

NATIONAL UNIVERSITY OF SINGAPORE

Department of Electrical Engineering

Digital Control Systems

(EE3304)

EXPERIMENT 2

An Introduction to Computer Aided Digital Controller Design

(via Matlab and Simulink)

1. Introduction

Engineering workstations are rapidly becoming an essential part of the engineer's toolbox. Just as first the slide rule and then the calculator served to remove some of the drudgery from design work, design software package running on workstations allows engineers to work much more efficiently by eliminating the tedious and often repetitive time consuming tasks which are part of their routine work. There is now a wide variety of package available for control system design, many of which even run on personal computers.

One package, which has become popular, is Matlab. Matlab is a highly interactive software package. Its use of "M-files" allows one to write programs. Many algorithms used in control design work have been compiled and marketed as toolboxes. One such toolbox is the Control Systems Toolbox.

Simulink is a simulation package, which runs under Matlab. It contains a number of routines to integrate differential equations. These routines allow it to simulate continuous time and nonlinear systems. Simulink also allows one to build systems graphically. The graphical construction allows simulations to be easily constructed and complicated systems could be built easily.

2. Simulation with Simulink and Matlab

To run Matlab/Simulink, "double click" the simulink icon with your mouse. The **Matlab Command** window appears and the line "HELP, DEMO, and INFO are available." are displayed. The **Matlab Graph** window is also started and you can toggle between these windows using the **Window** menu.

2.1 Revision on Matlab

A list of all the available commands is available by typing **HELP**. To get help on a particular command, you can type **HELP** followed by the particular command name.

Matrices are defined in Matlab using square brackets. For example,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

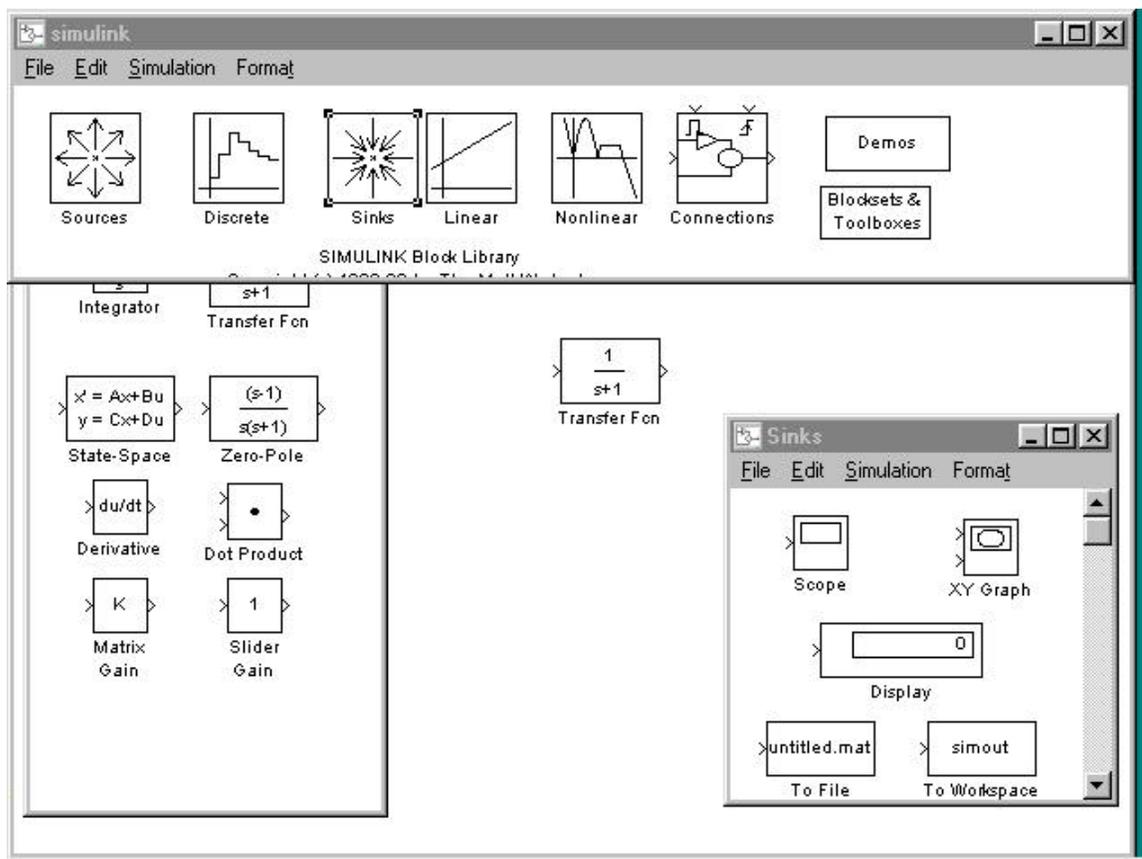
is defined in Matlab as `A=[1 2; 3 4]`

In the **Control Systems Toolbox** polynomials are defined as vectors. For example, $n(s)=s^4+2s^3+6s^2+7s+1$ is defined in Matlab as `n= [1 2 6 7 1]`.

Some commonly used m-file for control are **rlocus**, **nyquist** and **bode**. Check out the use of these commands by using **HELP**.

2.2 Use of Simulink

To run **Simulink**, type “Simulink” at the command prompt of the **Matlab Command** window.



2.2.1 Tutorial

In this section, we will learn how to use Simulink by going through an example.

To create a new simulation file, go to the **SIMULINK** window. Go to the **File** menu and select **New**. A window with the title, **Untitled**, appears. Use the window control menu to switch back to the **SIMULINK** window (you may also use **Ctrl-Esc**) and double click on the **Linear** block.

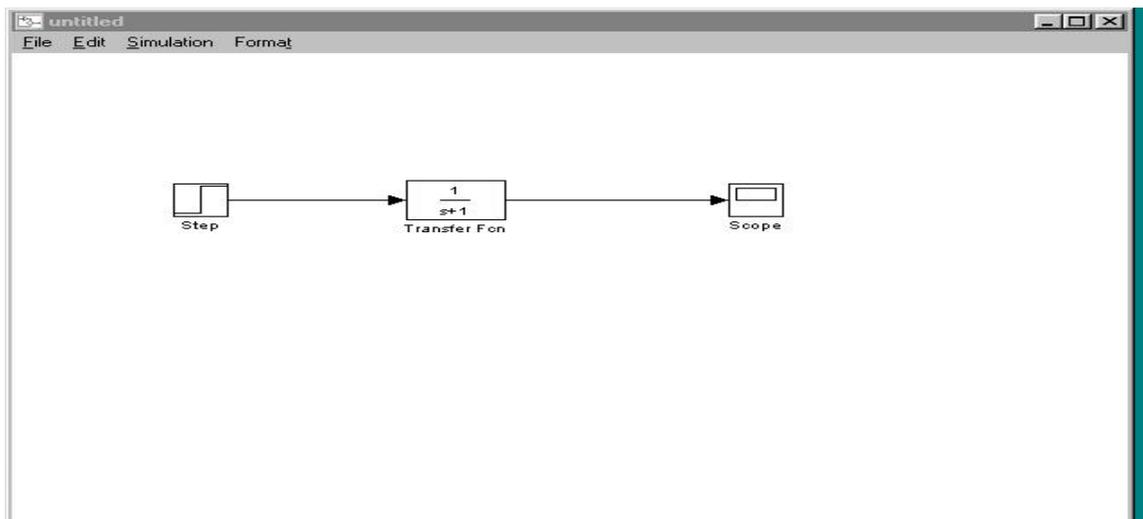
Put the **Untitled** window beside the **Linear** window as shown overleaf.

To add blocks into the new file, just click any of the block in the **Linear** window and drag it into the **Untitled** window. Place a **Transfer Fcn** block in the new file.

Go to the **Simulink** window and double-click on the **Sources** block. Add a step function into the new file. Double-click the **Sinks** block. From the **Sinks** window, add a scope to the new file.

We now connect the blocks. Go to the right hand side of the step function block. Click the mouse and drag the mouse to the left hand side of the **Transfer fcn** block. A connection is now made. Similarly, connect the scope to the **Transfer fcn** block. **Untitled** should look as shown below.

Double-click the scope to see the display. To change the scale of the display, modify the **Horizontal** or **Vertical** range.



Double-click any blocks in **Untitled** to change the parameters of the block.

Help on any particular block can be obtained by clicking the **Help** button available in the parameter window of the blocks.

Run a simulation by choosing **Start** from the **Simulation** menu. To stop simulation, click **Stop** from the **Simulation** menu. Use **Restart** to return your simulation. To specify the start and stop time of the simulation, click **Parameters** from the **Simulation** menu and change the parameters in the window.

2.2.2 Using Matlab with Simulink

Note that variables defined in Matlab could be used. For example if the variable **num** is defined in Matlab, the numerator parameter in **Transfer Fcn** could be entered as **num**.

Simulation result connected to **Scope** could not be stored. To save the simulation results, you have to pass it back to Matlab by using the **To Workspace** block in the **Sinks** window.

2.3 Design and analysis of a continuous time system

A simplified DC servo dynamics can be modeled as a second order system

$$P(s) = \frac{10}{s(s+1)}$$

The control system's performance could be analyzed using transient response, frequency response, etc.

Step 1: Enter a description of the open loop system in Simulink. Add in the appropriate source and sink to see the open loop unit step response. Comment on the response.

Step 2: Close the loop with a PD control

$$C(s) = k_p(1+T_d s)$$

under unity negative feedback. Choose the P control gain $k_p=10$ and the differential time constant $T_d=0.02, 0.2, 2$ respectively. Modify the description in *Step 1* to see the unit step response of the closed loop system. Comment on the response.

3. Design and analysis of a discrete time system

In this part of the experiment, a digital control system is simulated and the performance of a digital PD and a phase compensator will be evaluated.

An important class of digital control algorithm is obtained by direct discretization of well-known analog controllers. Several methods of discrete equivalence, such as backward difference and Tustin approximation are available and the common assumption is that the sampling interval T is small. Their performance will thus deteriorate rapidly when T exceeds certain limits.

Step 3: Discretize the simplified servo model

$$P(s) = \frac{10}{s(s+1)}$$

using Zero Order Hold associated with sampling period $T=0.05$ sec

Use Matlab's "C2DM" command to discretize the plant. Simulate the unit step responses and compare with that of *Step 1*.

Use Matlab's "MARGIN" command to obtain the Bode plot of the discretized plant and find the gain margin (k_g) and phase margin ϕ from the gain and phase curves.

Step 4: Discretize the PD control given in *Step 2* using the backward rule to get the discrete equivalent ($T=0.05$ sec). Simulate the closed loop step response, compare them with the continuous cases and use Bode plots to explain your observations.

Step 5: Discretize the PD control given in *Step 2* using the Tustin rule to get the discrete equivalent ($T=0.05$ sec). Simulate the closed loop step response, compare them with that of *Step 4*.

Step 6: In order to conduct phase compensation, apply W-transformation to achieve the fictitious continuous time transfer function. (You may use the formula provided by the Reference 2 on page 794)

Choose an appropriate compensator

$$C(w) = \frac{1 + w/a}{1 + w/b}$$

to improve the gain and phase margins if necessary. A proper gain margin is $k_g \geq 10$ dB. A proper phase margin is $\phi \geq 50^\circ$. Simulate the step response of the DC servo with the compensator in place. Compare the responses with it *Steps 4 & 5*, and give your conclusions.

4. Reference

1. Simulink User's Guide
2. Franklin, Powell and Workman, "Digital Control of Dynamic Systems", Addison Wesley, 1990.
3. Lecture Notes for EE4301 Digital Control Systems , 1999.