ELEN E3084: Signals and Systems Lab
Lab V: Feedback Control System

1 Introduction

In this lab, we'll analyze and design an automatic position control system described in Section 6.7: Application to Feedback and Controls of the Lathi course textbook. The system is used to control the angular position of a heavy object such as a tracking antenna or an anti-aircraft gun mount. Figure 1 represents the system. The input is the desired angular position of the object Θ_i, which can be set at any given value. The output is the actual angular position of the object Θ_o, which is measured by a potentiometer whose wiper is mounted on the output shaft.

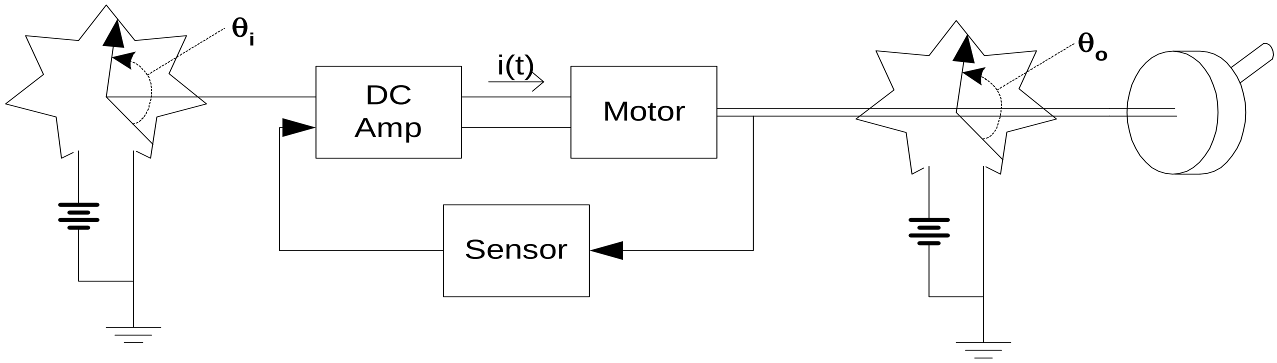

Figure 1: An automatic position control system.

The block diagram of the control system is shown in Figure 2, where $A(s)$ represents the DC amplifier, $G(s)$ represents the motor, and $H(s)$ represents the sensor. The transfer function relating the output to the input of the closed-loop system is:

$$T(s) = \frac{\Theta_o(s)}{\Theta_i(s)} = \frac{A(s)G(s)}{1 + A(s)G(s)H(s)}$$

From this equation, we shall investigate the behavior of the automatic position control system.

Let's first go to the `lab5` directory and start the `diary` function. This creates a diary file that must be submitted to the LA or TA at the end of the lab session. Don't forget to turn off the function at the end of the session.

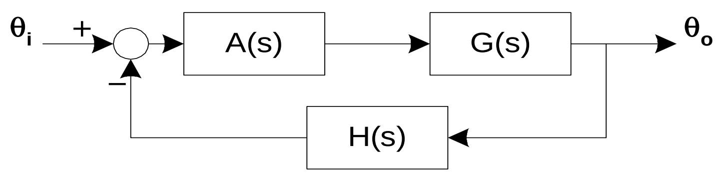

Figure 2: A block diagram representation of the system shown in Figure 1.

2 Mathematical Model

The first step in analyzing a control system is to derive a mathematical model of the system. In our case $A(s) = K$ and $H(s) = 1$. Now, to derive a mathematical model of the armature-controlled DC motor, $G(s)$, we can use a simple example depicted in Figure 3.

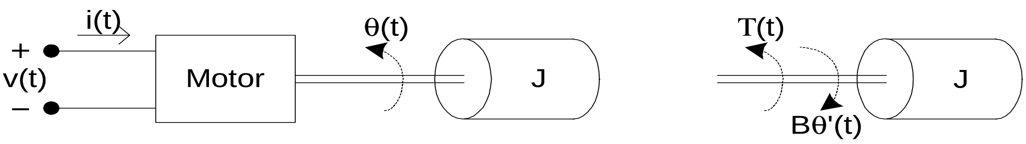

Figure 3: An armature-controlled DC motor. (a) A simple example of an armature-controlled DC motor. (b) A free diagram of a mechanical load with moment of inertia J.

Here, the armature-controlled DC motor is driven by the current $i(t)$, and $\Theta(t)$ is the angular position of the rotor. $T(t)$ is the torque generated in the rotor and it is proportional to the current $i(t)$: $T(t) = K_T i(t)$, where K_T is a constant of the motor. This torque drives a mechanical load whose free body diagram is illustrated in Figure 3(b). The viscous damping (with coefficient B), which is proportional to the angular velocity $\Theta'(t)$, dissipates a torque $B\Theta'(t)$. If J is the moment of inertia of the load, then we get the following:

$$J\Theta''(t) = T(t) - B\Theta'(t) \Rightarrow J\Theta''(t) + B\Theta'(t) = K_T i(t)$$

Taking Laplace transform of the above equation (assuming zero initial conditions), we get:

$$Js^2\Theta(s) + Bs\Theta(s) = K_T I(s) \Rightarrow \frac{\Theta(s)}{I(s)} = G(s) = \frac{K_T/J}{s^2 + (B/J)s}$$

For the rest of this lab, we let $K_T/J = 1$ and $B/J = 8$.

To Do 1

A MATLAB function `printtransfn` that prints the transfer function of the feedback system shown in Figure 2, using the mathematical model given in this section, is provided. The function takes as an argument the value of the DC amplifier gain K. Using the given function, find the transfer function for $K = 4$, 16, and 80.

```
> printtransfn(4);
> printtransfn(16);
> printtransfn(80);
```

---

3 Analysis

To analyze the control system, we'll use two test signals: the step and the ramp functions. The step input is used to test the system when the angular position of the object is instantaneously changed. This input will be one of the most difficult ones to follow; if the system can perform well for this input, it is likely to give a good account of itself under most other expected situations. The ramp function is used for the following case: if the anti-aircraft gun is tracking an enemy plane moving with a uniform velocity, the gun-position angle must increase linearly with t.

To Do 2

Find the unit step response of the system for $K = 4$, 16, and 80.

```
> unitstep(4);
> % Repeat the above command for other values of K.
```

Which value of K yields the best result? Why?
Does the output ever reach the desired value?

Find the unit ramp response of the system when $K = 4$, 16, and 80.

```
> unitramp(4);
> % Repeat the above command for other values of K.
```

Which value of K yields the best result? Why?
Does the output ever reach the desired value?

---

4 Design

As seen in the previous section, for unit step input, the system response generally takes one of the two shapes as shown in Figure 4. The response shown in Figure 4(b) is faster than the one shown in Figure 4(a), but unfortunately the improvement is achieved at the cost of ringing (oscillations) with high overshoot. The rise time t_r is defined as the time required for the response to rise from 10% to 90% of its steady-state value, and it indicates the speed of the response. We also define the settling time t_s to be the time required for the response to reach and stay within 2% of the final value. Note that a good system has a small overshoot, a small value of t_r and t_s.

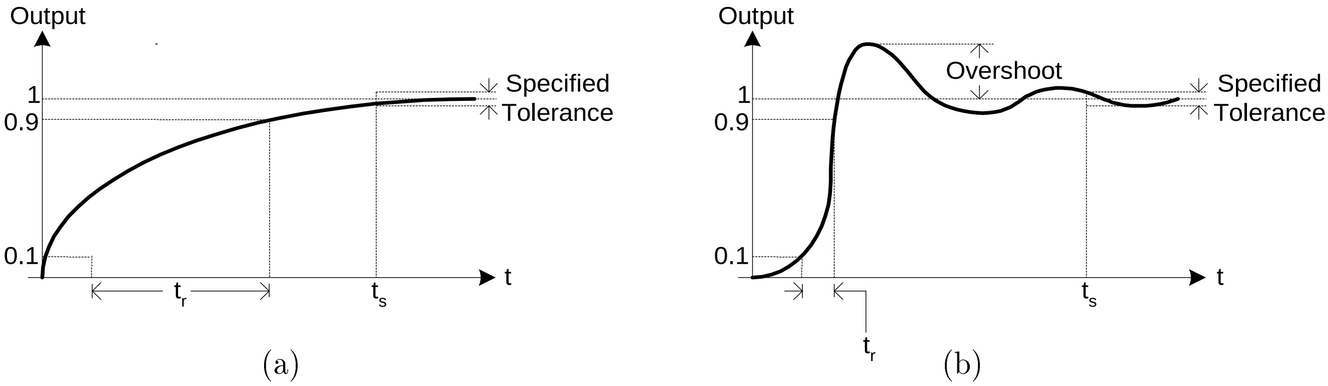

Figure 4: The unit step response. (a) Overdamped or critically damped. (b) Underdamped.

For unit ramp input, we see that there is a steady-state error. The error should be small for it to be tolerable.

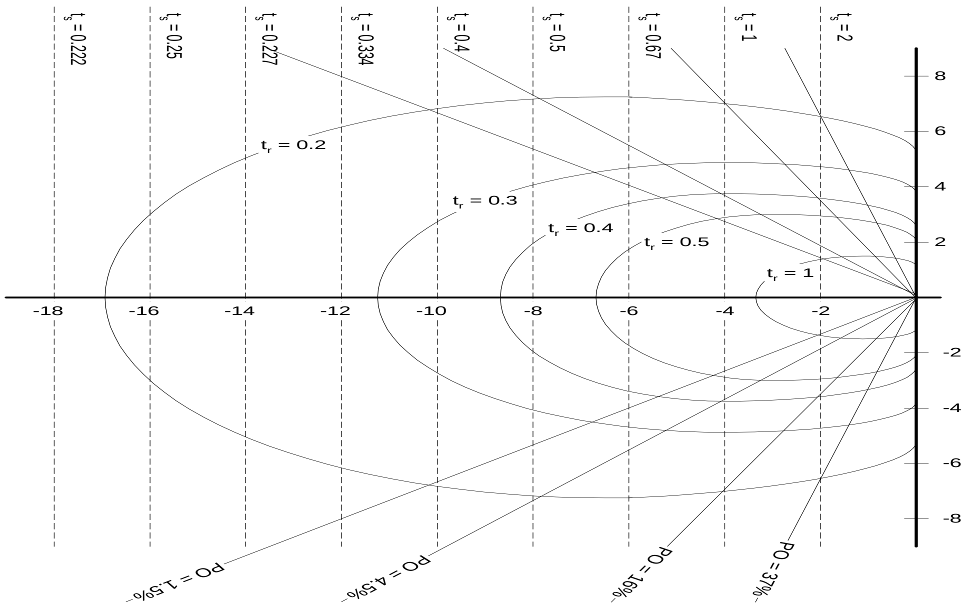

Figure 5: Contours of second-order system pole location for constant PO, constant t_r, and constant t_s in the s plane.

As described in Section 6.7-2 of the textbook, the basic characteristic of the transient response of a closed-loop system is closely related to the location of the closed-loop poles. For a second-order system with no zeroes, Figure 5 displays the contours of the system pole location for constant percent overshoot (PO), constant t_r, and constant t_s in the s plane. For example, for the system to meet the following specifications: PO $\leq$ 16%, $t_r \leq 0.5$s, and $t_s \leq 2$s, the closed-loop transfer function $T(s)$ must have its poles lie in the shaded region shown in Figure 6.

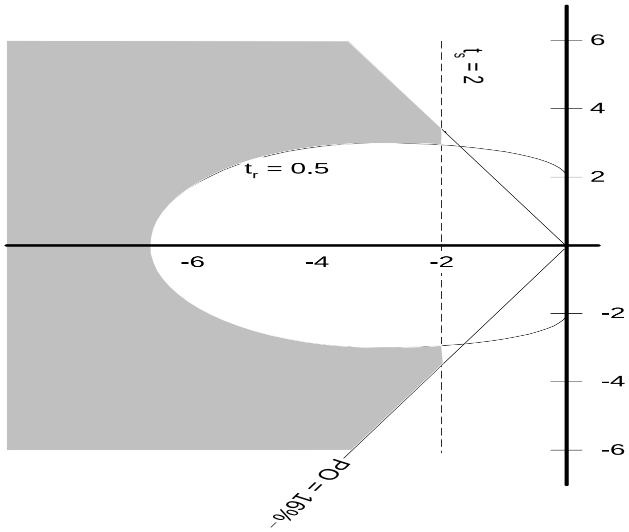

Figure 6: Poles satisfying the given specification: PO $\leq$ 16%, $t_r \leq 0.5$s, and $t_s \leq 2$s.

Root-Locus Analysis

If the system has a variable loop gain K, then the location of the closed-loop poles depend on the value of K chosen. Therefore, it is important that the designer know how the closed-loop poles move in the s plane as the loop gain is varied. The root-locus method allows us to plot the roots of the characteristic equation of the closed-loop transfer function for all values of the system parameter (in our case, K). Note that, once the desired value of K is known, the roots corresponding to the value can be located on the resulting graph. The students are highly recommended to read Section 6.7-3 of the textbook to understand the steps involved in sketching the root-locus.

To Do 3

Derive the range of K that satisfies the transient specifications: PO $\leq$ 16%, $t_r \leq 0.5$s, and $t_s \leq 2$s.

First, obtain the root locus for the system:

```
> rootlocus;
```

Using the graph displayed by MATLAB and the graph shown in Figure 5, find the point in the s plain at which the root locus enters the desired region – p_e, and the point at which the root locus leaves the desired region – p_l. Then, the values of K that yields closed-loop poles at p_e and p_l specify the range of values of K for which the specifications are satisfied. Given the poles at p_e and p_l, the corresponding values of K can be obtained from the equation:

$$A(s)G(s)H(s) = -1 \Rightarrow s^2 + 8s + K = 0 \Rightarrow s_{1,2} = -4 \pm \sqrt{16 - K}$$

You can also obtain a value of K by clicking on a point of intreset in the graph displayed by the `rootlocus` command.

Using the `unitstep` function, obtain the unit step response for the two values of K obtained above. Does the response satisfy the given specifications?

Note that the transient specifications are generally specified for the step input because the step input represents a sudden jump discontinuity: if a system has an acceptable transient response for the step input, it is likely to have an acceptable transient response for most of the practical inputs.

Another important design criterion is the steady-state error to certain expected inputs such as step or ramp. The error $e(t)$ is the difference between the desired output Θ_i and the actual output Θ_o: $e(t) = \Theta_i(t) - \Theta_o(t)$. Thus the steady-state error e_{ss} is the value of $e(t)$ as $t \to \infty$. It's shown in Section 6.7-4 of the text that, for our system, the steady-state error for the unit step input (e_s) is 0 and the steady-state error for the unit ramp input (e_r) is $8/K$.

To Do 4

Design an automatic position control system that meets the following specifications: PO $\leq$ 16%, $t_r \leq 0.5$s, $t_s \leq 2$s, and $e_s = 0$, $e_r \leq 0.15$. What is the value of K that you chose? Draw the unit step response and unit ramp response of the system.

---

5 Compensation: Lead Compensator

Now, consider the system with the following specifications: PO $\leq$ 16%, $t_r \leq 0.5$s, $t_s \leq 2$s, $e_s = 0$, and $e_r \leq 0.1$. To meet the steady-state specification, we must have $8/K \leq 0.1 \Rightarrow K \geq 80$. But, this is outside the acceptable range of the values of K that satisfies the transient specifications. Since we cannot meet both the transient and steady-state specifications, we must add some kind of compensation, which will modify the root locus to meet all the specifications: the root locus can be shifted to the left so that the poles corresponding to $K = 80$ are forced to be in the region that satisfies the transient specifications.

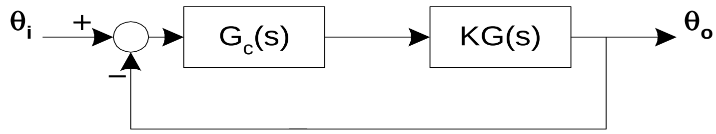

Figure 7: The block diagram of an automatic position control system with a lead compensator.

As shown in Figure 7, a compensator $G_c(s)$ can be placed in series with $KG(s)$. If $G_c(s)$ has a single pole and a single zero, then choosing the pole farther to the left of the zero would shift the root locus to the left (see the rules for sketching the root locus in Section 6.7-3 of the textbook). Thus, we have:

$$G_c(s) = \frac{s+\alpha}{s+\beta}, \qquad \text{where } \beta > \alpha$$

To Do 5

Draw the root locus of the new system where $\alpha = 8$ and $\beta = 16$.

```
> compensate(8, 16);
```

Can the specifications, PO $\leq$ 16%, $t_r \leq 0.5$s, $t_s \leq 2$s, $e_s = 0$, and $e_r \leq 0.1$, be satisfied with the given compensator? Note that we have chosen $\alpha = 8$ so that there is a pole-zero cancelation and the resulting system is still a second order system. This way, we can still use the graph provided in Figure 5.

---

6 Lab Problems

We are required to meet the following specifications with our system: PO $\leq$ 16%, $t_r \leq 0.2$s, $t_s \leq 0.5$s, $e_s = 0$, and $e_r \leq 0.06$. Is it possible to meet these specifications just by adjusting K? If not, suggest a suitable form of compensator and find the resulting PO, t_r, and t_s.

Hint: *If a compensator is used, modify the* `transfn` *function provided with the lab, incorporating the compensator added to the system. Then use the* `unitstep` *function with the chosen K value to obtain the resulting values for PO, t_r, and t_s.*

Appendix

This appendix lists the MATLAB code of the functions used in the lab.

transfn

```
function [CTnum, CTden] = transfn(K)
% TRANSFN Returns the transfer function of the feedback system described in
% Section 6.7 of the Lathi course textbook.
%   [CTnum, CTden] = transfn(K)
%
%   K: DC amplifier gain
%   CTnum: numerator of the transfer function
%   CTden: denominator of the transfer function

Anum=[0 0 K]; % numerator of A(s)
Aden=[0 0 1]; % denominator of A(s)
Gnum=[0 0 1]; % numerator of G(s)
Gden=[1 8 0]; % denominator of G(s)
Hnum=[0 0 1]; % numerator of H(s)
Hden=[0 0 1]; % denominator of H(s)
[FTnum, FTden]=series(Anum, Aden, Gnum, Gden); % feed-forward transfer fn
[CTnum, CTden]=feedback(FTnum, FTden, Hnum, Hden); % closed-loop transfer fn
```

printtransfn

```
function printtransfn(K)
% PRINTTRANSFN prints the transfer function of the feedback system described in Section 6.7 of
% the Lathi course textbook.
%   printtransfn(K)
%
%   K: DC amplifier gain

% Call the transfn function with given K
[CTnum, CTden] = transfn(K);
% Print the transfer fn
printsys(CTnum, CTden);
```

unitstep

```
function unitstep(K)
% UNITSTEP displays the unit step response of the feedback system described in Section 6.7 of
% the Lathi course textbook.
%   unitstep(K)
%
%   K: DC amplifier gain

% Obtain the transfn function
[CTnum, CTden] = transfn(K);
% Step response of the closed-loop system
step(CTnum, CTden);
```

unitramp

```
function unitramp(K)
% UNITRAMP displays the unit ramp response of the feedback system described in Section 6.7 of
% the Lathi course textbook.
%   unitramp(K)
%
%   K: DC amplifier gain

% Obtain the transfn function
[CTnum, CTden] = transfn(K);
```

```
% Note that the unit ramp response of this system is the same as the unit step response of the
% system with the transfer function $T(s)/s$.

% Obtain the numerator of the transfer fn T(s)/s
CTRnum = CTnum;
% Obtain the denominator of $T(s)/s$
CTRden = conv([0 1 0], CTden);
step(CTRnum, CTRden);
title('Ramp Response');
```

rootlocus

```
function rootlocus()
% ROOTLOCUS Displays the root locus plot of the feedback system described in Section 6.7 of
% the Lathi course textbook.
%   rootlocus

% The numerator of the open-loop transfer fn A(s)G(s)H(s)
OTnum = [0 0 1];
% The denominator of the open-loop transfer fn A(s)G(s)H(s)
OTden = [1 8 0];
% Obtain the root-locus plot
rlocus(OTnum, OTden);
% Set the range for the x and y axis of the current plot
axis([-10, 2, -10, 10]);
```

compensate

```
function compensate(alpha, beta)
% COMPENSATE Displays the root locus plot of the feedback system described in Section 6.7 of
% the Lathi course textbook.
%   compensate(alpha, beta)
%
%   alpha: -zero of the compensator
%   beta: -pole of the compensator

% The numerator of the compensator
Cnum = [0 1 alpha];
% The denominator of the compensator
Cden = [0 1 beta];
% The numerator of the open-loop transfer fn A(s)G(s)H(s)
OTnum = [0 0 1];
% The denominator of the open-loop transfer fn A(s)G(s)H(s)
OTden = [1 8 0];
% The new system
[COTnum, COTden] = series(Cnum, Cden, OTnum, OTden);
rlocus(COTnum, COTden);
axis([-40, 2, -10, 10]);
```