CONTROL SYSTEMS LABORATORY
ECE311
LAB 3: Control Design Using the Root Locus

1 Purpose

The purpose of this laboratory is to design a cruise control system for a car using the root locus.

2 Introduction

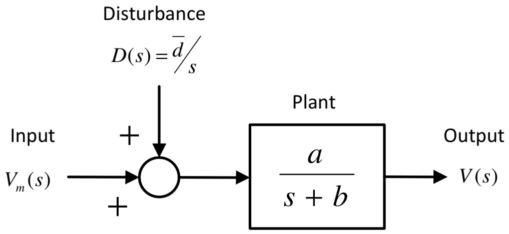

Figure 1: Block diagram of the plant

Recall the mathematical model of a car moving on a straight road with unknown slope θ derived in the introduction to laboratory 2. In the block diagram above, $\bar{d} = \frac{g}{a}\sin\theta$, where g is the gravitational constant.

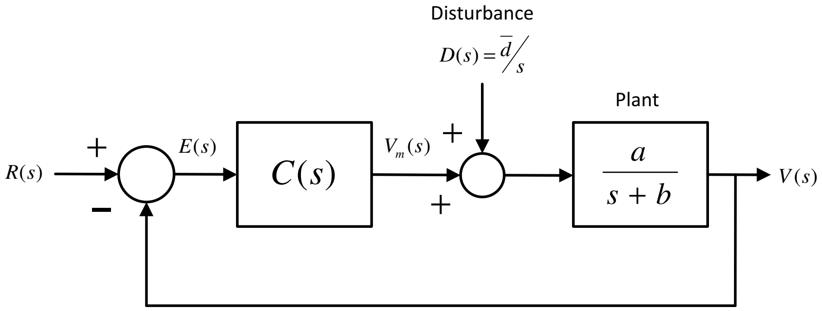

Figure 2: Block diagram of the closed-loop system

Problem Statement: Consider the feedback control system in Figure 2. Design the controller C(s) to meet the specifications below.

- (SPEC1) The output $v(t)$ of the closed-loop system should asymptotically track reference step signal $r(t)$ despite the presence of the unknown disturbance $d(t) = \overline{d} \cdot \mathbf{1}(t)$
- (SPEC2) The closed-loop system (input $R(s)$, output $V(s)$, and no disturbance, $D(s) = 0$) should be BIBO stable.
- (SPEC3) All poles of the closed-loop system should lie on the real axis, so that the output $v(t)$ does not have oscillatory behavior.
- (SPEC4) When $D(s) = 0$ (i.e., when the cart track is horizontal), the settling time T_s should be less than 0.3s.
- (SPEC5) The magnitude voltage $v_m(t)$ imparted by the control system to the DC motor should be less than 11.75V.

Concretely, you are required to design a control system making the cart, track constant reference speeds even when the track is not horizontal, and meeting certain transient performance specifications.

The experimental setup used in the lab is shown in figure 3.

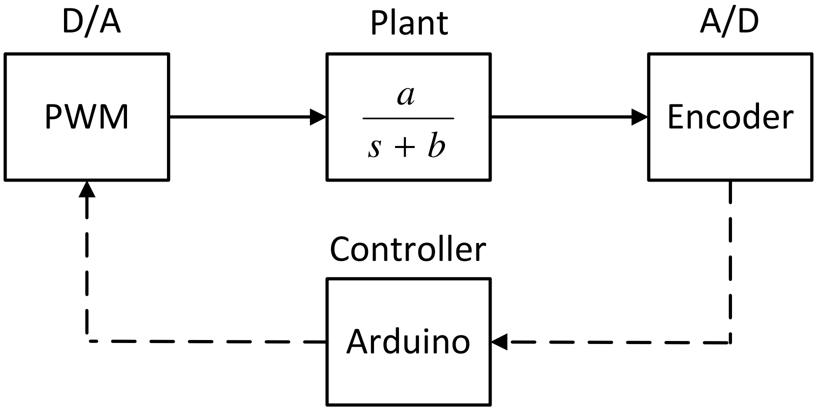

Figure 3: Block diagram of the experimental setup

The design controller will be implemented on the Arduino board. As shown in the figure, the micro controller samples the position of the cart from the encoder (A/D) and computes the control input to be applied to the cart. Given the desired control input, the motor shield generates a PWM signal that approximates the desired control value (D/A).

3 Preparation

1. Referring to Figure 2, verify that the plant has type 0. Hence, in order to meet SPEC1, you need a controller with a pole at zero. Letting $E(s) = R(s) - V(s)$, define the PI controller

$$\frac{V_m(s)}{E(s)} = K(1 + \frac{1}{T_I s}),\ K, T_I > 0.$$

Write the transfer function from $R(s)$ to $E(s)$ (assuming $D(s) = 0$), and from $D(s)$ to $E(s)$ (assuming $R(s) = 0$).

2. Let $R(s) = \bar{v}/s$ and recall that $D(s) = \bar{d}/s$. Using superposition, write the expression for $E(s)$ in the closed-loop system.

3. Applying the final value theorem to $E(s)$, show that if $\lim_{x\to\infty} e(t)$ exists, then it must be zero. You have thus shown that a PI controller is capable of meeting SPEC1.

Submit your derivations for points 1-3 to your lab TA at the beginning of the lab. Thoroughly read the next section prior to the lab.

4 Experiment

4.1 Identification of model parameters a and b

If you are using a different cart and track unit than the one used in lab 2, you need to re-stimate the plant parameters a and b following the procedure outlined in lab 2. If instead you are working with the same experimental setup, use the values of a and b you estimated in lab 2.

4.2 Controller design using Matlab

1. In your preparation you've shown that a PI controller is capable of meeting SPEC1. We now focus on SPEC2-SPEC4. Let $\theta = 0$ or, equivalently, $D(s) = 0$. Recall that an approximate expression for the settling time in a second-order system with complex conjugate poles is $T_s = \dfrac{4}{\sigma}$, where σ is the real part of the poles of the closed-loop system . In order to satisfy SPEC4, you thus need $\dfrac{4}{\sigma} \leq 0.2$, or $\sigma \geq 20$. You'll achieve this by an appropriate choice of K and T_I.
   Pick a guess for T_I, $T_I = 1$. Using Matlab, you will now draw the root locus of the system, i.e., the locus of the poles of the closed-loop system as the gain K is varied between 0 and ∞. First, define the transfer function $G(s)$ of the open-loop system without the gain K. This is given by
   $$G(s) = \frac{T_I s + 1}{T_I s} \frac{a}{s+b}.$$
   In Matlab, you can define $G(s)$ as follows
   ```
   >> TI=1; a=(your value); b=(your value);
   >> G=tf([TI 1],[Ti 0])*tf([a],[1 b]);
   ```
   The root locus is the locus of the poles of $KG(s)/(1 + KG(s))$ as K is varied. Plot the root locus by issuing the command.
   ```
   >> rlocus(G)
   ```
   Verify, using the plot, that for all $K > 0$ the poles of the closed-loop system have negative real part, and hence SPEC2 is met. However, for the present value of T_I, SPEC4 cannot be met by any choice of $K > 0$. Specifically, there doesn't exist $K > 0$ such that the closed-loop

system has two poles on the real axis with real part ≤ -20. Prove the truth of the claim by using the root locus plot.

Evidently, we need to choose a different value for T_I. Try different (positive) values of T_I. For each choice of T_I, plot the corresponding root locus. By trial and error, find the value of T_I compatible with this requirement: *there exist $K > 0$ such that the closed-loop system has two poles close to $s = -20$ on the negative real axis.* Save the root locus plot.

Once you found $T_I > 0$ satisfying the requirement above, you need to find $K > 0$ for which the closed-loop system has two poles close to $s = -20$. First, plot the root locus corresponding to the value of T_I you just selected. Next, issue the command

```
>> rlocfind(G)
```

Move the mouse cursor over the root locus and click on the desired location of the closed-loop poles on the real axis as $s = -20$. This action will return the value of K you were looking for. Notice that the PI controller with the values of K and T_I you have just found should meet SPEC2-SPEC4.

Have your TA sign here before proceeding to the next step.

2. Next, you'll double-check that using the values of K and T_I you just found, SPEC2-SPEC4 are met in simulation. Download the file `lab3.mdl` from Blackboard and open it. Begin by double-checking that your controller indeed meets SPEC4. Enter the values of θ, T_I and K into the Matlab workspace

```
>> a=(your value); b=(your value); theta = 0; TI=(your value); K=(your value);
>> VMAX_UPM=11.75;
```

Run the Simulink block by clicking on `Simulation > Start`. The scopes depict (i) the output $v(t)$ versus the reference signal $r(t)$, (ii) the tracking error $e(t)$, and (iii) the voltage $v_m(t)$. The reference signal is a square wave of frequency 0.5Hz and amplitude 0.2m/s.

Recall that the settling time T_s of $v(t)$ is the time $v(t)$ takes to reach and stay within the range

$$[v(\infty) - 0.02(v(\infty) - v(0)), v(\infty) + 0.02(v(\infty) - v(0))]$$

where $v(\infty)$ is the value $v(t)$ asymptotically settles to. By zooming in on one period of the simulation output, graphically estimate the settling time T_s. Save the plot you used to derive your estimate. Is it true that $T_s \approx 0.2$ sec.? Verify that SPEC5 is approximately met.

Have your TA sign here before proceeding to the next step.

3. Now you'll try to design a more "aggressive" PI controller. Similarly to what you did in step 1, use the root locus and trial error to find the value of $T_I > 0$ such that there exists $K > 0$ such that the closed-loop system has two poles close to $s = -30$. Use the command `rlocfind` to find the value of K for which the closed-loop system has two poles close to -30.

Enter the values of K and T_I you just found in the Matlab workspace and run the Simulink diagram `lab3.mdl`. Similarly to what you did in step 2, evaluate the settling time T_s by zooming in on one period of the simulation output. Save the plot. Compare the performance of this "aggressive" controller to that of the controller you evaluated earlier.

How do the settling times and overshoots compare?
Which controller is best suited to meet the specs?
What is the cause of the differences you observe?

Have your TA sign here before proceeding to the next step.

4.3 Controller Implementation

To perform the practical experiment for each part, open its Arduino file with ".ino" suffix and the associated Matlab m-file. You will enter your controller in the Arduino code and will record the data using Matlab. Follow your TA's instructions on how to compile the Arduino code and how to perform the experiment. Perform the following steps:

- Open the Arduino code file named as "$Lab3.ino$", The velocity reference is set to a square wave form with the amplitude of 0.2 m/s and frequency of 0.5 Hz.
- In the beginning of the code, modify and adjust the variables "K" and "Ti".
- Enter the values of K and T_I associated with the less "aggressive" controller. Upload the code and launch the Matlab m-file located in the same directory named "$real\_time\_data\_plot\_Lab3$".
- Save the obtained plot. Does your controller satisfy the design specifications?

  Have your TA sign here before proceeding to the next step.
- Now you'll test the performance of your controller against the unknown disturbance $d(t)$. Using one or more books, raise one end of the track. This corresponds to setting $\theta \neq 0$. While holding the track in a firm position, run the experiment again and compare the results.
- Now enter the values of K and T_I associated with the more "aggressive" controller and repeat all the above steps.

  Have your TA sign here before proceeding to the next step.

5 Report

Please submit your final report before the assigned deadline to your laboratory TA.