Department of Electrical and Computer Engineering

ECED4601 Digital Control System

Lab1 Time responses of continuous-time and discrete-time systems

1 Introduction

The objective of this lab is to show the students how to use MATLAB and Simulink to create models of discrete-time systems either in transfer functions or state-space form and to analyze digital control systems.

2 From continuous time to discrete time

To better understand how a digital system works, it is necessary to see the effect of sampling a continuous-time signal. Consider the system shown in Figure 1 (see page 184 in the textbook):

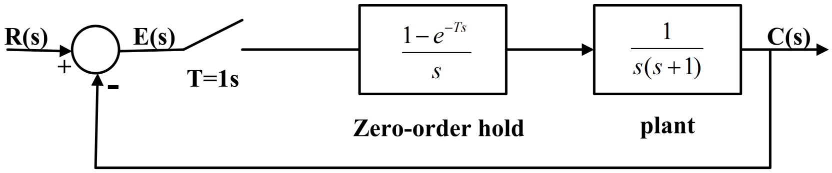


Figure 1: Discrete System Block Diagram

To simulate this system in MATLAB, open up a new m-file in MATLAB and type the following commands:

```
num=[1];den=[1 1 0];
% c2d
sysc = tf(num,den)
T = 1;
sysd = c2d(sysc,T,'zoh')
sys_c = d2c(sysd,'zoh')
% directly enter the transfer function in z domain
numz=[0.368 0.264];
denz=[1 -1 0.632];
sys1 = tf(numz,denz,T)
% closed loop system
sys_c_cl=feedback(sysc,1)
sys_d_cl=feedback(sysd,1)
% plot
step(sys_c_cl,':'),grid,hold
step(sys_d_cl)
[c,t] = step(sys_d_cl);
plot(t,c,'or');
legend('continuous system','discretized system','sampled data');
title('Step Response of continuous system and discretized system');
```

Now, let us break down the code

```
sysc = tf(num,den)
```

Construct the open-loop transfer function.

```
sysd = c2d(sysc,T,'zoh')
```

CT to DT conversion, sysc here is the plant in continuous-time, T is the sampling time, ‘zoh’ means discretizing the plant using zero-order hold.

```
sys_c = d2c(sysd,'zoh')
```

DT to CT conversion. Note that, we didn’t put ‘;’ in the end of the command so that we can directly see the resulting transfer function in the MATLAB *command window*.

```
sys_c_cl=feedback(sysc,1)
sys_d_cl=feedback(sysd,1)
```

Add a feedback loop to the open-loop system to obtain the closed-loop transfer function.

Run the m-file, you will get the plot in Fig. 2.

Check out sysc and sys_c from the output of MATLAB in *command window*, they should be almost the same.

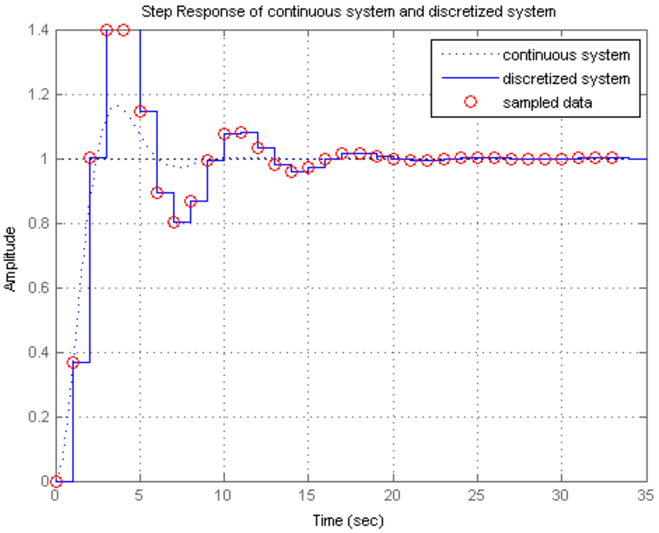


Figure 2: Step response of the system

To convert the transfer function to state-space form, use command 'tf2ss'. Try to convert the closed-loop discrete system (sys_d_cl) to state-space representation and include the result (value of A,B,C,D) in your report. (Hint: sys_d_cl.num and sys_d_cl.den to get the nominator and denominator of the transfer function, and you may also need 'cell2mat' function to do the class conversion)

3 Work in Simulink

Create the block diagram in Simulink, shown in Fig. 3:

Use the default settings in Simulink. Run the model, we can view the output to a step input in 'Scope' block.

Note that y axis of the scope is not scaled properly(Fig. 4). Hit the binoculars button to the autoscale the axes.

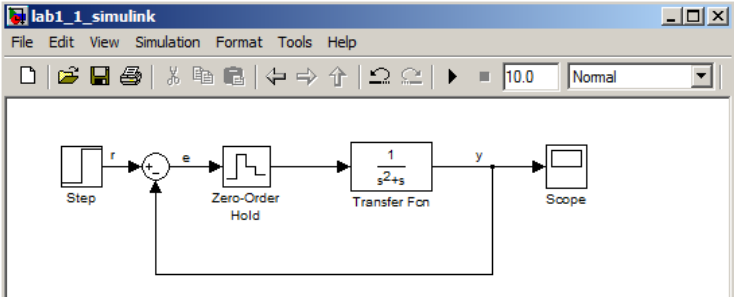
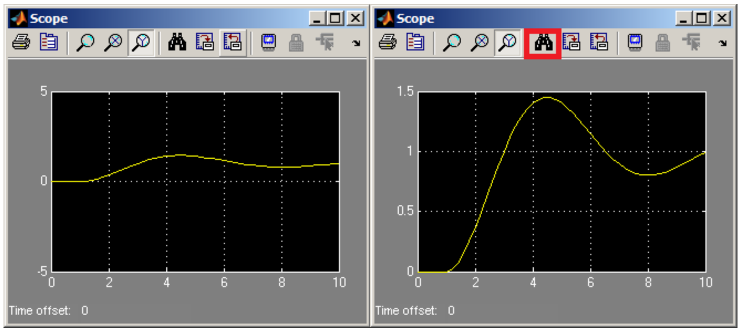


Figure 3: Simulink Model



(a) Before rescale

(b) After rescale

Figure 4: Autoscaling the scope

4 Exchanging Simulink with Matlab *command window*

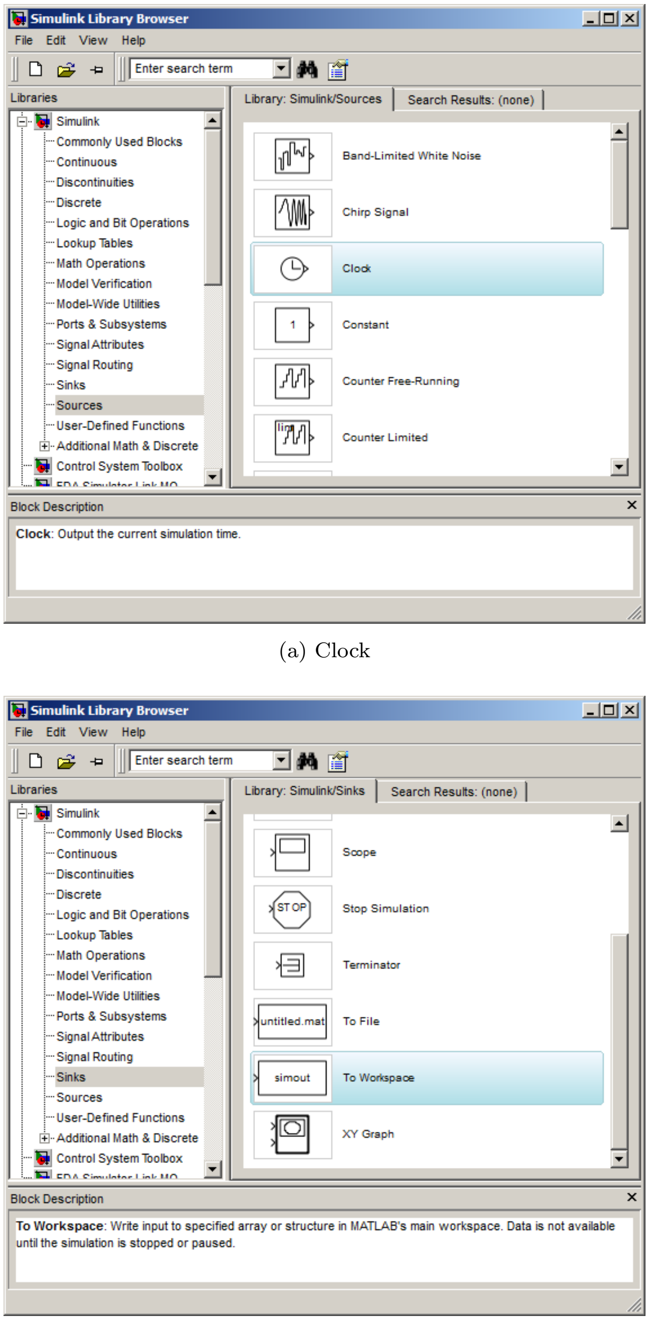
In some circumstances, we want to use the results from Simulink in MATLAB *command window* for further analysis. Variables, signals, and even entire systems can be exchanged between MATLAB and Simulink.

Suppose we want to use the output signal 'y' in MATLAB.

First, find the 'clock' block from 'Souces' window, drag it to the Simulink model file. Then open the 'Sinks' window, drag and drop 'To workspace' block, shown in Fig. 5. Connect all the blocks as shown in Fig. 6.

Double click the 'to workspace block' connected to y to bring up the block parameters window as shown in Fig 7.

Change the 'Variable name' to 'y' will output the signal to MATLAB variable 'y', and change the 'save format' to Array. To extract models from Simulink



Clock: Output the current simulation time.

(a) Clock

To Workspace: Write input to specified array or structure in MATLAB's main workspace. Data is not available until the simulation is stopped or paused.

(b) To Workspace

Figure 5: Connection Blocks

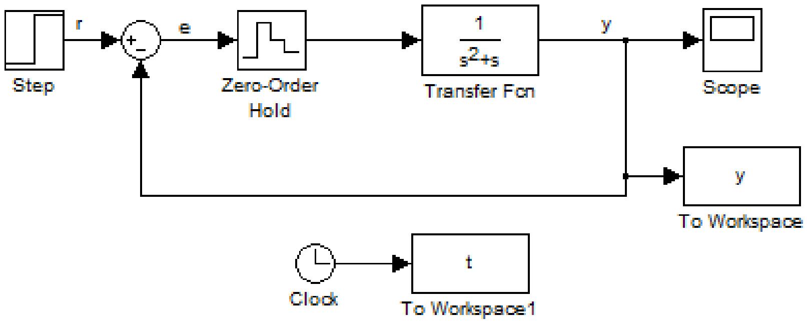
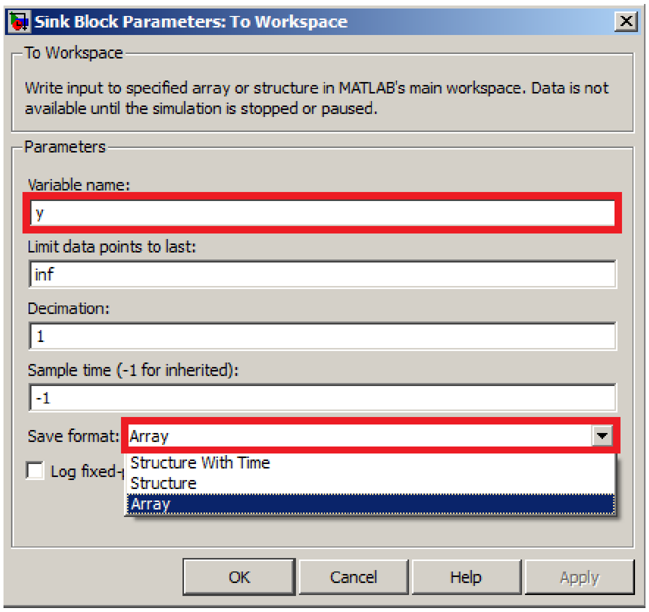


Figure 6: Exchange Variables


Sink Block Parameters: To Workspace

To Workspace

Write input to specified array or structure in MATLAB's main workspace. Data is not available until the simulation is stopped or paused.

Parameters

Variable name: y

Limit data points to last: inf

Decimation: 1

Sample time (-1 for inherited): -1

Save format: Array

□ Log fixed-

Figure 7: Parameters Window

into MATLAB, we need to tell Simulink where is the input and output of the model. This can be done using In and Out Connection blocks, from 'Sources' and 'Sinks' window respectively. We will extract the closed-loop model. Replace the Step block with the In block. And drag an Out block, and connect it to the output 'y'. The whole system should appear as follows

Save the model as "lab1_2_simulink.mdl". The following command can be used to extract a closed-loop transfer function from this model file.

```
[num_sim,den_sim]=dlinmod('lab1_2_simulink',1)
```

Note that, the ',1' in this command is the sampling period T.

Question: How to yield a state-space model using dlinmod() function?

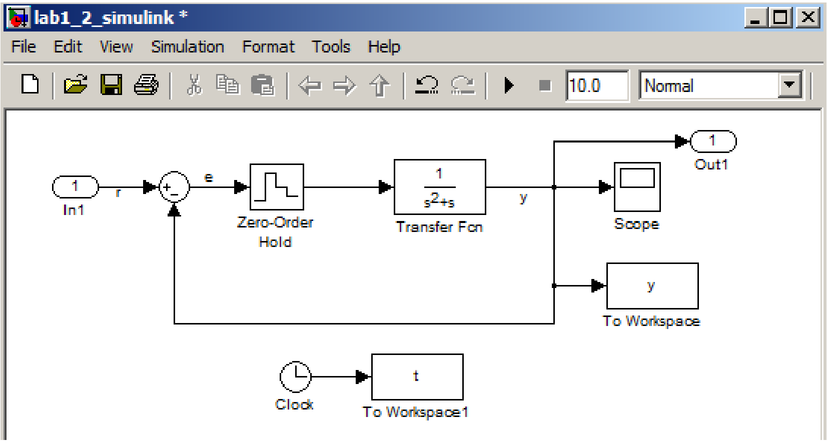


Figure 8: Simulink block of the in-out model

Include the command in your report (Hint: check out MATLAB help documentation on dlinmod())

5 System transient-response characteristics

Say the poles of the discrete system are $z_{1,2} = re^{\pm j\theta}$
The damping ratio ζ can be given by:

$$\zeta = \frac{-\ln r}{\sqrt{\ln^2 r + \theta^2}}$$

The natural frequency ω_n:

$$\omega_n = \frac{1}{T}\sqrt{\ln^2 r + \theta^2}$$

The time constant τ:

$$\tau = \frac{1}{\zeta\omega_n} = -\frac{T}{\ln r}$$

The characteristic equation of the system in Fig. 1 is

$$z^2 - z + 0.632 = 0$$

Open a new m-file in MATLAB, enter the following command:

```
p = [1 -1 0.632];
r = roots(p)
```

```
mag = abs(r(1,1))
theta = angle(r(1,1))
zeta = -log(mag)/(sqrt(log(mag)^2+theta^2))
wn = sqrt(log(mag)^2+theta^2)
tau = -1/log(mag)
```

Here, `roots()` gives the poles of the closed-loop system.

Questions: According to the output, what are the values of the damping ratio ζ , natural frequency ω_n and time constant τ? For the original continuous system as shown in Fig. 9, find ζ , ω_n and τ. (Hint: Transfer function

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Compare these 3 parameters of the continuous system and discrete system, explain how this is related to the step response shown in Fig. 2. What's the effect of discretizing a continuous-time system?(Hint: ζ is directly related to overshoot, τ is related with settling time)

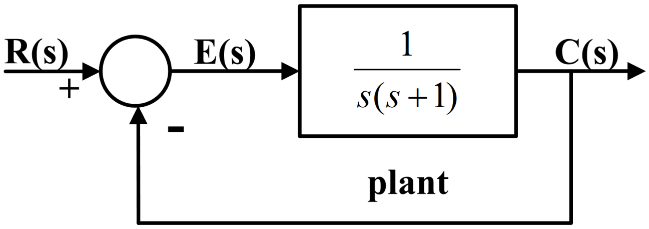


Figure 9: Continuous-time model

6 How sampling period impact the time response?

Change the sampling period to T=0.5s and T=0.1s. Implement the corresponding systems in Matlab, plot the closed-loop step response for the 3 systems(T=1s, T=0.5s, T=0.1s) and overlay on the same graph. And calculate the three parameters ζ , ω_n and τ.(Include the calculating process or the Matlab code, and the plot in your report). Based on the output you get, explain how sampling period affects the time response of the system.

7 Guidelines for writing the Lab Report

The report has to include:

- **Title page** - Including: course number, lab title, student names, student IDs, Date the report was submitted.
- **Abstract** - A summary of the contents of the lab report.
- **Procedure of methods or approach to the design or/and conduct of the experiments**
- **Diagrams** - Include all Simulink blocks used in the lab if any.
- **Plots** - All plots of system response should be included in your report, including title, labels with unit, legends, etc. Note that Diagrams and Plots must be computer generated. Hand-drawn plots will not be accepted. All diagrams and plots must be labelled. The labels and annotations can be done by hand if appropriate but have to be clear.
- **Code** - Include any code used in the lab. The code must be commented properly. It is good practice to put your code in separate m-files.
- **Discussion and answer to questions if any**.
- **Conclusion or any other relevant ideas**.