# Step By Step Lab Instructions

## Lab 1 : System Identification and Discrete Equivalents.

[Login to the XP machine, and open ECP Executive].

[Open Load settings from “path”]

1. Load the initial settings
   1. Click File -> Load Settings
   2. Load Settings from the appropriate path (C:/ECE442/ece\_442\_lab\_1.cfg)
2. Verify the basic algorithm settings
   1. Click Setup + Control Algorithm
   2. Ensure “Type” is set to Discrete Time with Ts = 0.000884 (the minimum value).
   3. Ensure “Control Algorithm” is set to “State Feedback”
   4. Click “Setup Algorithm”
   5. Ensure K1 through K6 are all “zero” and Kpf is 1.0 (see image)
   6. Click “Implement Algorithm”
   7. This will allow us to measure the system’s open loop properties.
3. Verify Data capture settings.
   1. Click Data -> Setup Data Acquisition
   2. Verify Sample Period (servo cycles) is 1.
   3. Selected items should be : Commanded Position, Control Effort, Encoder 1 Position
4. Verify Trajectory Settings
   1. Click “Command + Trajectory
   2. Ensure Selection is “Step”
   3. Click Setup
   4. Verify the following settings:
   5. Step Size (counts): 100
   6. Dwell Time (msec): 2000
   7. Number of reps: 1
   8. Closed Loop Step
5. We are going to try to find the open loop properties of the system by finding the step response.
   1. Click Command + Execute
   2. Click Run
   3. Once the data is finished exporting, click Plotting + Plot Data.
   4. What final value did the system reach? What was the commanded value? What would you guess the command would need to be to reach a value of about 2000?
6. Adjust the commanded position (by setting the “step size” parameter in the trajectory settings window (similar to step 4) to a different value.
   1. Adjust it until the step reaches a final value near 2000. (+/- 10% is OK).
   2. Is the required command close to what you guessed in step 5?
7. In the following steps, we will try to find a discrete equivalent of the system.
8. Export the data to a convenient location.
   1. In notepad, find the time to peak (T\_p) and percent overshoot.
   2. Use these to calculate the damped frequency and zeta (damping factor?) in the continuous time transfer function.
   3. Compute sigma. Find the natural frequency.
   4. Compare the final value with the commanded value. This is the DC gain of the open loop system .
   5. Use the DC gain, sigma, and w\_d to find the continuous time transfer function of the system.
9. What are the poles of the the open-loop system? What is the natural frequency? What would be an appropriate sampling period for a controller for this system (remember the rule of thumb from class!)
10. In the ECP software, click Setup + Algorithms. You may notice that the discrete time sampling period (Ts) can only have certain values : multiples of 0.000884s. Select an allowed sampling period that is close to the one you chose in (9). It may be a good choice to select one that is slightly lower (faster sampling).
11. Use the matlab C2D function to find the equivalent discrete approximation of H(s). Use the transfer function obtained in (8) and the sample period you chose in (10).
12. How many poles and zeros appear in the discrete equivalent system?
    1. Where are the poles and zeros? (Draw a pole-zero diagram).
    2. What is the Nyquist Frequency (folding frequency) given the sampling period we chose?
    3. Keep in mind that we won’t want the final closed loop system to be anywhere near as fast as the Nyquist frequency.
    4. What is an upper limit on wc (using wc ~= 30\* wn (Nyquist frequency).

## Lab 2: Proportional Control

1. Recall the transfer function and poles and zeros from your discrete equivalent

Transfer Function: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Poles: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Zeros: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Sketch a Root Locus for a proportional-only controller.
2. Use the matlab rlocus command. Note that you can click any point on the root locus to find its gain. Graphically find the maximum gain K to achieve stability
3. Pick three points values of K within the stable range. It would be useful to pick one towards the lower end, one towards the upper end, and one near the middle.
4. For each K selected, find the closed loop transfer function and plot the step response in Matlab (see example code). Visually estimate the overshoot, DC accuracy, and Peak Time. Summarize your findings in the following table:

K1 \_\_\_\_\_\_\_ Closed Loop Transfer Function: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

K2\_\_\_\_\_\_\_ Closed Loop Transfer Function: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

K3\_\_\_\_\_\_\_ Closed Loop Transfer Function: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Gain Value K | % Overshoot | DC Accuracy | Peak Time |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Now we will actually implement the system with the gains you have selected
2. Load the initial settings
   1. Click File -> Load Settings
   2. Load Settings from the appropriate path (C:/ECE442/ece\_442\_lab\_1.cfg)
3. Verify the basic algorithm settings
   1. Click Setup + Control Algorithm
   2. Ensure “Type” is set to Discrete Time with Ts = **[Your Sample Period]**
   3. Ensure “Control Algorithm” is set to “PID”
   4. Click “Setup Algorithm”
   5. Set Kp = **[Your Gain]**, Kd = 0.0, and Ki = 0.0.
   6. Click “OK, then “Implement Algorithm”
4. Verify Data capture settings.
   1. Click Data -> Setup Data Acquisition
   2. Verify Sample Period (servo cycles) is 1.
   3. Selected items should be : Commanded Position, Control Effort, Encoder 1 Position
5. Verify Trajectory Settings
   1. Click “Command + Trajectory
   2. Ensure Selection is “Step”
   3. Click Setup
   4. Verify the following settings:
   5. Step Size (counts): 2000
   6. Dwell Time (msec): 6000
   7. Number of reps: 1
   8. Closed Loop Step
6. Test the step response of your closed loop system with your first gain K1
   1. Click Command + Execute
   2. Click Run.
   3. Wait until the data is finished exporting
   4. Click Plotting + Plot Data.
   5. Does the step response look anything like what you expected?
   6. Export your data (Click Data + Export raw data and pick a convenient save directory)
   7. Graph the data (Click Plotting + Plot Data) and approximate the Percent overshoot, TP, and DC accuracy of your closed loop system.
7. Test the step response of your closed loop system with the second and third gains, K2 and K3
   1. Adjust the gain using step 8.
   2. Test the system response using step 11.
   3. Graph and export this data like you did with K1. Summarize your findings in the below table.

|  |  |  |  |
| --- | --- | --- | --- |
| Gain Value K | % Overshoot | DC Accuracy | Peak Time |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Did the results for your test match the values predicted by Matlab?
2. If not, what could be causing the MATLAB simulation and the physical ECP hardware to yield different behavior?

## Lab 3: PI Control

We would like to create a closed-loop system with DC Accuracy (i.e. we would like it to eventually reach the position we commanded it to). Let’s use Matlab to design a PI controller for our system, and then use the ECP software to test it.

Compute a new G(z) = [PI] \* G(z) (copy from other lab outline).

Use matlab to plot the root locus of your controller.

Let’s say we would like the pair of complex conjugate poles to have a tp of at most \_\_\_ and damping factor such that its percent overshoot is at most \_\_\_ (65%)? And we would like the pole on the real axis to be dominant and have a 1% settling time of at most \_\_\_\_?

Select a zero location alpha and gain k to achieve this.

[gain range k vs stability if we have time. If not no biggie]

Plot the step response of your system in matlab.

Find the time at which the system reaches 50%, 70%, 90%, and 95% of the control input.

Compute gains K\_p and K\_i that will mechanize this system in the ECP software … (need to include derivation and compare/contrast with Dr Tharp’s deriviation from class)

1. Now we will actually implement the system with the gains you have selected
2. Load the initial settings
   1. Click File -> Load Settings
   2. Load Settings from the appropriate path (C:/ECE442/ece\_442\_lab\_1.cfg)
3. Verify the basic algorithm settings
   1. Click Setup + Control Algorithm
   2. Ensure “Type” is set to Discrete Time with Ts = **[Your Sample Period]**
   3. Ensure “Control Algorithm” is set to “PID”
   4. Click “Setup Algorithm”
   5. Set Kp = **[Your Kp Gain]**, Kd = 0.0, and Ki = **[Your Ki gain]**.
   6. Click “OK, then “Implement Algorithm”
   7. Click “OK”.
4. Verify Data capture settings.
   1. Click Data -> Setup Data Acquisition
   2. Verify Sample Period (servo cycles) is 1.
   3. Selected items should be : Commanded Position, Control Effort, Encoder 1 Position
   4. Click “OK”
5. Verify Trajectory Settings
   1. Click “Command + Trajectory
   2. Ensure Selection is “Step”
   3. Click Setup
   4. Verify the following settings:
   5. Step Size (counts): 2000
   6. Dwell Time (msec): 10000
   7. Number of reps: 1
   8. Closed Loop Step
6. Test the step response of your closed loop system with your gains Kp and Ki.
   1. Click Command + Execute
   2. Click Run.
   3. Wait until the data is finished exporting
   4. Click Plotting + Plot Data.
   5. Does the step response look like what you expected?
   6. Export your data (Click Data + Export raw data and pick a convenient save directory)
   7. View your data in notepad. Find the time at which the system reached 50%, 70%, 90%, and 95% of the commanded reference input (i.e. 1000, 1400, 1800, and 1900 counts).
7. Do the numbers match what was predicted by Matlab?
8. If not, why do you think these are different?