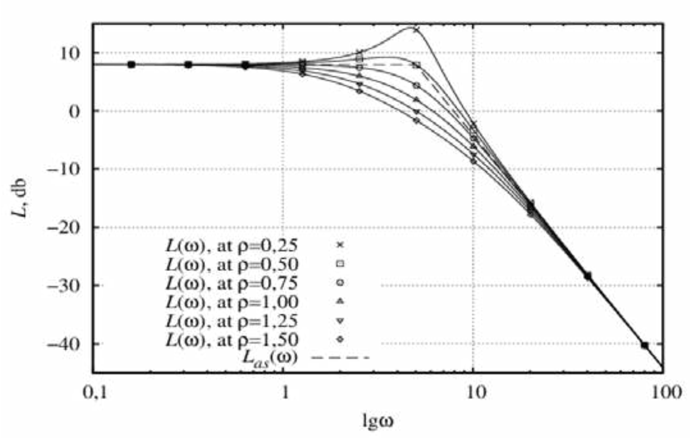

Fig. 3. Bode plot of the second order system with different ρ values

Fig. 3 shows that approaching the damping ratio to zero increases magnitude and frequency at which the maximum is reached and approaches to resonant frequency $\omega \to \omega_s = \omega_0$. Dependence of phase angle versus frequency is calculated as:

$$\varphi(\omega) = \arctan\frac{\Im G(j\omega)}{\Re G(j\omega)} = \arctan\frac{\dfrac{-K\beta(\omega)}{\alpha^2(\omega)+\beta^2(\omega)}}{\dfrac{K\alpha(\omega)}{\alpha^2(\omega)+\beta^2(\omega)}} = \tag{49}$$

$$\arctan\frac{-\beta(\omega)}{\alpha(\omega)}.$$

Substituting (36) to expression (49) yields:

$$\varphi(\omega)=\arctan\frac{-2\rho\dfrac{\omega}{\omega_0}}{1-\left(\dfrac{\omega}{\omega_0}\right)^2}. \tag{50}$$

In low frequency range, i. e. at $\omega << \omega_0$, and damping ratio is of units order, the phase angle:

$$\varphi(\omega)\Big|_{\omega T<<1}=\arctan\left(-2\rho\frac{\omega}{\omega_0}\right)\approx\arctan(-0)\approx-0^\circ; \tag{51}$$

thus, the curve is a bit below frequency axis. At high frequency range, at $\omega >> \omega_0$, expression (49) is rewritten in this way:

$$\varphi(\omega)\Big|_{\omega T>>1}=\arctan\frac{-2\rho\omega_0}{-\omega}\approx-180^\circ. \tag{52}$$

Thus, the phase angle characteristic with increasing of frequency approaches to –180º. The other characteristic point is

$\omega = \omega_{0.}$ At this point α(ω) has interrupt, therefore limits analysis should be done of the left hand side and the right hand side of this point:

$$\varphi(\omega)\Big|_{\omega\to\omega_{0-}}=\lim_{\alpha(\omega)\to\infty}\arctan\frac{-2\rho}{\alpha(\omega)}=-90^\circ; \tag{53}$$

and

$$\varphi(\omega)\Big|_{\omega\to\omega_{0+}}=\lim_{\alpha(\omega)\to-\infty}\arctan\frac{-2\rho}{\alpha(\omega)}=-90^\circ. \tag{54}$$

Thus, if the frequency ω acquires value ω_0, the phase angle at this frequency reaches –90º. Nevertheless, this result could be obtained from analysis of phase angle versus frequency, i. e. expression (39).

Phase angle dependence versus frequency in semilog scale for the second order system with parameters $\omega_0 = 5$, $K = 2{,}5$ and $\rho = [0{,}25; 0{,}5; 0{,}75; 1{,}0; 1{,}25; 1{,}5]$ are presented in Fig. 4.

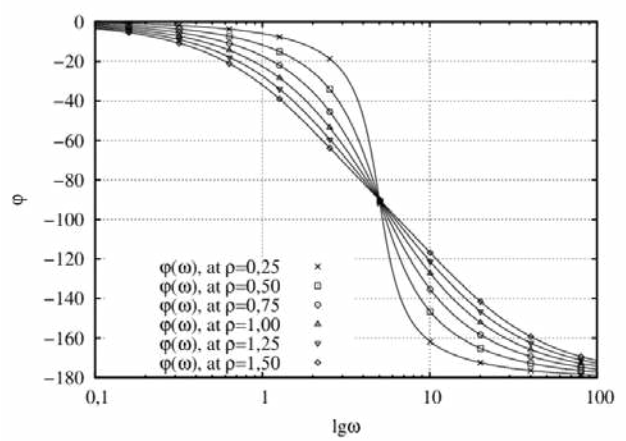

Fig. 4. Phase angle Bode plot of the second order system with different ρ values

Mechanical and electrical circuits, shown in Fig. 5 a and 5 b are typical examples of mechanical and electrical second order systems.

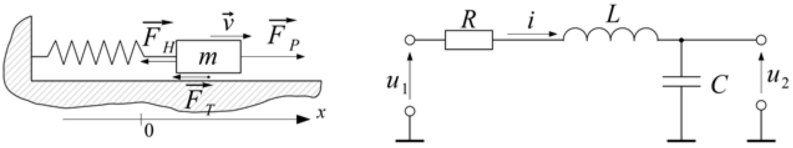

Fig. 5. The second order systems

Elaboration of mathematical model of mechanical system shown in Fig. 5 a is based on Newton‘s laws:

$$m\frac{d\vec{v}}{dt} = \sum \vec{F}; \quad \rightarrow \quad m\frac{d\vec{v}}{dt} = \vec{F}_P + \vec{F}_H + \vec{F}_T; \qquad (55)$$

where $\vec{F}_P, \vec{F}_H, \vec{F}_T$ are external impact, elasticity and friction forces correspondingly. Expression (55) can be rewritten as vector projections in x axis:

$$m\frac{dv}{dt} = F_P - F_H - F_T. \quad (56)$$

Force magnitude of spring elasticity is described by Robert Hooke law:

$$F_H = k_H x; \quad (57)$$

where k_H is spring elasticity; x – displacement of body.

Magnitude of sliding friction force is:

$$F_T = k_T v. \quad (58)$$

Substituting (57) and (58) to (56) and expressing speed and acceleration by derivatives of displacement, yields:

$$m\frac{d^2x}{dt^2} + k_T\frac{dx}{dt} + k_H x = F_P. \quad (59)$$

Damping ratio has order of some units.

Denoting $\omega_0^{-2} = m/k_H ;\rightarrow \omega_0 = \sqrt{k_H/m}$ and $2\rho\omega_0^{-1} = k_T/k_H ;\rightarrow \rho = k_T\Big/\left(2\sqrt{mk_H}\right)$, expression (59) is the conventional form (3) of the second order system.

Mathematical model of electrical system, shown in Fig. 5 b is based on the Kirchhoff's law (KVL):

$$\begin{cases} L\dfrac{di}{dt} + u_C + Ri = u; \\ u_C = \dfrac{1}{C}\int idt. \end{cases} \rightarrow \quad LC\frac{d^2u_C}{dt^2} + CR\frac{du_C}{dt} + u_C = u. \quad (60)$$

Denoting $\omega_0^{-2} = LC; \rightarrow \omega_0 = 1/\sqrt{LC}$

and $2\rho\omega_0^{-1} = CR; \rightarrow \delta = R\sqrt{C}\Big/\left(2\sqrt{L}\right)$ expression (60) can be rewritten in conventional form of the second order system.

Problems of the second order system analysis

Practically useful example of the second order system is the second order filter, shown in Fig. 6.

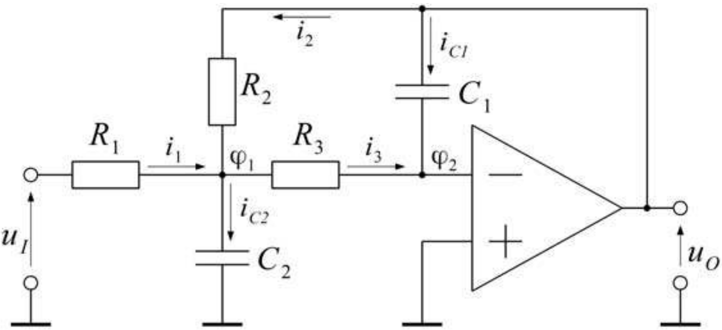

Fig. 6. Second order system

It is required:

1. To describe the filter by differential equation, similar to equation (3), and define its transfer function.
2. Assuming the circuit parameters $R_1 = 13$ kΩ; $R_2 = 33$ kΩ; $R_3 = 13$ kΩ; $C_1 = 2$ μF; $C_2 = 47$ μF, to calculate gain K, ω and ρ.
3. Using MATLAB plot Bode diagrams of magnitude and phase angle.

Mathematical model of the circuit in Fig. 6 is derived using Kirchhoff‘s laws. For simplicity the operational form of equations is used and Laplace transform is applied for variables: $U_I \equiv L[u_I]$; $U_O \equiv L[u_O]$; $\Phi_1 \equiv L[\varphi_1]$; $\Phi_2 \equiv L[\varphi_2]$, where L denotes Laplace transform. Regarding directions of currents, for nodes φ_1 and φ_2 the equations are valid:

$$\begin{cases} \dfrac{U_I - \Phi_1}{R_1} + \dfrac{U_O - \Phi_1}{R_2} - \dfrac{\Phi_1 - \Phi_2}{R_3} - \dfrac{\Phi_1}{1/(C_2 s)} = 0; \\ \dfrac{\Phi_1 - \Phi_2}{R_3} + \dfrac{U_O - \Phi_2}{1/(C_1 s)} = 0. \end{cases} \tag{61}$$

Application of virtual ground principle gives assumption $\varphi_2 \approx 0$. Then (61) is rewritten in this way:

$$\begin{cases} \dfrac{U_I - \Phi_1}{R_1} + \dfrac{U_O - \Phi_1}{R_2} - \dfrac{\Phi_1}{R_3} - \Phi_1 C_2 s = 0; \\ \dfrac{\Phi_1}{R_3} + \dfrac{U_O}{1/(C_1 s)} = 0, \quad \to \quad \Phi_1 = -U_O C_1 R_3 s. \end{cases} \tag{62}$$

Substituting the second equation of set (62) to the first one yields:

$$\frac{U_I + U_O C_1 R_3 s}{R_1} + \frac{U_O + U_O C_1 R_3 s}{R_2} + U_O C_1 s + U_O C_1 C_2 R_3 s^2 = 0. \tag{63}$$

Rearranging the expression by leaving voltages u_I and u_O in different sides of equation gives:

$$U_I = -U_O \frac{C_1 C_2 R_1 R_2 R_3 s^2 + C_1 s\left(R_2 R_3 + R_1 R_3 + R_1 R_2\right) + R_1}{R_2}. \tag{64}$$

Thus the system transfer function is:

$$G(s) \equiv \frac{U_O}{U_I} = -\frac{R_2/R_1}{C_1C_2R_2R_3s^2 + C_1s\left(R_2R_3/R_1 + R_3 + R_2\right) + 1}. \quad (65)$$

Expression (65) has five variable parameters: R_1, R_2, R_3, C_1 and $C_{2,}$ but the second order system is characterized by three parameters: K, ω_0 and ρ has no unambiguous solution.

In this case the values of circuit elements R_1, R_2, R_3, C_1 and C_2 are calculated:

$$\frac{1}{\omega_0^2} = C_1C_2R_2R_3; \quad \rightarrow \quad \omega_0 = \frac{1}{\sqrt{C_1C_2R_2R_3}} \approx 5; \qquad (66\text{ a})$$

$$\frac{2\rho}{\omega_0} = C_1\left(\frac{R_2R_3}{R_1} + R_3 + R_2\right); \rightarrow$$
$$\rightarrow \rho = \frac{C_1\left(\frac{R_2R_3}{R_1} + R_3 + R_2\right)}{2}\omega_0 \approx 0,4; \qquad (66\text{ b})$$

and

$$K = -\frac{R_2}{R_1} = -\frac{33 \cdot 10^3}{13 \cdot 10^3} \approx -2,54. \qquad (66\text{ c})$$

The similar system has already been analyzed in this work. Bode plots can be constructed using specialized commands:

```
G_sist = tf(-2.54, [1/25, 2*0.4/5, 1])
Bode(G_sist)
```

While similar system was already discussed, the plots are not repeated here.

Content of the report

1. Elaboration of mathematical model of the system, indicated by the teacher.
2. Plots of steps responses and asymptotic Bode plots.
3. Frequency response magnitude and phase angle characteristics calculated and plotted by Matlab.

Control questions

1. How do differential equation and transfer function of the second order system look like?
2. What a physical meaning of parameters K, ω_0 and ρ?
3. What values of ρ correspond to oscillating step response?
4. Can the system have overshoot of time response if $\rho \geq 1$?
5. What is the step response at $\rho = 0$?
6. How does frequency response polar plot look like?

ANALYSIS OF ALGEBRAIC METHODS OF STABILITY

Laboratory work No. 5

Objectives: to get aquainted with algebraic methods of system stability and to learn to apply them.

Tasks:

1. Elaborate mathematical model of the system, indicated by the teacher.
2. Check stability of the system by algebraic methods.

Theoretical part

In general the differential equation, describing the system, has the form:

$$\sum_{k=0}^{N} a_k \frac{d^k y}{dt^k} = \sum_{l=0}^{M} b_l \frac{d^l u}{dt^l}; \tag{1}$$

where: a_k and b_l are the constants, y is system output signal.

Solution of equation (1) is expressed by sum of two solutions: the general solution of homogeneous equation and particular solution.

Homogeneous equation is the same equation as (1) with the right hand side set equal to zero:

$$\sum_{k=0}^{N} a_k \frac{d^k y}{dt^k} = 0. \tag{2}$$

Its solution called (free response (movement)) is found using the roots of characteristic polynomial:

$$d(\lambda) = \sum_{k=0}^{N} a_k \lambda^k. \tag{3}$$

Roots of characteristic polynomial, set equal to zero, are called system poles and have notation p_n, where n = 1, 2, ..., N. It is evident, that $d(p_n) = 0$.

Linear system can be applied as principle of superposition, therefore it is possible to discuss contribution of each homogeneous equation root to solution of the system, and the overal solution is the sum of all contributions.

Therefore all roots are classified as:
1) aliquant roots;
2) multiple roots.

If the system has G aliquant roots, then their contribution to solution of homogeneous equations is:

$$y_{h1} = \sum_{d=1}^{D} C_d \exp\left(p_d t\right); \tag{4}$$

where: d is index of aliquant root.

Expression (4) is valid for real and complex aliquant roots, but its application for complex roots gives complex constants C_d, therefore (4) has to be rearranged.

It is evident that complex roots always appear as complex conjugate numbers. Denoting are pair of complex roots $p_\pm = \alpha \pm \beta j$ and applying Eq. (4) yields:

$$\begin{aligned} y_{h2} &= c_1 \exp\left[\left(\alpha+\beta j\right)t\right] + c_1 \exp\left[\left(\alpha+\beta j\right)t\right] = \\ &= c_1 \exp\left(\alpha t\right)\exp\left(\beta j t\right) + c_2 \exp\left(\alpha t\right)\exp\left(-\beta j t\right) = \\ &= \exp\left(\alpha t\right)\left[c_1 \cos\left(\beta t\right) + c_1 \sin\left(\beta t\right) j + c_2 \cos\left(-\beta t\right) + c_2 \sin\left(-\beta t\right) j\right]; \end{aligned} \tag{5}$$

where: c_1 and c_2 are constants, therefore assuming $C_1 \equiv c_1 + c_2$; $C_2 \equiv c_1 j - c_2 j$, solution (5) can be rewritten in this form:

$$y_{h2} = C_1 \exp\left(\alpha t\right)\cos\left(\beta t\right) + C_2 \exp\left(\alpha t\right)\sin\left(\beta t\right). \tag{6}$$

Constants C_1 and C_2 are calculated according to initial conditions and are real.

In general there may be several groups of multiple real roots. For example, $(\lambda + 1)^2(\lambda + 2)^2 = 0$ has 4 roots, which can be devided in two groups of multiple roots: $p_{1,2} = -1$ and $p_{3,4} = -2$. If the group of multiple roots has E multiple roots $p_1 = p_2 = ... = p_E \equiv p$, then contribution of those roots to the solution of homogeneous equation is described as:

$$y_{h3} = \exp(t\,p)\sum_{e=1}^{E} C_e t^{e-1}; \tag{7}$$

where: e is index of multiple root in the group. Expressions (7) and (4) can be applied for complex values of characteristic equation roots. Complex roots repeat themselves by pairs of complex conjugate numbers. Thus, if a pair of roots $p_{\pm} = \alpha \pm \beta j$ repeats itself E times, then its contribution to solution of homogeneous equation is:

$$y_{h4} = \exp(\alpha t)\sum_{e=1}^{E} t^{e-1}\left[C_{2(e-1)}\cos(\beta t) + C_{2(e-1)+1}\sin(\beta t)\right]. \tag{8}$$

Thus, summarizing we can say, that contribution of each aliquant root to solution of homogeneous equation is calculated by (4), each pair of aliquant complex roots – by (5), each multiple real roots group – by (7) and each multiple pair of complex roots group – by (8) expression.

System stability is called system availability to remain in steady-state in absense of external disturbance or to acquire new steady-state value after external disturbance, or to come back to steady-state from other state when external disturbance dissapears. In mathematical meaning, the solution of homogeneous Eq. (2) at $t \to \infty$ asymptotically approaches to zero:

$$\lim_{t\to\infty} y_h = 0. \tag{9}$$

One more definition of stability is realated to behaviour of asymptotic non-homogeneous equation (1) solution. It is stated that the system is stable if the condition:

$$\exists Y \neq \pm\infty : \lim_{t\to\infty} y = Y; \tag{10}$$

is fulfilled.

While solution of non-homogeneous equation depends on input signal u, therefore the limit of y for different u is different.

The last definition of stability states, that if the system is stable at bounded input $|u| \leq M$, where M is finite real number, then the system output y, if $t \to \infty$, also remains bounded. This definition is called „Bounded Input – Bounded Output“ (BIBO).

Linear system is valid superposition principle, therefore if the characteristic equation has no N roots, then its solution is equal to the sum of N contributions. Therefore the system is stable, when every solution, corresponding to every root, is stable.

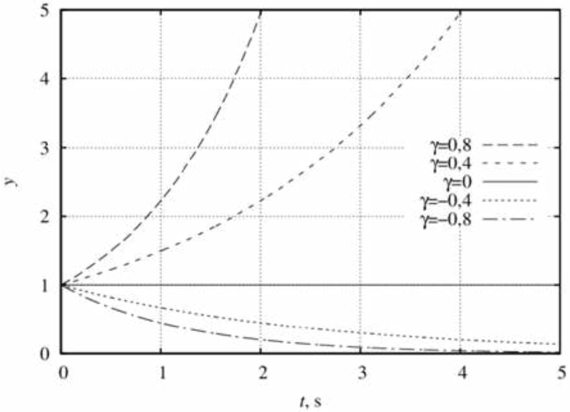

Fig.1. Set of graphs of function exp(αt) at different values of α

Analysis of solutions (3) and (4) corresponding to aliquant roots shows, that behaviour of asymptotic solutions at $t \to \infty$ depends on function exp(αt), where $\alpha \equiv \mathrm{Re}(p_n)$ asymptotic behaviour. Figure 1 shows set of functions exp(αt) at variable parameter α. It is evident, that exp(αt) $\to 0$ when $t\to\infty$ if $\alpha < 0$.

Solution (3) corresponds to aliquant real roots. Thus, the system is stable, if real roots or real parts of roots are negative: $\mathrm{Re}(p_n) < 0$. Analysis of solutions with multiple roots (7 and 8 expressions) can be seen, that they have product $t^e\cdot\exp(\alpha\ t)$, where $\alpha \equiv \mathrm{Re}(p_e)$. At $t\to\infty$, function $t^e\to\infty$. Fig. 1 shows at $t\to\infty$ function $t^e\cdot\exp(\alpha\ t)\to\infty$ at any positive α and the system will be unstable. If $\alpha > 0$, then the limit of the mentioned expression is calculated:

$$\lim_{t\to\infty} t^e \exp\left(-|\alpha| t\right) = \lim_{t\to\infty} \frac{t^e}{\exp\left(|\alpha| t\right)}. \tag{11}$$

At $t\to\infty$ value of numerator and denominator evenly rises and indeterminacy ∞/∞ appears. To solve this problem the Lopital's rule is applied for (11) and numerator and denominator is differentiated e times:

$$\lim_{t\to\infty} t^e \exp\left(-|\alpha| t\right) = \lim_{t\to\infty} \frac{\dfrac{d^e t^e}{dt^e}}{\dfrac{d^e \exp\left(|\alpha| t\right)}{dt^e}} = \lim_{t\to\infty} \frac{e!}{|\alpha|^e \exp\left(|\alpha| t\right)} = 0. \tag{12}$$

Thus, at $\alpha < 0$ product $t^e\cdot\exp(\alpha\ t)\to 0$ with $t\to\infty$, therefore for this case solutions corresponding to multiple roots are stable.

If $\mathrm{Re}\ p_n = 0$, it can not be said unambiguous about system stability, it depends on poles type and repeatability. If multiple poles appear on the imaginary axis, the expression t^{e-1} in (7) and (8) expressions make the system unstable. If the real parts of all other roots are negative and aliquant complex poles appear on the imaginary axis, then solution (6) at ($\alpha = 0$) can be rewritten as:

$$y_{h2} = C_1 \cos\left(\beta t\right) + C_2 \sin\left(\beta t\right). \tag{13}$$

According to (13), the undamped oscillations occure in the system and this system is stable by the meaning „bounded input-bounded output“, but it is not stable according to asymptotic solution meaning, because (9) is not valid. It is said that this system is on the boundary of stability.

Real pole in this point $p_n = 0 + 0j$ corresponds to differential equation:

$$a\frac{dy}{dt} = u; \quad \rightarrow \quad y = \frac{1}{a}\int u dt. \tag{14}$$

If $u = 1(t)$ (unit step), then at zero initial conditions:

$$y = \frac{1}{a}\int 1 dt = \frac{t}{a}. \tag{15}$$

It is evident, though $y \to \infty$ and $t \to \infty$, nevertheless input signal is bounded and the system is unstable.

The analysis shows, that stability of the system can be defined from roots of characteristic equation. According to this statement, the algebraic criterion of stability is formulated in this way:

1. The system is stable when all real roots or real parts of roots system roots are negative.
2. The system is unstable if even one root of characteristic equation is positive.
3. The system is on the border of stability (i.e. undamped oscillation occurs), if it has no roots with positive real part and has only imaginary roots.

Therefore roots of characteristic equation (3) define stability of the system. Roots of characteristic equation are set in the complex plane and this is called pole map. If all roots appear in the left hand semiplane, the system is unstable (Fig. 2).

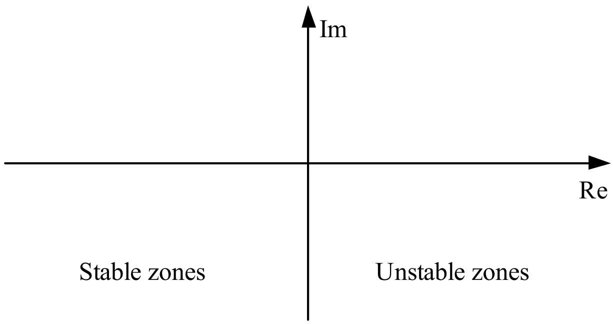

Fig. 2. Stable and unstable zones in complex plane

If the characteristic equation has $N \leq 2$ order, the roots can be easily found in analytical way. It is difficult or impossible to calculate roots of higher order system in this way, therefore software *Matlab*, *Octave* has the commands for calculation of `roots`. *Matlab* has command `roots`, which argument is vector of polynomial coefficients:

```
roots(coef_vect)
zone_vect = roots(coef_vect)
```

Matlab has specialized commands for calculation of system poles (the same as system roots) and depicting that in complex plane.

Control System Toolbox package has command pole, which calculates the system‘s pole. The form of that command is:

```
p_vect = pole(G_sys)
```

where `p_vect` is returnable vector with calculated system poles, `G_sys` is transfer function, described by zero-pole or state variables form (see Lab. Work No 1).

The same Control System Toolbox has command `pzmap`, which depicts system zeros and poles in the complex plane. Its form is:

```
pzmap(G_sys)
[p_vect, z_vect] = pzmap(G_sys)
```

where `z_vect` is zeroes, i. e. vector of coefficients of transfer function polynomial.

Numerical methods of roots calculation does match system analysis problems, because they do not give relationship between roots and coefficients of characteristic polynomial for evaluation of system stability, it is enaugh to know only signs of the roots, their exact values are not required. At least all numerical methods for calculation of characteristic equation roots are rather complex, they can not be applied in absence of computer.

Rauth-Hurwitz method is algebraic metod and has no mentioned shortcomings. It allows discussing stability of the system without finding the roots.

There are two formulations of criterion: Rauth and Hurwitz.

Hurwitz formulation is as this:
if the system polynomial is:

$$a_n s^n + a_{n-1} s^{n-1} + ... + a_1 s + a_0 = 0 \tag{16}$$

then, the system will be stable, when all polynomial coefficients a_n, a_{n-1},..., a_1, a_0 are positive and determinants are positive:

$$D_1 = a_1 > 0;\ D_2 = \begin{vmatrix} a_1 & a_3 \\ a_0 & a_2 \end{vmatrix} > 0;\ D_3 = \begin{vmatrix} a_1 & a_3 & a_5 \\ a_0 & a_2 & a_4 \\ 0 & a_1 & a_3 \end{vmatrix} > 0; \tag{17}$$

$$D_4 = \begin{vmatrix} a_1 & a_3 & a_5 & a_7 \\ a_0 & a_2 & a_4 & a_6 \\ 0 & a_1 & a_3 & a_5 \\ 0 & a_0 & a_2 & a_4 \end{vmatrix} > 0; \cdots D_n = \begin{vmatrix} a_1 & a_3 & a_5 & \dots & 0 \\ a_0 & a_2 & a_4 & \dots & 0 \\ 0 & a_1 & a_3 & \dots & 0 \\ 0 & a_0 & a_2 & \ddots & 0 \\ 0 & 0 & \dots & \dots & a_n \end{vmatrix} > 0. \quad (18)$$

Application of table of *n* rows and *n* colomns facilitates calculation of determinants:

$$\begin{vmatrix} a_1 & a_3 & a_5 & a_7 & a_9 & \dots \\ a_0 & a_2 & a_4 & a_6 & a_8 & \dots \\ 0 & a_1 & a_3 & a_5 & a_7 & \dots \\ 0 & a_0 & a_2 & a_4 & a_6 & \dots \\ 0 & 0 & a_1 & a_3 & a_5 & \dots \\ 0 & 0 & a_0 & a_2 & a_4 & \dots \\ \dots & & & & & \end{vmatrix}$$

The first row is fulfilled by coefficients with odd numbers, starting from a_1, the second row – by coefficients with even numbers, starting from zero. The third row repeats the first one and is shifted by one position to the right and the fourth is the same as the second, also is shifted and repeats the third one, and the sixth is the same as the fourth, but shifted to the right. In this way the table is fulfilled until its row number becomes equal to *n*.

Then the determinants, separated by dash lines, are calculated.

Calculation of determinants higher than 3 order is difficult. Therefore Rauth present a simplyfied formulation of criterion not including high order determinants.

For Rauth formulation the table of characteristic equation (16) coefficients is composed. The general view of the table is:

a_n	a_{n-2}	a_{n-4}	a_{n-6}
a_{n-1}	a_{n-3}	a_{n-5}	a_{n-7}
c_1	c_3	c_5	c_7
d_1	d_3	d_5	d_7
e_1	e_3	e_5	e_7
.....			

The first two rows of the table are made directly from coefficients of characteritic equation.

Then coeffcients c are calculated as:

$$\begin{aligned}
c_1 &= \operatorname{sgn}\left(a_{n-1}\right)\cdot\left(a_{n-1}a_{n-2} - a_n a_{n-3}\right);\\
c_3 &= \operatorname{sgn}\left(a_{n-1}\right)\cdot\left(a_{n-1}a_{n-4} - a_n a_{n-5}\right);\\
&\vdots\\
c_i &= \operatorname{sgn}\left(a_{n-1}\right)\cdot\left(a_{n-1}a_{n-i-1} - a_n a_{n-i-2}\right);
\end{aligned} \tag{19}$$

Where $\operatorname{sgn}\left(a_{n-1}\right)$ is sign function of coefficient a_{n-1}, defined as: $\operatorname{sgn} a_{n-1} = a_{n-1}/\left|a_{n-1}\right|$.

Coefficients d are calculated from expression:

$$\begin{aligned}
d_1 &= \operatorname{sgn}\left(c_1\right)\cdot\left(c_1 a_{n-3} - a_{n-1}c_3\right);\\
d_3 &= \operatorname{sgn}\left(c_1\right)\cdot\left(c_1 a_{n-5} - a_{n-1}c_5\right);\\
&\vdots\\
d_i &= \operatorname{sgn}\left(c_1\right)\cdot\left(c_1 a_{n-i-2} - a_{n-1}c_{n-i-2}\right).
\end{aligned} \tag{20}$$

Coefficients e are calculated in a similar way:

$$\begin{aligned}
&e_1 = \operatorname{sgn}(d_1)\cdot(d_1c_3 - c_1d_3);\\
&e_3 = \operatorname{sgn}(d_1)\cdot(d_1c_5 - c_1d_5);\\
&\vdots\\
&e_i = \operatorname{sgn}(d_1)\cdot(d_1c_{n-i-2} - c_1d_{n-i-2}).
\end{aligned} \tag{21}$$

Accordingly with the system order, it is possible to make more rows. Rauth table number of rows and number of columns depend on order of characteristic polynomial. The table has n+1 row and (n/2 + 1) columns. Rauth stability criterion is formulated in this way: system is stable if all coefficients of the first row of the table are not equal to zero and have the same sign.

Matlab do not have standart function for this purpose, despite in the internet – it can be found in *Matlab* scenario generating the Rauth tables.

Examples of algebraic stability metods application

Example 1.

Given: two objects with transfer functions:

$$\begin{aligned}
&G_1 = \frac{5s+3}{s^2+1,2s+1};\\
&G_2 = \frac{5s+3}{s^5+8s^4+6s^3+5s^2+4s+1}.
\end{aligned} \tag{22}$$

Required:

1. Plot unit response of system $G_1(s)$ and $G_2(s)$.
2. Depict the system poles and zeroes in the complex plan and decide about system stability.

3. Check stability of the system, described by transfer function $G_1(s)$, using Hurwitz formulation of Rauth-Hurwitz method.
4. Check stability of the system, described by transfer function $G_2(s)$, using Hurwitz formulation of Rauth-Hurwitz method.
5. Check stability of the system, described by transfer function $G_2(s)$, using roots method.
6. Plot step response of the system, described by transfer function $G_2(s)$.

Solution:

While both systems are described by transfer functions, then the objects, corresponding to those, are developed using `tf` commands:

```
G_1 = tf([-5, 3], [1, 1.2, 1])
G_2 = tf([-5, 3], [1, 8, 6, 5, 4, 2])
```

Step responses of both functions are calculated by `step` command:

```
t_vect = [0:0.01:10]
y1_vect = step(W_1, t_vect)
y2_vect = step(W_2, t_vect)
plot(t_vect, y1_vect, t_vect, y2_vect)
```

Graphs of step responses are shown in Fig. 3.

Figure 3 shows, that system $G_1(s)$ is stable and $G_2(s)$ is unstable, because step response of $G_1(s)$ approaches to steady-state value and that of $G_2(s)$ oscillating increases to infinity.

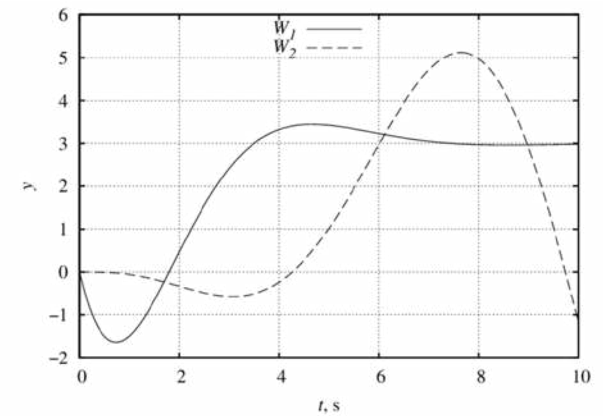

Fig. 3. Step responses of systems $G_1(s)$ and $G_2(s)$

Displacement of zeroes and poles in complex plane is obtained by `pzmap` command:

```
pzmap(G_1)
```

which gives the plot, shown in Fig. 4.

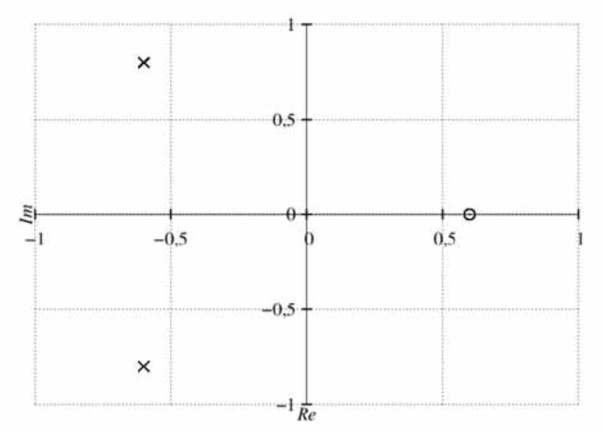

Fig. 4. Zeros and poles map of system $G_1(s)$

Zeros of transfer function are designated by „0“ and poles „x“ symbols. Fig. 4 shows that all system poles, that mean roots of characteristic equation are located on the left hand side semiplane of complex plane, therefore the system is stable.

The characteritic equation of the system is:

$$s^5 + 8s^4 + 6s^3 + 5s^2 + 4s + 2 = 0$$

where: $a_5 = 1$; $a_4 = 8$; $a_3 = 6$; $a_2 = 5$; $a_1 = 4$ ir $a_0 = 2$

The Hurwitz determinant is constructed as this:

$$D_3 = \begin{vmatrix} a_1 & a_3 & a_5 \\ a_0 & a_2 & a_4 \\ 0 & a_1 & a_3 \end{vmatrix} = \begin{vmatrix} 4 & 6 & 1 \\ 2 & 5 & 8 \\ 0 & 4 & 6 \end{vmatrix} = -72.$$

Hurwitz determinant in the third order determinant. The other determinants are:

$$D_1 = |a_1| = 4 > 0;\ D_2 = \begin{vmatrix} a_1 & a_3 \\ a_0 & a_2 \end{vmatrix} = \begin{vmatrix} 4 & 6 \\ 2 & 5 \end{vmatrix} = 8 > 0.$$

Hurwitz stability formulation states, that the system is stable, when Hurwitz determinant and all the lower order determinants are positive. Hurwitz determinant of this system $D_3 < 0$ is negative, therefore the system is unstable. Application of Rauth formulation requires to construct the table, which looks like this:

1	$a_5 = 1$	$a_3 = 6$	$a_1 = 4$
2	$a_4 = 8$	$a_2 = 5$	$a_0 = 2$
3	$c_1 = a_4 a_3 - a_5 a_2 = 43$	$c_3 = a_4 a_1 - a_5 a_0 = 30$	$c_5 = 0$
4	$d_1 = c_1 a_2 - a_4 c_3 = -25$	$d_3 = c_1 a_0 - a_4 c_5 = 86$	$d_5 = 0$
5	$e_1 = (-1)(d_1 c_3 - c_1 d_3) = 4448$	$e_3 = 0$	$e_5 = 0$

Analysis of the first column shows, that the coeffcients have different signs, therefore the system $G_2(s)$ is unstable. While the sign of coefficients changes twice, passing from the third row to the fourth and from the fourth to the fifth, it can be concluded, that the system has two positive roots on the right hand semiplane.

Content of report

1. Elaboration of mathematical model describing the system by transfer function.
2. Check stability of the system by algebraic method of roots and Rauth-Hurwitz method.
3. Conclusions.

Control questions

1. How do you understand concept „stable system“? What are defitions of stable system?
2. How is the algebraic root stability criterion formulated?
3. How do you understand concept „marginal stability“?
4. If the Rauth-Hurwitz criterion is actual, what are advantages of that criterion?
5. What are formulations of Rauth-Hurwitz criterion?
6. Is the system stable, if its output signal is periodic rectangular impulse?

REFERENCES

Aström K. J., Murray R. M. *Feedback Systems*. 2008. Princeton University Press.

Bishop R. H. *Modern Control Systems Analysis and Design with Matlab*. Prentice Hall, 2005. 1018 p.

Brogan W. L. *Modern Control Theory* (3rd Edition), 1990.

Bubnicki Z. *Modern Control Theory*, Springer, 2002, 421 p.

Burns R. S. *Advanced Control Engineering.* Butterworth Heinemann, 2001, p. 465, p. 450.

Chi-Tsong Chen. *Linear System Theory and Design,* International 3rd. edition, Oxford Series in Electrical and Computer Engineering. 1998. 334 p.

Close Ch. M., Frederick D. K., Newell J. C. *Modeling and Analysis of Dynamic Systems*, 3rd edition, 2001. 576 p.

Franklin G. F., Powell J. D., Emami-Naeini A. *Feedback Control of Dynamic Systems: 5th ed*. 2006. 910 p.

Golnaraghi F., Kuo B. C. *Automatic Control Systems*. Wiley. 2010. 786 p.

Internet access: http://pweb.yazduni.ac.ir/engineering/mech/bozorg/cours-es/courses.htm.

Kuo B. C. *Automatic Control Systems*. John Wiley & Sons.2002. 378 p.

Kuo B. C. 77H*Automatic Control Systems.* 6th ed. Englewood Cliffs (N.J.): Prentice Hall, 1992, 760 p.

Lurie B. J. *Classical Feedback Control with Matlab*. Marcel Dekker, Inc. 2000. 456 p.

Michel A. N., Hou L., Liu D. *Stability of Dynamical Systems. Continuous, Discontinuous and Discrete Systems*, Birkhauser, 2008. 515 p.

Nagratah I. J. *Control Systems Engineering*. 2005. 858 p.

Nise Norman S. *Control Systems Engineering*. John Wiley & Sons, 2004. 983 p.

Ogata K. *Modern Control Engineering*. 2010. 894 p.

Ogata K. *Matlab for Control Engineers*. 2007. 433 p.

Van de Vegte J. 78H*Feedback Control Systems*. Englewood Cliffs: Prentice Hall, 1994. 451 p.