ECE-320 Linear Control Systems
Laboratory 7

Pole Placement - Diophantine Equations

Preview In this Lab you will obtain a model of a one degree of freedom system using the log decrement method and measuring the frequency response, determine the closed loop gain, and then do pole placement to achieve acceptable closed loop behavior.

Pre-Lab

1) Print out this lab and **read** it.

2) For the systems described by

$$G_{p1}(s) = \frac{28.6}{0.0012s^2 + 0.0073s + 1}$$
$$G_{p2}(s) = \frac{20.6}{0.0024s^2 + 0.0018s + 1}$$

use the program **diophantine.m** on the web site to construct both a first and second order controller to meet the desired constraints:

- the settling time is less than 0.5 sec
- the position error is less than 0.2 cm
- the percent overshoot less than 25%
- the magnitude of the real part of the closed loop poles is less than 100.

The program **diophantine** has the following arguments

- the amplitude of the step input (in cm)
- the system transfer function, in the form

$$G_p(s) = \frac{K_{clg}}{\frac{1}{\omega_n^2}s^2 + \frac{2\zeta}{\omega_n}s + 1}$$

- the location of the desired closed loop poles as a row vector (three poles for a first order controller, four poles for a second order controller)
- the order of the controller (1 or 2). If a second order controller is chosen, one pole will be automatically placed at the origin, making a first order system
- the final time to run the simulation to
- the filename containing the ECP data. If the ECP data is not yet available, enter ‘’

Lab

We need to first **identify** the system:

1. Estimate the second order system model using time domain analysis. Use the **log_dec** program to get estimates of ζ and ω_n.

2. Measure the frequency response and use **opt_fit_bode.m** to fine tune the model.

3. Estimate the closed loop gain, K_{clg} three times and average the result (see the last page if you still can't do this...).

Our general goals are as follows:

- track a step input of amplitude 1 cm
- the settling time is less than 0.5 sec
- the position error is less than 0.1 em
- the percent overshoot less than 25%
- the magnitude of the real part of the closed loop poles is less than 100.

The general form of the controller is

$$G_c(s) \quad = \quad \frac{B_0 + B_1 s + B_2 s^2}{A_0 + A_1 s + A_2 s^2}$$

For a first order system B_2 and A_2 are set to zero, while for a second order system we set A_0 to zero to give us a type one system.

You need to:

1. Construct your plant transfer function in the Matlab workspace. For this part, use the form

$$G_p(s) \quad = \quad \frac{K_{clg}}{\frac{1}{\omega_n^2}s^2 + \frac{2\zeta}{\omega_n}s + 1}$$

2. Run the program **diophantine.m**.

3. If the performance of the model is acceptable, you need to implement the controller on the ECP system. To do this:

   (a) Click **Setup** → **Control Algorithm**

   (b) Select **Dynamic Forward Path**

   (c) Click **Setup Algorithm**

   (d) Click **Import**

(e) Select **values.par** and click **open**. You should see values entered into the R and S arrays. These should look like the values the program **diophantine.m** spits to the screen.

(f) Click **OK**

(g) Click **Implement Algorithm**

(h) Click **OK**

4. Once the ECP system has run acceptably, you need to export the data, edit it, and the rerun **diophantine.m** to produce a plot of the predicted response and the model response.

5. You may need to reset the controller often, such as every time you want to implement a new controller. Click **Utility** → **Reset Controller**. Only do this **before** you have implemented a controller.

6. You may need to rephase the motor. Click **Utility** → **Rephase Motor**

7. Be sure to **Implement** the controller you have designed.

You should obtain an acceptable first order controller and an acceptable second order controller.

Memo Your memo should compare (briefly) the response of the model and the response of the real system for the different models you tried. You should have some description of the configuration of the system you were trying to control.
Specifically, it should include:

1. The configuration of the system you modelled.

2. The values of ζ, ω_n and K_{clg} for each of the two systems you modelled.

3. The optimal parameter estimates (from the optimal frequency response).

4. One plot of the step response (model and predicted) for your first order controller.

5. One plot of the step response (model and predicted) for your second order controller.

Estimating K_{clg}

0. Set the units

Click **Setup** → **User Units** and set the units to **cm**.

1. Setting up the controller

Click **Setup** → **Control Algorithm**. Be sure the system is set for *Continuous Time*. Select **PID** under **Control Algorithm**. Click on **Setup Algorithm**. Be sure **Feedback** is from **Encoder 1**. Set k_p to a small number (less than or equal to 0.05) and be sure $k_d = 0$ and $k_i = 0$. Then click **OK**. Next Click **Implement Algorithm**. The click **OK**.

2. Setting up the closed loop trajectory

Click **Command** → **Trajectory**. Select **Step** and click on **Setup**. Select **Closed Loop Step** and set **Step Size** to 0.5 to 1.5 cm. Be sure to record this step size (we'll refer to the amplitude as *Amp* below). Set the **Dwell Time** to something like 2000 ms, this is the time the system will be recording data. Finally click **OK**, then **OK** and you should be back to the main menu.

3. Executing the closed loop step

Click **Command Execute**. A menu box will come up with a number of options, and a big green **Run** button. Click on the **Run** button. When the system has finished collecting data, a box will appear indicating the how many sample points of data have been collected. (If you have hit a stop, the system stops recording data. This usually means you're input amplitude was too large or k_p was too large.) Click on **OK** to get back to the main menu.

4. Determining the steady state value

Click **Plotting** → **Setup Plot**, or just **Plotting Data** → **Plot Data**. Look at the steady state value ($x_{1,ss}$). You may need to change the dwell time if your system has not reached steady state.

5. Estimating K_{clg}

Estimate K_{clg} using the formula derived previously:

$$K_{clg} = \frac{x_{1,ss}}{k_p(Amp - x_{1,ss})} \tag{1}$$

You need to go through this procedure at least three times for each configuration. You must use at least two different values of k_p and two different values of input amplitude *Amp*. If none of the steady state values is larger than 0.4 cm, increase either k_p or *Amp*. Average the three results to get your K_{clg} (they should be similar). For the trials I've run, I've got K_{clg} between 8 and 40. Your's may be outside this range though.