# Lab 1. MODELING OF DYNAMIC SYSTEM RESPONSE ON UNIT STEP INPUT (weeks of 09.10.2023, 16.10.2023; due date 26.10.2023)

# 1. Theoretical part

## 1.1. Analytical consideration of dynamic systems

Let’s consider controlled object behavior of which can be described by ordinary differential equation of the 2-nd order

, (1)

where y denotes system output, x – system input, , y=y(t), x=x(t), t – time variable. Coefficients are assumed here to be constant, and  (characterizes friction),  (characterizes spring forces for mechanical systems). Equation (1) may be used for description of many real systems such as electrical motors, liquid control systems, mechanic translational systems and so on.

To design controllers, it is necessary to investigate system behavior, or how does it react on various input signals (reference inputs). Typical for consideration input signals are: unit step, impulse, ramp function and sine function. We shall consider now response of the system (1) on unit step input.

Unit step is defined as follows:

1(t) =  (2)

Unit step function means that before t=0 input was 0, and at moment t=0 it becomes equal to 1. We assume that initial state of the system output is also zero:

 (3)

So, our problem is to solve (1),(3), assuming that

x(t)=1(t). (4)

Controlled system should be stable. It may be shown that system is stable if its characteristic equation

 (5)

has roots with negative real part (in general case, roots may be complex numbers having real, Res, and imaginary, Ims, parts: s=Res+jIms, where j= - imaginary 1).

Roots of (5) are

 (6)

There may be considered 3 cases:

Case 1.  This case corresponds to not existence of real roots of (5): both roots are complex with the same real part () and imaginary parts differing in sign (). So, roots are:

 (7)

For system to be stable, we require that  (real parts of the roots (7) should be negative).

It can be shown that solution of (1), (3), (4) under given conditions of Case 1 is

 (8)

where , . It is obvious that , i.e. output will become proportional to the input reference with time running, and this approach to the limit value is performed in oscillating mode with frequency .

Case 2. . This case corresponds to existence of 2 identical real roots of characteristic equation (5):



In such a case solution of (1), (3), (4) will be

 (9)

We also see asymptotic coming of y to  with time running, but in this case it is made without oscillations.

Case 3. . In this case characteristic equation (5) has 2 different real roots:



and respective solution of (1), (3), (4) will be

 (10)

where both roots  are negative, so, this function will approximate  with time .

If to introduce parameter, then we can observe that Case 1 corresponds to , Case 2 – to , Case 3 – to . Behavior of the system output is shown pictorially in the figure below for various values of  (all curves try to reach value of 1).



Fig.1 (taken from Katsuhito Ogata, Modern Control Engineering, Prentice Hall, Englewood Cliffs, N.J., 1970, 836 p.,ISBN 13-590232-0, p. 232).

Time scale in the figure above is selected according to the character system time which is defined by value of  - character frequency of the system in the absence of friction (). Unit of time in the picture is chosen as, which results in one full period of oscillations during  =6.28 units of time.

We can state that given above considerations correspond to analog way of controlling of the system, but for digital control we need discrete time and we are to choose step of time discreteness providing approximation agreeing with considered above results.

## 1.2. Discrete approximation of system dynamics

Let’s again consider problem (1), (3), (4) – we are to calculate response of the system on unit step input signal. Now we shall use finite difference approximations of the derivatives and, respectively, discrete time. It means that we shall consider values of functions only at the moments of time, where T is a time step. Derivatives may be represented by

 (11)

If we shall introduce, then (1) may be rewritten as a system of the two 1st order ordinary differential equations:

 (12)

Using similar to (11) approximation for, from (11), (12) we can get

 (13)

If to denote



then (13) can be rewritten:

 (14)

From (14) we can write out working formulae for numerical calculation of system outputs each next moment of the time depending on the values of input and output at the previous moment of the time:

 (15)

Equations (15) represent Euler method of solving ordinary differential equations (12). It provides little accuracy. More accurate and is widely used the fourth-order Runge-Kutta method. Let’s consider it. Again, we assume that a second-order equation is given in general form:

 (16)

accompanied by initial conditions that x= and  when t=. In the case of equation (1), equation (16) looks as follows:

.

By calling the first-order derivative a new variable p, the equation (16) can be written in the form of two simultaneous first-order equations

 (17)

with initial values . The fourth-order Runge-Kutta formulas for marching from  to  are

 (18)

in which the individual terms are computed in the following order:

 (19)

Time step T (it is denoted by h in (19)) should be selected to be sufficiently small in comparison with character time scale of the system (there should be at least 2 time steps inside the period of oscillations). For example, in Fig.1 such a time step is equal to. Usually, finite differences approximation works well if T<<.

# 2. What to do?

1. Program an interface for getting parameters of the system (1) - 

2. Write a program of computation of the system output according to formulae (8)-(10) corresponding to all the three considered cases.

3. Provide graphical output of the results of computations similar to presented in Fig.1

4. Run your program and by modification of the system parameters get graphics similar to shown in Fig. 1 for all 3 Cases. For example,  corresponds to Case 1 (2 complex conjugated roots),  to Case 2 (2 equal roots), and to Case 3 (2 different real roots).

5. Select time step T=0.1\* (10 steps inside one period of oscillations)

6. Write a program of computation of output using numerical approximations (15), (18)-(19)

7. Provide graphical output of both numerical (obtained by Euler (15) and Runge-Kutta (18)-(19) methods) and exact solutions (8)-(10) for all 3 Cases. Compare them. In the case of discrepancies, reduce time step (T in (15), h in (18), (19)) twice and so on until you will get good agreement of the numerical and exact solutions.

**You may use any programming languages and tools available in the Lab for task implementation and next demonstration.**

# 3. Defense of the work

You are to show developed working product to evaluator and to be able to explain

* what was done,
* what for it was done,
* how it was done.

**You are to prepare and printout report on the work, in which**

1. Show the title of the work and your name(s)
2. Formulate task of the work (requirements) – 1-2 pages
3. Describe team members’ responsibilities
4. Describe your design (algorithm, interfaces) – 1-3 pages
5. Describe program implementation (functions, their goals, interfaces) – 1-2 pages
6. Give snapshots of the working program showing its proper work for testing examples, compare accuracy of Euler and Runge-Kutta methods - 5-6 snapshots
7. Conclusion – brief summary of the work done - 1 page

**Zip file with all Lab related materials (report, sources, executables) shall be uploaded in the Teams assignment (to be posted by Lab Coordinator** [LEILA MOHAMMADIAN VAIGHAN](https://cmpe.emu.edu.tr/tr/hakkimizda/personel/personel-detayi?sid=266&n=leila-mohammadian-vaighan))

**Copies are not allowed and will be zero graded.**

**Teams are at most of 4 people, one report/team**

**Due date for submission of the paper report – 26.10.2023, Thursday, 16.30-17.20, CMPE239.**

**Grading policy: 40% report, 60% explanations on the implementation**