UNIVERSITY OF CALIFORNIA AT BERKELEY
Department of Mechanical Engineering
ME134 Automatic Control Systems
Spring 2002

Report Due: Tuesday, February 26 One report per group is required.

Control Systems Simulation Using Matlab and Simulink

1 Introduction

In ME134, we will make extensive use of Matlab and Simulink in order to design, analyze and simulate the response of control systems.

2 Control of Second Order System

We will simulate the open loop and closed loop step response of the dynamic system described by the state and output equations

$$\begin{aligned} \frac{d}{dt}x_1 &= -.1\,x_1 + .1\,x_2 \qquad (1) \\ \frac{d}{dt}x_2 &= -.2\,x_2 + .1\,u \\ y &= x_1 \qquad (2) \end{aligned}$$

and transfer function

$$\begin{aligned} G(s) &= \frac{.01}{s^2 + .3s + .02} \qquad (3) \\ Y(s) &= G(s)U(s) \end{aligned}$$

Here u is the input, y is the output, and x_1 and x_2 are the two states of the system.

The two tank fluid system shown in Fig. 1 can be modeled by the above state and output equations and/or transfer function.

2.1 Open loop unit-step response

Consider the open loop unit-step input response of this system. The unit-step input is given by $u(t) = \mu(t)$, were

$$\mu(t) := \begin{cases} 0 \text{ if } t < 0 \\ 1 \text{ if } t \geq 0 \end{cases}$$

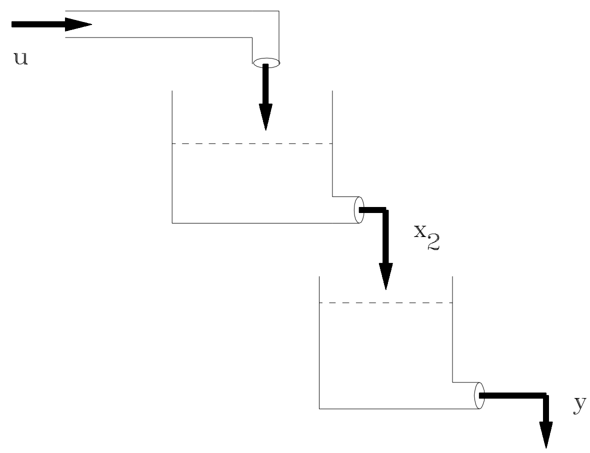

Figure 1: fluid system

Load the simulink file *tank_open.m*. Using simulink, modify the system to the obtain the open loop unit-step input response of this system. Plot the open loop response on a plot.

2.2 Continuous Time (C.T.) closed loop unit-step response

Consider now the closed loop unit-step input response of this system. The control system is described by the block diagram in Fig. 2 where the controller is a PID type controller given by the

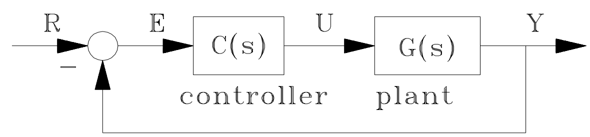

Figure 2: feedback control system

transfer function

$$\begin{aligned} C(s) &= K_p + \frac{K_i}{s} + K_d s \\ U(s) &= C(s)E(s)\,. \end{aligned}$$

In the time domain the PID control action can be described by

$$\begin{aligned} u_p &= K_p e\,, \ \frac{d}{dt}u_i = K_i e\,, \ u_d = K_d \frac{d}{dt} e \\ u &= u_p + u_i + u_d \end{aligned}$$

where $e = r - y$ and the reference input is a unit-step $r(t) = \mu(t)$. (Notice that pure D action is unrealizable and must be approximated by numerical differentiation.)

Using simulink, modify the system in the file *tank_continuous.m* so that the continuous time (C.T.) PID control block is connected in the feedback loop. Run simulations of the closed loop unit-step input response of this system for different combinations of the PID gains. Try first P action only (i.e. $K_i = 0$) and observe how the response of the closed loop system varies when K_p is increased. Subsequently analyze the effect of introducing the I and D actions in the feedback control system. Try at least the following cases:

	K_p	K_i	K_d
controller 1	1	0	0
controller 2	10	0	0
controller 3	20	0	0
controller 4	20	1	0
controller 5	20	1	20

Plot all the unit-step output (y(t) vs. t) responses of the system in one plot. Indicate which response corresponds to which feedback gain selection. Comment on your results and on the effect that each feedback action has on the response of the control system.

Plot all the unit-step control input (u(t) vs. t) responses of the system in one plot. Indicate which response corresponds to which feedback gain selection. Comment on your results and on the effect that each feedback action has on the control input, u. What do you think would occur if the input u, saturates?

2.3 Discrete Time (D.T.) closed loop unit-step response

Consider now the closed loop unit-step input response of this system under a discrete time PID controller. The discrete time PID controller is given by

$$\begin{aligned} u_p(k) &= K_p e(k) \\ u_i(k) &= u_i(k-1) + K_i e(k) \\ u_d(k) &= K_d[e(k) - e(k-1)] \\ u(k) &= u_p(k) + u_i(k) + u_d(k) \end{aligned}$$

where $e(k) = r(k) - y(k)$ and the reference input $r(k)$ is a unit-step. Notice that the D.T. PID control gains should not be chosen to be numerically equal to the corresponding gains of the C.T. controller. Factors in the the I and D gains should be included to account respectively for numerical integration and differentiation.

Modify the simulink file *tank_discrete.m* so that the discrete time PID blocks are in the feedback path. Set the controller sampling time $T = 2.5$ sec. Use the procedure describe Section 6. The following performance specifications should be satisfied:

settling time:	$\approx$ 30 sec
overshoot:	$\approx$ 20% (for a unit step response)
steady state error:	$\approx$ 0.01

Compare the response of the D.T. and C.T. PID controllers for similar conditions. Comment on the effect of the length of the sampling time on the response of the discrete time feedback system.

3 Two-mass vibratory system

Consider the two-mass vibratory system shown in Fig. 3 below.

This system is similar to the experimental setup which will be used in subsequent ME134 laboratories (although the values of the masses and spring constant are vastly different). The system in Fig. 3 also represents a model for a computer disk file actuator.

The state of the system are defined as: x_1 is the position of m_1, (relative to an inertial frame); x_2 is the velocity of m_1; x_3 is the position of m_2, (relative to an inertial frame); and x_4 is the velocity of m_2.

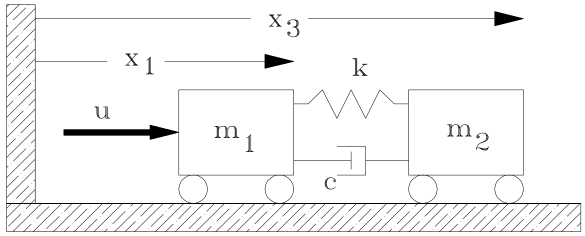

Figure 3: spring-mass system

The reference trajectory is denoted by $\mathtt{ref}(t)$. The objective of the control system is to make the position of m_2, which is x_3, follow the reference trajectory as close as possible. Hence, we could define an error at each time t by

$$\mathtt{error}(t) := \mathtt{ref}(t) - x_3(t) \tag{4}$$

In a disk-drive system (which we used to motivate this example) the reference trajectory would be a staircase-like signal, and x_3 would be the position of the read/write head. The read/write head must be moved to a particular track, and held there for a short time to either read or write, and then moved to a different track. The head must be very still before the read/write process can take place. Hence, the response of the head position due to a step-change in desired position is important.

In this investigation, we will simply use a unit-step reference trajectory, $\mathtt{ref}(t) = \mu(t)$, where

$$\mu(t) := \begin{cases} 0 \text{ if } t < 0 \\ 1 \text{ if } t \geq 0 \end{cases}$$

In this simulation example, rather than using the error in Eq. (4), we will first assume that only the position of the first mass, m_1, is measured and the error signal used by the controller is

$$e(t) := \mathtt{ref}(t) - x_1(t) \,. \tag{5}$$

Notice that the control input u gets applied to m_1.

The controller has 1 state, and its dynamics are governed by

$$\begin{aligned} \dot{x}_c &= -a_c x_c + b_c \left(\mathtt{ref} - x_1\right) \\ u &= c_c x_c + d_c \left(\mathtt{ref} - x_1\right) \end{aligned}$$

where the gains, a_c, b_c, c_c and d_c are chosen to achieve the tracking objective.

The parameters of the two mass vibratory system and the state equations for this system are:

$$\begin{aligned} m_1 &= 1 \ \mathtt{kg} \\ m_2 &= 0.1 \ \mathtt{kg} \\ k &= 5 \ \mathtt{newtons/meter} \\ c &= 0.1 \ \mathtt{newtons/meter/sec} \end{aligned}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m_1} & -\frac{c}{m_1} & \frac{k}{m_1} & \frac{c}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k}{m_2} & \frac{c}{m_2} & -\frac{k}{m_2} & -\frac{c}{m_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m_1} \\ 0 \\ 0 \end{bmatrix} u$$

1. The file twomass.m in Simulink contains the model of the mechanical system described above and the dynamic controller. This file is not complete and has some errors. You are asked first to check the dimensions of the system matrices and insert some of the missing elements in the file. Secondly, you are asked to test the response of the feedback control system due to a unit-step reference input of 1 meter for each of the following 5 control gain selections:

	a_c	b_c	c_c	d_c
controller 1	5	-4	1	1
controller 2	5	-4	4	4
controller 3	5	-4	8	8
controller 4	5	-4	16	16
controller 5	5	-4	20	20

   Notice that, for the above control gain selections, the controller transfer function can be written as follows

   $$C(s) = K\frac{s+b_1}{s+a_1}\,, \quad U(s) = C(s)\;E(s)$$

   where $K = c_c = d_c$, $a_1 = a_c$ and $b_1 = a_c + b_c$.

   Be sure to save as output the position of each mass, as well as the control force (ie., the output of controller) used to cause the motion.

2. Compare the performance of each controller. Keep in mind that the motor which is actually providing the force on MASS 1 is probably limited in the total amount of force it can generate, as well as the rate at which it can develop force.

3. Try a constant-gain controller (no dynamics) of the form

   $$u(t) = K\,(\texttt{ref}(t) - x_1(t))$$

   (you get to pick the value of K). Are you able to achieve good performance with such a simple controller?

4. Modify the control structure so that the error which is fed to the controller is

   $$e = \texttt{ref} - x_3\,.$$

   Test the performance of the 5 controllers described in 1. above when the position of the second mass x_3 is measured instead of the position of the first mass x_1. Comment on the results obtained.

4 Ball-and-Beam

Consider the Ball-and-Beam system shown in Fig. 4:

A ball of mass m slides on a beam which has moment of inertia J about its center of mass. The control torque u is applied to the beam at its center of mass. The equations of motion for this system are:

$$m\ddot{r} - mr\dot{\theta}^2 + mg\sin(\theta) + b\,\dot{r} = 0$$
$$J_T\ddot{\theta} + 2mr\dot{r}\dot{\theta} + mgr\cos(\theta) - u = 0\,,$$

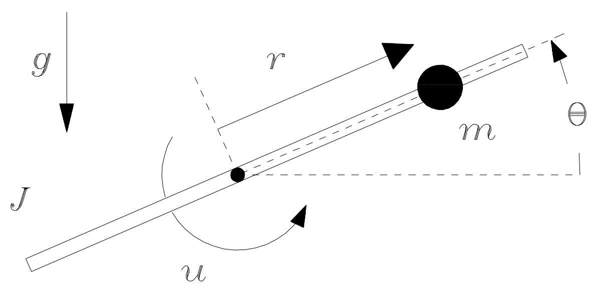

Figure 4: ball-and-beam

where $J_T = J + mr^2$ and b is the coefficient of viscous friction between the ball and the beam.

The state vector for this system is

$$x = \begin{bmatrix} r \; \dot{r} \; \theta \; \dot{\theta} \end{bmatrix}^T .$$

The objective of the control system is to bring the state to $x = 0$.

1. Load the file *ball_n_beam.m* using simulink. This file contains a simulink model of the beam-and-ball system in the block labeled "ball" and a **linear state-feedback controller**. This controller is of the form

$$\begin{aligned} u(t) &= k_r r(t) + k_{\dot{r}} \dot{r}(t) + k_\theta \theta(t) + k_{\dot{\theta}} \dot{\theta}(t) \\ &= k_r x_1(t) + k_{\dot{r}} x_2(t) + k_\theta x_3(t) + k_{\dot{\theta}} x_4(t) \end{aligned}$$

   where the constant gains $k_r, k_{\dot{r}}, k_\theta$ and $k_{\dot{\theta}}$ are real numbers chosen by the control system designer to achieve the objective, which in this case is to regulate the ball/beam system at the point

$$\begin{bmatrix} r \\ \dot{r} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

   Simulate the response of the feedback system for the following initial conditions:

$$x(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{bmatrix},$$

   In words, what do each of these initial conditions represent?

2. Find initial conditions of the form

$$x(0) = \begin{bmatrix} r_o \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad , x(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{\theta}_o \end{bmatrix},$$

   for which the controlled system is unstable (ie., when started from this initial condition, the system does not restore all of the states to 0). For each case try to find both a stable initial condition and an unstable initial condition that are within 10 to 20 percent of each other.

3. Find the constant input torque $\bar{T}$, such that the state $\bar{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ is an equilibrium point.

5 Report: (One report per group)

Write a small report describing your findings when performing this lab.

5.1 Recommended Form for Lab Reports:

Abstract- This is a short (1/2 to 1 page) synopsis of the intent and results of the exercise. Often the most difficult part of the report to do correctly, it may be prudent to write it last.

Introduction- In this section the objective should be stated and backed up with details. Assumptions to be made should be listed and explained. The boundaries of the experiment, (i.e., what the experiment will encompass) should be clearly defined. The apparatus and/or model should also be described.

Theory- Any relevant theory can be included here. Proofs are recommended, again, if relevant.

Discussion- This is the main body of the report. In this section should be discussed the questions asked in the laboratory handouts.

Results- Any pertinent results should be included here. Tabular forms of presenting results are encouraged. Plots or graphs should be placed in an organized manner at the end of this section.

Calculations- As implied by the heading, relevant calculations are included here. In cases of multiple similar calcs., only one example need be presented.

Conclusion- The general conclusions for the experiment should be summarized here. Remember, hard numbers carry weight, e.g. "The overshoot was 30specification", is better than "The overshoot was unsatisfactory".

Note: While I realize than many will have other formats for the the writing of lab reports, it will be appreciated if students attempt to follow the above plan.

5.2 Helpful Hints

- The report should be comprehensive, so that anyone could read and understand it without having the lab handout in front of them. For example, in the first lab instead of writing "In part B of this lab we did...", it should be "After applying a unit step reference input, it was found that...". In other words, the report should stand alone. Try to write the report as if the reader knows the material, but doesn't know the specific details of your project.

- Figure titles should be as descriptive as is reasonably possible so the figures can stand alone. For example, the title "Digital PID Control of the Hubble Telescope" (with accompanying parameter values) is better than "PID Control" or "Output vs. Time". Be sure to label the axes and include units.

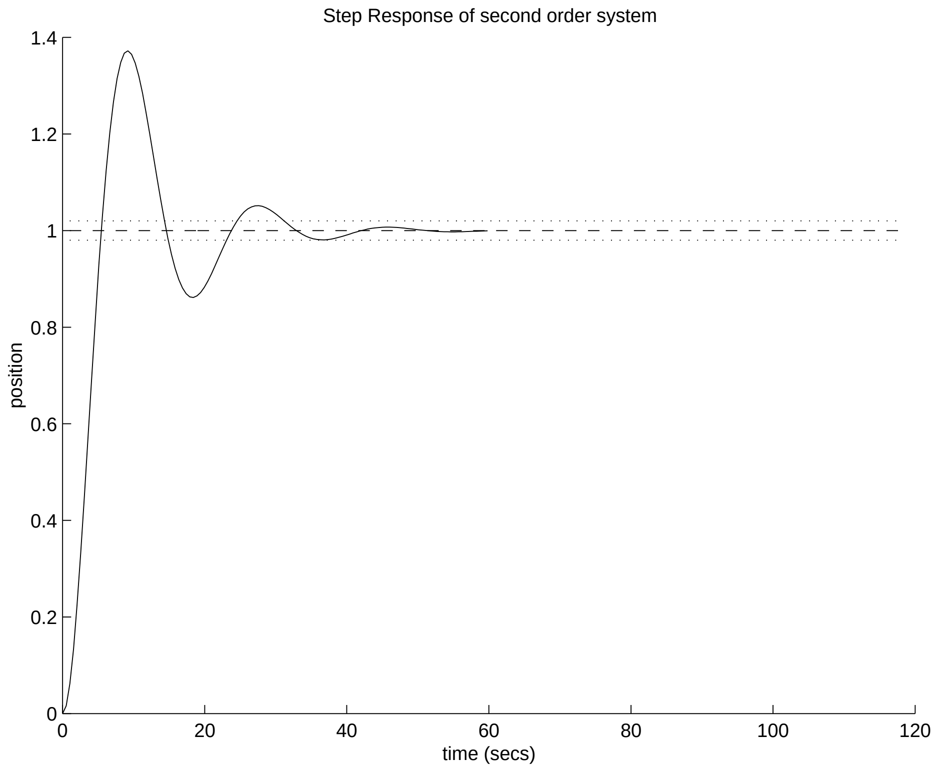

Figure 5: Step Response of second order system

6 "Hands-on" Design of Control Systems

"Hands on Design systems" refers to directly working with the system to be controlled and trying a variety of controllers and control parameters.

6.1 Performance Specification

One of the first things that must be done during hands-on design is deciding upon a criterion for measuring how *good* a response is. For example, when we deal with systems where we are not bothered with the actual dynamics of how the steady state is reached, but only care about the steady state itself, a good measure will be the *steady state error* of the system defined by

$$e = x_{final} - x_{ref} \tag{6}$$

However, in dynamic systems where the *transient* behaviour is also important, it becomes important to introduce several other criterion. The most common are:

Settling Time : This is a measure of how long it takes for the system to stabilize in its new final value and is usually defined by the time it takes for the system response to come within a specified tolerance band of the setpoint value and stay there.

Overshoot : This is the maximum distance beyond the final value that the response reaches. It is usually expressed as a percentage of the change from the original value to the final, steady state value.

Steady State Error : This is a measure of how far the final value reached in the step response is from the actual desired value.

6.2 Feedback Control

In feedback control, we are interested in designing a *Control Law* which operates on the present error of the system ($x_{desired} - x_{actual}$) and provides an actuation (u) which will act upon the system and bring it nearer to the desired value.

We are interested in three types of controllers:

6.2.1 Proportional Control (P-control):

This is the simplest type of continuous control law. The controller output is made proportional to the error. The proportionality constant is called the *gain*.

$$u_p = K_p e \tag{7}$$

where u_p is the output and K_p is the gain.

One important thing to be noted about Proportional Control is that is incapable of maintaining the output steady state value at the desired value. This is clear from the equation above. We can see that as long as a non-zero actuation is required to maintain the system at the desired value, the error cannot be zero. Mathematically, $u_p \neq 0$, therefore $e = u_p/K_p \neq 0$. As we increase the value of the gain, we can see the steady state error will decrease. However, K_p is limited by the dynamics of the system. Therefore the value of K_p will have to be arrived at by compromising between the steady state error and the dynamic stability of the system.

6.2.2 Integral Control (I-control):

If we want a zero steady state error, we want a control mode that is a function of the history of error accumulation. The longer the error persists, the stronger the control action should be to cancel it. The mathematical operation of integration is a means of implementing this action.
Mathematically,

$$u_i = K_i \int_0^t e(\tau) d\tau \tag{8}$$

It can be seen that the integral action is capable of reducing the steady state to zero. This is because even though the steady state error is reduced to zero, the integral controller is still capable of maintaining some actuation (i.e $u_i \neq 0$) because of the past history of error values.

Integral control is usually combined with Proportional Control to give a PI controller.

$$u_{pi} = K_p e + K_i \int_0^t e(\tau) d\tau \tag{9}$$

The PI controller has two *tuning parameters*, namely K_p and K_i. The easiest way to tune the control system is to start with the integrator turned off, i.e $K_i = 0$. and find a reasonable proportional gain, as described in P-control. The integral gain (K_i) can then be slowly increased until the steady state error is brought to zero in a reasonable amount of time.

It should be noted that the Integral action has a de-stabilizing influence on the system and therefore it may be necessary to reduce the proportional gain somewhat if the integral action causes too much oscillation. Ideally, we may have a mechanism whereby the integral action does not start till the system has actually settled down a little. (say come within a 5% error band of the desired value). However, for our case, we will simply set the integral gain to a small value as compared to K_p, to minimize oscillation.

6.2.3 Derivative Control (D-control)

The use of integral action is sufficient to reduce the steady state error to zero. However, the dynamic or the transient response may still be poor because of large oscillations, overshoots etc. Derivative control can be used in such cases to stabilize the dynamic behaviour of the system. Mathematically the derivative control law can be written as

$$u_d = K_d \frac{de}{dt} \tag{10}$$

Thus the derivative action can be used to create *damping* in a dynamic system and thus stabilize its behaviour. It must however be noted that derivative action slows down the initial response to the system.

Derivative action is usually used with proportional and integral control to achieve what is called **PID control**. Mathematically,

$$u_{pid} = K_p e + K_i \int_0^t e(\tau)d\tau + K_d \frac{de}{dt} \tag{11}$$

We now have three independent parameters (K_p, K_i and K_d) to play around with and optimizing them may present a formidable problem. However, a reasonable performance can be obtained by following the procedure described below:

- Start with the P and D actions only. That is, initially set the integral gain to zero. This means that initially we are concentrating on making the system exhibit a sufficiently good dynamic response. This is done by first increasing K_p until unacceptable dynamic behaviour is obtained. Then we increase K_d and see if the behaviour improves. If it does, we again increase K_p. We keep iterating like this till a reasonable response time and steady state has been obtained.

- After this, we start increasing K_i to decrease the steady state error. If we notice that even after K_i has increased enough to cause too much oscillations, the steady state error has not decreased enough, then we may need to go back to the first step and alternately increase and decrease P and D gains.

6.3 Sampling Time with PID controllers

We can state intuitively that the sampling time has to be much smaller than the open loop settling time for discrete control to be effective. This is because, between two successive applications of the actuation, we are leaving the system to behave without interference. In other words, the response between two successive applications of control actuation is essentially open-loop behaviour. In order to choose a reasonable sampling time, we start with a sampling time which is quite small (say, $1/20^{th}$ of the open loop settling time). Then we gradually increase the sampling time and stop when the dynamic behaviour becomes unacceptable.