1 Introduction

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

![Figure 1: An automatic position control system.](image)

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 load, and \( H(s) \) represents the sensor. For our purposes in this lab the sensor measures the output position perfectly and without delay, therefore \( H(s) = 1 \).

Notice what this system is doing: it takes the target position \( \theta_i(t) \) and subtracts off the current object position \( \theta_o(t) \). The resulting signal is called the error signal. Now depending on the difference we actuate the motor to move the object towards our goal (if the error
signal is zero then the motor stays still). This is exactly the same procedure we do when balancing a broom on our hand.

Because this is a feedback system we see that the overall transfer function relating the output to the input of the system is

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

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.

## 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 \( J \).](image)

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:

\[ J s^2 \Theta(s) + B s \Theta(s) = K_T I(s) \Rightarrow \Theta(s) = G(s) = \frac{K_T/J}{s^2 + (B/J)s}. \]

For the rest of this lab, we let \( K_T/J = 1 \) and \( B/J = 8 \) so

\[ G(s) = \frac{1}{s(s + 8)}. \]
The block diagram of the model we are going to study in Sections 3 and 4 is depicted in Figure 4.

![Block Diagram](image)

Figure 4: Block diagram of the system studied in Sections 3 and 4.

**To Do 1**
A MATLAB function `printtransfn` that prints the transfer function of the feedback system shown in Figure 4, 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 \( 80 \).

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

Note that the poles of the system (the roots of the polynomial in the denominator of the transfer function) change as you change \( K \), therefore effecting the behavior of the system. Next we will see how the choice of \( K \) affects the system.

### 3 Analysis

To analyze the control system, we’ll use two test signals: the unit step signal \( u(t) \) and the unit ramp signal \( tu(t) \). The step input is used to test the system when we desire to change the position of the load to a fixed place as fast as possible. This kind of input is one of the most difficult 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 representative of a scenario where you want to track an object (e.g., a star if the load is a telescope) that is moving with uniform angular velocity. The telescope-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.

In your opinion which value of $K$ is the most desirable? 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 $K$ seems to be the most desirable? 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 5. The response shown in Figure 5(b) is faster than the one shown in Figure 5(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 it for our system is desirable to have either none or a small overshoot, a small value of $t_r$ and $t_s$. We see these are conflicting goals. Experiment with various values of $K$ and see that a small overshoot increases the values of $t_r$ and $t_s$.

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

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

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, which depend on $K$ in our system. Let’s see how the poles “move” depending on the value of $K$. 
Figure 5: The unit step response. (a) Overdamped or critically damped. (b) Underdamped.

**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 (the complex plane) as the loop gain is varied. For our system we see that the transfer function $T(s)$ is given by

$$T(s) = \frac{K}{s^2 + 8s + K}.$$  

The poles of the system are therefore given by $-4 \pm \sqrt{16 - K}$.

The root locus method is a procedure that 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. For the particular system under consideration the root locus can be drawn very easily, since the poles are given by $-4 \pm \sqrt{16 - K}$. 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**

For what are the values of $K$ are the poles of the system real? For what values of $K$ are the poles complex? You can do this analytically, but let’s also use the root locus functionality of the matlab control toolbox.

```
> rootlocus;
```

At first this plot might seem a little strange and hard to understand, but bear in mind that each point in the plot corresponds to the location of a pole for a particular value $K$. You can obtain a value of $K$ by clicking on a point of interest in the graph displayed by the `rootlocus` command.
Figure 6: Contours of second-order system pole location for constant PO, constant $t_r$, and constant $t_s$ in the s plane.
Design of $K$ given some specifications

For a second-order system with no zeroes, Figure 6 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: $PO \leq 16\%$, $t_r \leq 0.5s$, and $t_s \leq 2s$, the closed-loop transfer function $T(s)$ must have its poles lie in the shaded region shown in Figure 7.

![Figure 7: Poles satisfying the given specification: $PO \leq 16\%$, $t_r \leq 0.5s$, and $t_s \leq 2s$.](image)

**To Do 4**

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

Using the graph shown in Figure 6 and the root locus previously plotted we can find the point in the $s$ plane at which the root locus enters the desired region (let’s call it $s_e$), and the point at which the root locus leaves the desired region (call it $s_l$). The values of $K$ that yield closed-loop poles at $s_e$ and $s_l$ specify the range of values of $K$ for which the specifications
are satisfied. You can either analytically find the values of $K$ corresponding to $s_e$ and $s_l$ or obtain $K$ by clicking on a point in the root locus plot.

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 $\Theta_i$ and the actual output $\Theta_o$: $e(t) = \Theta_i(t) - \Theta_o(t)$. The steady state error is therefore given by $\lim_{t \to \infty} e(t)$. 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 5**

Design an automatic position control system that meets the following specifications: $PO \leq 16\%$, $t_r \leq 0.5s$, $t_s \leq 2s$, 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

Suppose now we would like our system to fulfill the following specifications: $PO \leq 16\%$, $t_r \leq 0.5s$, $t_s \leq 2s$, $e_s = 0$, 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 with the current system (no value of $K$ will ever work) we need to modify the system slightly. This can be done by adding some kind of compensation, which will modify the root locus to meet all the specifications.

$$\theta_o \quad G_c(s) \quad KG(s) \quad \theta_i$$

Figure 8: The block diagram of an automatic position control system with a lead compensator.
As shown in Figure 8, 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:

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

**To Do 6**

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

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

Can the specifications, $PO \leq 16\%$, $t_r \leq 0.5s$, $t_s \leq 2s$, $e_s = 0$, 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 (in fact using this compensator we replace the pole in the in the top block of the original system of Figure 4, which was $-8$ by a pole at $-\beta$). This way, we can still use the graph provided in Figure 6.

---

## 6 Lab Problems

We are required to meet the following specifications with our system: $PO \leq 16\%$, $t_r \leq 0.2s$, $t_s \leq 0.5s$, $e_s = 0$, 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` and `unitramp` function with the chosen $K$ value to obtain the resulting values for $PO$, $t_r$, $t_s$, and $e_r$. 

---

9
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 \( \frac{T(s)}{s} \).

Obtain the numerator of the transfer function \( \frac{T(s)}{s} \)
\[ \text{CTRnum} = \text{CTnum}; \]
Obtain the denominator of \( \frac{T(s)}{s} \)
\[ \text{CTRden} = \text{conv}([0 1 0], \text{CTden}); \]
\text{step}(\text{CTRnum}, \text{CTRden});
\text{title('Ramp Response')};

\textbf{rootlocus}

\textit{function rootlocus()}
\textbf{}\% ROOTLOCUS Displays the root locus plot of the feedback system described in Section 6.7 of \% the Lathi course textbook.
\textbf{}\% \hspace{5em} \textbf{rootlocus}
\textbf{}\%
\textbf{}\% The numerator of the open-loop transfer function \( A(s)G(s)H(s) \)
\textbf{}\hspace{5em} \textbf{0Tnum} = [0 0 1];
\textbf{}\%
\textbf{}\% The denominator of the open-loop transfer function \( A(s)G(s)H(s) \)
\textbf{}\hspace{5em} \textbf{0Tden} = [1 8 0];
\textbf{}\%
\textbf{}\% Obtain the root locus plot
\textbf{}\hspace{5em} \textbf{rlocus}(\text{0Tnum}, \text{0Tden});
\textbf{}\%
\textbf{}\% Set the range for the x and y axis of the current plot
\textbf{}\hspace{5em} \textbf{axis([-10, 2, -10, 10])};

\textbf{compensate}

\textit{function compensate(alpha, beta)}
\textbf{}\% COMPENSATE Displays the root locus plot of the feedback system described in Section 6.7 of \% the Lathi course textbook.
\textbf{}\% \hspace{5em} \textbf{compensate(alpha, beta)}
\textbf{}\%
\textbf{}\%
\textbf{}\% \hspace{5em} alpha: -zero of the compensator
\textbf{}\% \hspace{5em} beta: -pole of the compensator
\textbf{}\%
\textbf{}\% The numerator of the compensator
\textbf{}\hspace{5em} \textbf{Cnum} = [0 1 alpha];
\textbf{}\%
\textbf{}\% The denominator of the compensator
\textbf{}\hspace{5em} \textbf{Cden} = [0 1 beta];
\textbf{}\%
\textbf{}\% The numerator of the open-loop transfer function \( A(s)G(s)H(s) \)
\textbf{}\hspace{5em} \textbf{0Tnum} = [0 0 1];
\textbf{}\%
\textbf{}\% The denominator of the open-loop transfer function \( A(s)G(s)H(s) \)
\textbf{}\hspace{5em} \textbf{0Tden} = [1 8 0];
\textbf{}\%
\textbf{}\% The new system
\textbf{}\hspace{5em} [\text{COTnum}, \text{COTden}] = \text{series}(\text{Cnum}, \text{Cden}, \text{0Tnum}, \text{0Tden});
\textbf{}\% The new system
\textbf{}\hspace{5em} \textbf{rlocus}(\text{COTnum}, \text{COTden});
\textbf{}\%
\textbf{}\% Set the range for the x and y axis of the current plot
\textbf{}\hspace{5em} \textbf{axis([-40, 2, -10, 10])};