EE 4689

Control System Laboratory Supporting Materials and Experiments

Khosrow Rad

2016

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
CALIFORNIA STATE UNIVERSITY, LOS ANGELES
ECP Systems, Inc

Dr. Rad
Revise 6/2016
Published July 13, 2016

BRIEF CONTENTS

Experiment 0	ECE Hardware & Software	01
Experiment 1	Background Theory Torsional System System Identification of Torsional System	17
Experiment 2	Introduction to PD & PID PD& PID of Torsional System	26
Experiment 3	Collocated PD Control with 2 DOF Plant	33
Experiment 4	Successive Loop Closure/Pole placement Design	39
Experiment 5	LQR Control Design For 2 DOF plant	44
Experiment 6	Background Theory of Rectilinear System System Identification of a linear Plant	49
Experiment 7	Introduction to PD & PID PD & PID of a linear Plant	57
Experiment 8	Collocated PD Control With 2 DOF Plant	62
Experiment 9	PD Plus Notch Filter Control	65
Experiment 10	Successive Loop Closure/Pole Placement Design For 2 DOF Plant	69
Experiment 11	Background Theory of Inverted Pendulum System Identification of a linear Plant	76
Experiment 12	PD Control Of The Sliding Rod	95
Experiment 13	Low Pass Filter Augmentation Of Controller	102
Experiment 14	Pole Placement Controller For Unstable	105
Experiment 15	Background Theory of Gyroscope	107
Experiment 16	Matlab, Simulink, state feedback control	107

These laboratory note are reproduced in part from the ECP

ECP HARDWARE & SOFTWARE

Turning on Hardware
Turn on the ECP (model 205 or 210) control box by pushing the **BLACK ON** button. Make sure the mechanical system (weights/springs) is located in the correct position before you start your experiments. The ECP control box also has a **RED OFF** button. Use the **RED OFF** button to turn of the hardware in case of emergency.

In Case of Emergency
In case you experience a wildly moving mechanical system or when the mechanical system is vibrating strongly, make sure to push the **RED OFF** button on the front of the ECP control box. Turning of the ECP control box avoids damage to the experiment in case you specified the wrong control algorithm (unstable) and or reference signals that are too large. At all times, keep away from any moving parts. Make sure to push the **RED OFF** button also when someone is too close (less than 4 inches) to the experiment.

Start ECP software
Log in with your username and password and start the ECP software via the ECP icon on the desktop. Once ECP starts up, you will see a window similar to Figure 1.

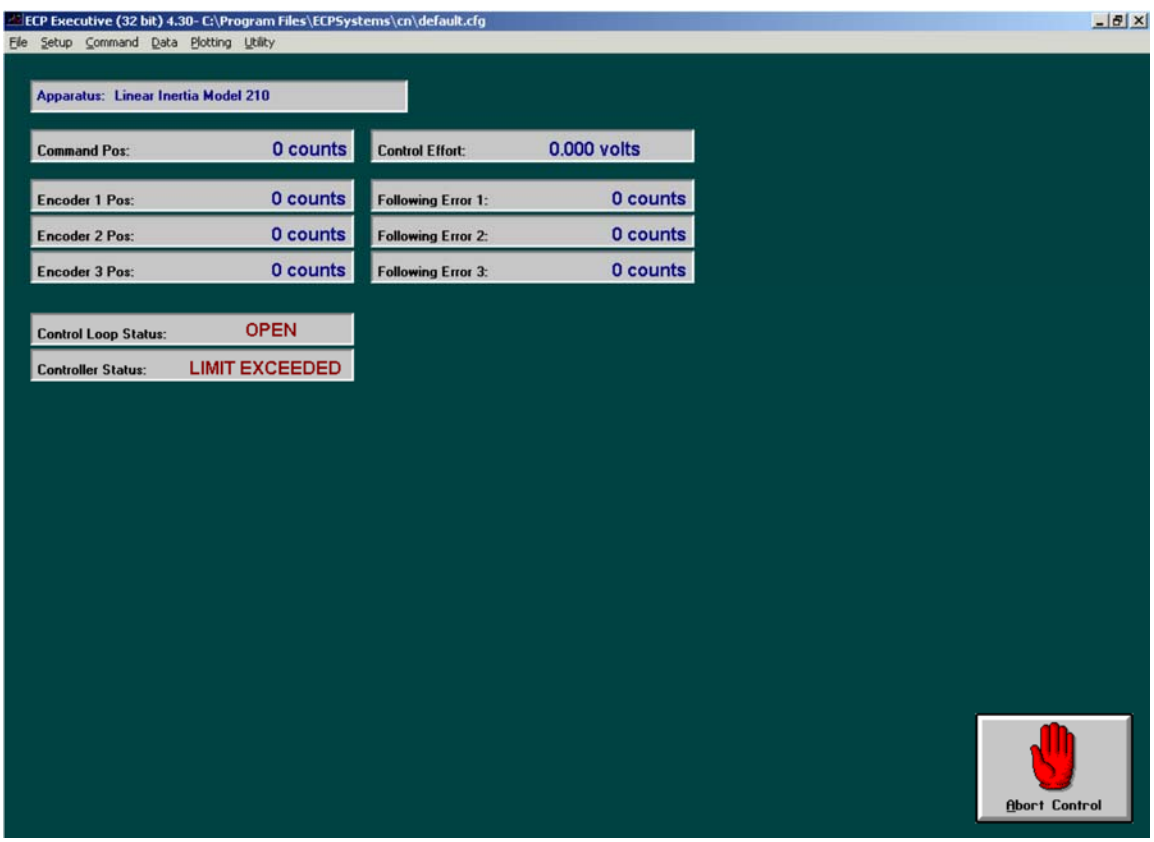


Figure 1: ECP main window

Open-loop versus Closed-loop experiments

- Open-loop experiments (no feedback controller implemented) are used to study the dynamics of the uncontrolled mechanical system. Closed-loop experiments require the *design of a feedback control algorithm* that continuously monitors the encoder positions and computes control signals for the Servo Motor to control the forces applied to the mechanical system. In summary:
- Open-loop experiments consist of applying *input signals to the Servo Motor* and *measuring the outputs* (encoder position of the different rigid bodies) of the mechanical system via Encoder 1 and/or 2. These experiments are done typically during your 1st and 2nd week of your lab.

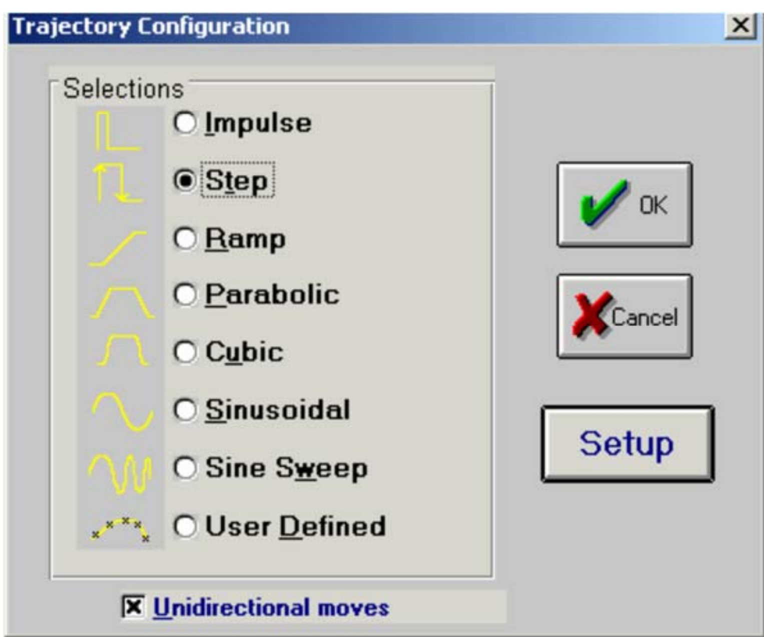
- Closed-loop experiments consist of applying *reference signals to the Servo Motor control loop* and *measuring the outputs* (encoder position of the different rigid bodies) of the mechanical system.

OPEN-LOOP EXPERIMENTS

An open-loop experiment for the rectilinear or torsional system only requires the specification of a open loop trajectory. For the specification of an open-loop trajectory, please follow these steps:

Setup of open-loop trajectory

1. First turn of any closed-loop control algorithm by clicking the large **Abort Control** button in the main ECP window (see Figure 1).
2. Select **Command - Trajectory** from the main menu bar (see Figure 1) to specify the test signal (called trajectory) for the servo Motor. A window similar to Figure 2 will open, allowing you to specify various test signals (trajectories). Typically we will use Step and Sinusoidal inputs.



Dr. Rad
Revise 6/2016

Figure 2: Trajectory Configuration Window

3. Select **Step** and click on **Setup** button to set up the step signal. A window similar to Figure 3 will open, allowing the specification of the Step Size, Dwell Time (how long is the step) and the number of repetitions.
   For an open-loop experiment, make sure to select **Open-Loop Step**. This causes the **Step Size** to be expressed in units of Volts, indicating an input signal on the Servo Motor in Volts.
4. Close the Configure Step Trajectory window (Figure 3) by a click on the **OK** button and close the Trajectory Configuration window (Figure 2) also by a click on the **OK** button

EXPERIMENTS & DATA ACQUISITION

Performing experiments (open- and/or closed-loop) requires setting up data acquisition parameters to indicate which signal should be measured during the experiment. Subsequently the experiment must be run to upload the experimental data. These steps are described in the following.

Set up data acquisition

Make sure you have set up the correct (open-loop or closed-loop) experiment as described earlier.

1. Select **Data - Setup Data Acquisition** from the main menu bar (See Figure 1) to specify which signals to measure during your open-loop experiment. A window similar to Figure 7 should appear.

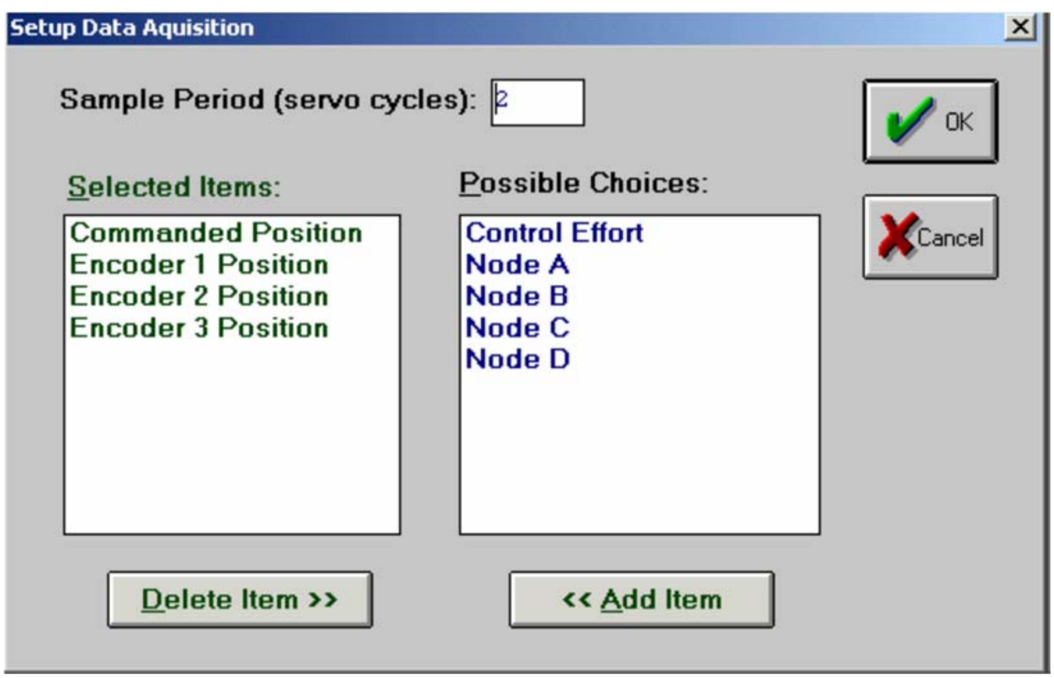


Figure 7: Setup Data Acquisition Window

2. Select the data you would like to measure by clicking on the following names:

   Encoder 1 Position = Output Position of Mass/Inertia 1
   Encoder 2 Position = Output Position of Mass/Inertia 2
   Encoder 3 Position = Output Position of Mass/Inertia 3 (typically NOT used)
   Control Effort = Input Signal to Servo Motor
   Commanded Position = Reference signal specified under Setup Trajectory (see above)
   Click on the **Delete Item** or **Add Item** buttons to respectively exclude or include that variable.
3. Think which data you would like to measure when you do your experiment. If you do an *openloop experiment*, you probably only want to measure **Control Effort**, **Encoder 1 Position** and/or **Encoder 2 Position**. For **closed-loop experiments** it is worthwhile to also measure the **Commanded Position** (reference) signal to inspect the steady-state error. Click on **OK** button to
   close the Setup Data Acquisition Window.

Perform Experiment for Data Acquisition
Make sure your have set up the data acquisition according to the steps described above.
1. Select **Utility - Zero Position** from the main menu (see Figure 1) to reset all encoder values to 0. This gives nice plots that will start at 0. You might have to do this several times in case of closedloop experiments.
2. Select **Command - Execute** from the main menu to execute your experiment and a window similar to Figure 8 will open.

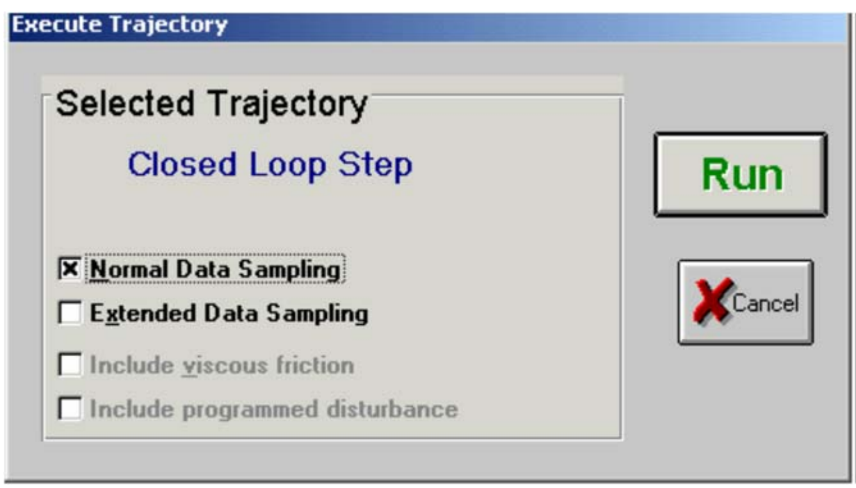


Figure 8: Execute Trajectory Window

Click on **RUN** button to start your experiment and data logging.

4. After the experiment ran successfully (no violent movement of mechanical system), an Upload Successful window similar to Figure 9 appears.

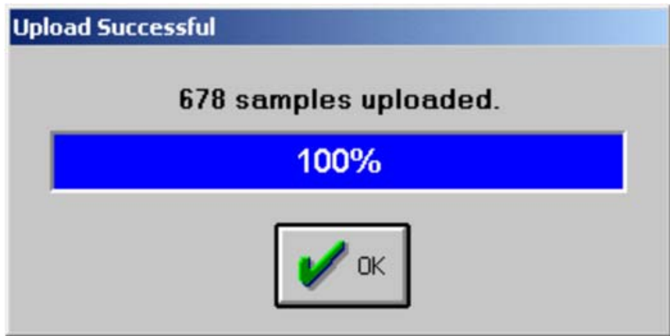


Figure 9: Upload Successful window after experiments completed

Click on **O.K.** to finish your experiment. If the Upload Successful window appears very quickly, the mechanical system might have hit the safety switches, causing the experiment to be terminated abruptly. In addition the control algorithm will be shut off and requires reimplementation. Try to reduce your step size or adjust the (PID) control parameters in case of closed-loop control to avoid this error message.

PLOTTING & SAVING DATA

Plotting Data

Make sure you were able to perform a successful experiment (no violent movement of the mechanical system) and that you were able to upload the data according to the steps described above.

1. Select **Plotting - Setup Plot** from the main menu (see Figure 1) to plot and examine the data from your experiment and a window similar to Figure 10 will appear.

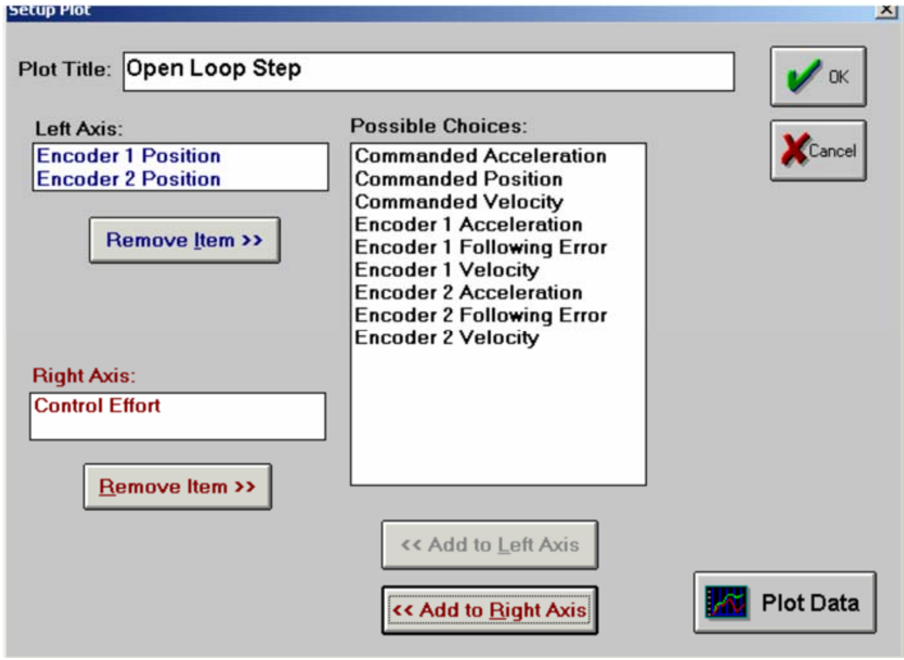


Figure 10: Setup Plot window

2. Select the data you would like to plot by clicking on the names and click on the **Remove Item** or **Add to Left Axis** or **Add to Right Axis** buttons to respectively exclude or include the plot of that variable. You can plot a maximum of 2 variables per axis and depending what you selected under Setup Data Acquisition, the following variables can be plotted.

   Encoder 1 Position = Position of Mass/Inertia 1
   Encoder 2 Position = Position of Mass/Inertia 2
   Encoder 3 Position = Position of Mass/Inertia 3
   Control Effort = Input Signal to Servo Motor
   Commanded Position = Trajectory specified

   Note that Velocity and Acceleration measurements are found by numerically differentiating the measurements and tend to be noisy!
3. For the open-loop experiments, it is best to plot the output (**Encoder 1 and/or 2 Position**) on the **Left Axis** and the input (**Control Effort**) on the **Right Axis**. For closed-loop experiments, it is best to plot the output (**Encoder 1 or 2 Position**) and the **Commanded Position** on the same axis to inspect overshoot and steady state errors.
4. Click on **Plot Data** button and a plot of the data will appear in a new window.

Saving Data

Make sure you have performed a successful experiment and were able to plot the data as described in the steps above. Whenever you run a new experiment, the data of the previous experiment will be overwritten in memory. So if you like your measurement and you would like to save the data for your report for plotting purposes, follow these steps:

1. Select **Data - Export Raw Data** from the main menu (See Figure 1) to save your data before you start a new experiment.
2. When saving the raw data, make sure you save it in your directory under**C:\labcourse**
3. The saved data will be text file where the data is stored column-wise and can be opened with Notepad and/or Excel. The data file can directly be read by the Matlab program **ecpread** available in your directory. The Matlab program **ecpread** is also used by the Matlab script **maelab** available in your directory to plot your simulation and experimental results for Model Validation and Controller validation (see next page).
4. You can also modify the text file into an m-file so Matlab can read the data. Details to convert the text file into an m-file can also be found in your lab handout and requires the following editing

steps:

a. First line in text file: Comment out the first line with **%**

b. Second line in text file: Enter **dummy=** before the opening bracket **[**.

c. Last line in text file: put a semicolon **;** behind the closing bracket **]**.

d. After last line in text file: define time vector **t**, input vector **u** and output **y** by **selecting the appropriate columns from the dummy variable**. For example, if you have selected to save the control effort (input **u**) and the encoder 1 position (output **y**), this can be done by adding the following lines to the end of the text file:

```
t=dummy(:,2);
y=dummy(:,3);
u=dummy(:,4);
clear dummy;
```

5. Save the raw text file as a file with the extension **.m**. Result is a Matlab script that can be run toread in your measurements.
6. Final note: make sure you use the variables **t**, **u** and **y** to define respectively the time vector, the input vector and the output vector. This allows the data file to also be read by the script file **maelab.m** to validate your models.

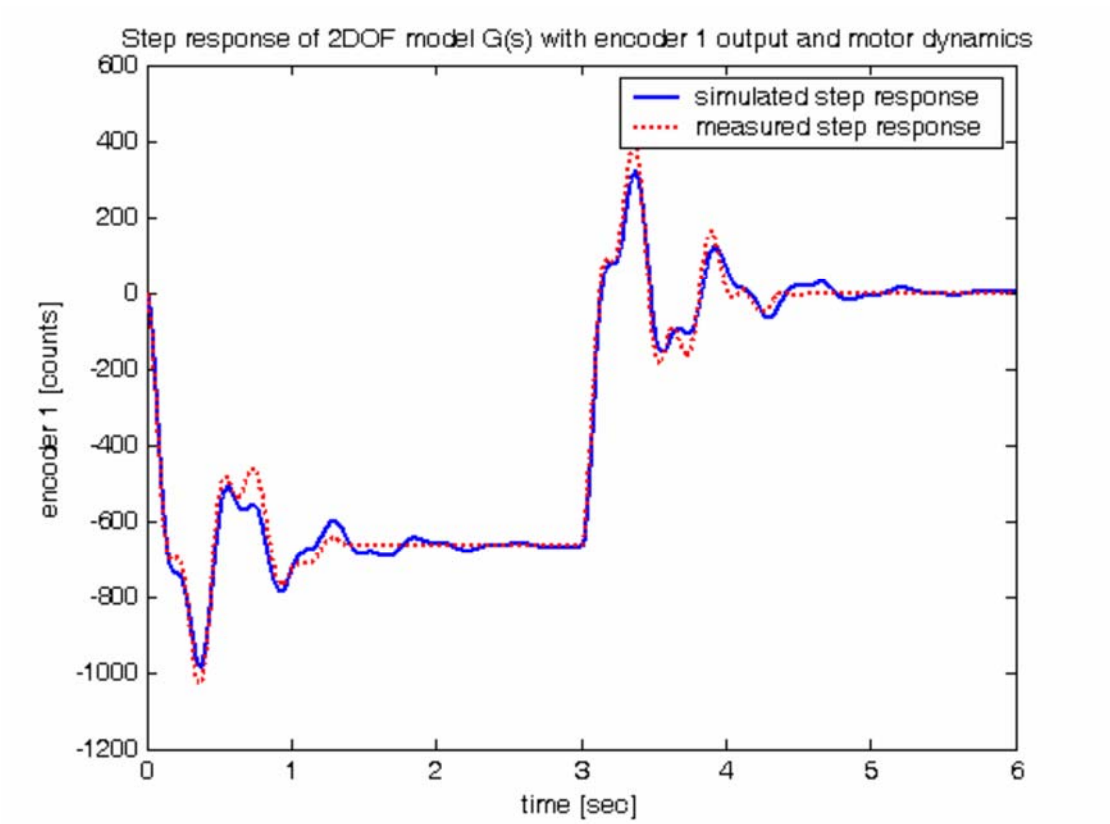
VALIDATION OF MODELS

Comparing experimental data with a simulation can validate models of the ECP rectilinear and torsional system. Matlab can handle experimental data and simulations and you are provided with a script file called **maelab** to perform all the necessary simulation, validation and control steps. To use the **maelab** script file, follow the following steps:

1. Start Matlab
2. In the Matlab command window, type in **pwd** and verify that you are indeed in your working directory under C:\labcourse\
3. Edit the file **parameters.m** by typing in **edit parameters** to specify the parameters of your model. Make sure to save the **parameters.m** file before you continue.
4. Run **maelab** script file by typing in **maelab** in the Matlab command window.
5. Specify the encoder output (1 or 2) you are interested
6. Specify the name of the filename that contains your model parameters (default **parameters.m**)
7. Specify degrees of freedom you would like to simulate. Specifying 1 will simulate a 1 Degree of Freedom (DOF) mass/damper/spring system, using only the model parameters

m1, d1 and k1. Specifying 2 will simulate a 2DOF mass/spring/damper system using m1, d1, k1 and m2, d2, k2.

8. Specify the sign of the encoder (1 or -1). This is necessary when a **positive step** on the system results in a **negative reading** (use -1) of the encoder output.
9. Use the menu option to "simulate open loop step response" or "simulate open loop sinusoidal response" and enter the values for the simulation. Typically the values should be the same as done during the experiment to be able to compare simulations with experiments.
10. NOTE: when asked for a filename, make sure to put the filename between quotes, e.g. **`myfile'**
11. For validation purposes, you should have a simulation and an experiment (based on step or sinusoidal excitation) that show close resemblance, similar to the figures below.

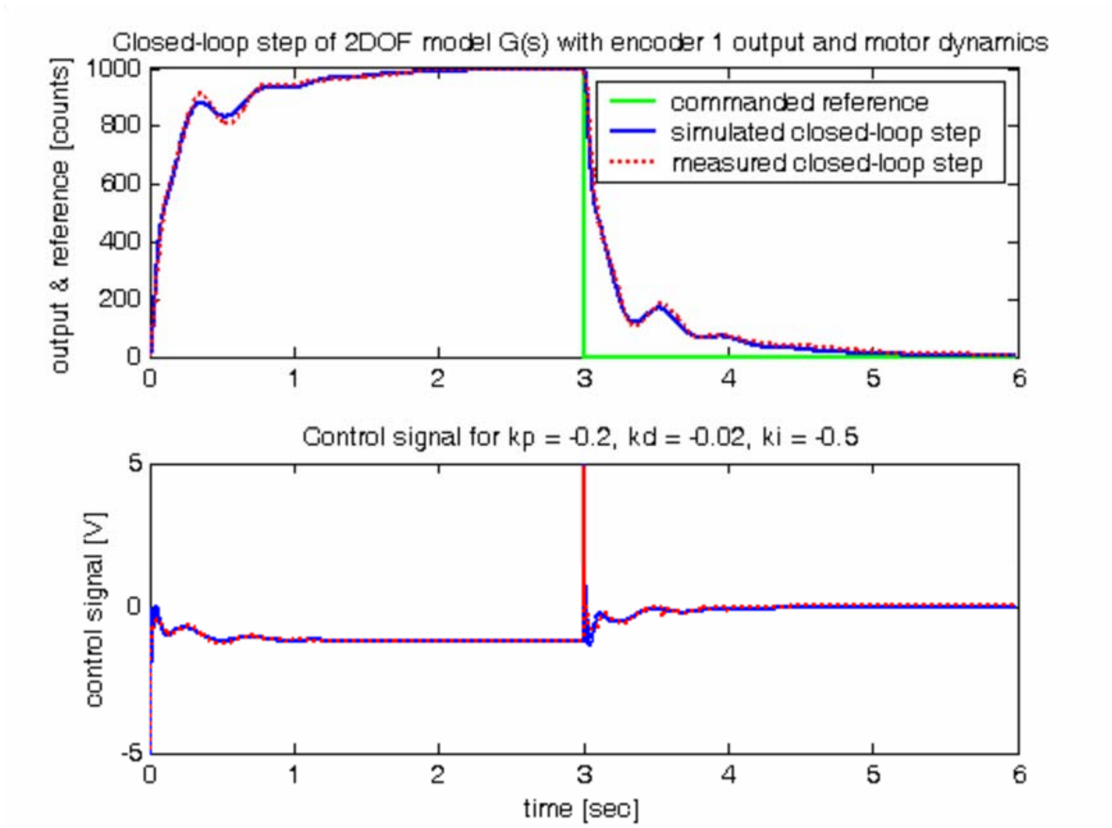


Example of figure for open-loop step-based model validation

VALIDATION OF CONTROL

Before implementing a (new) control algorithm, first verify the performance of your proposed P-, PD- or PID-control algorithm with Matlab by running a closed-loop simulation with the Matlab script file called **maelab**. Based on the model that you have validated (as indicated above) It allows you to verify whether your control algorithm will be stable on the actual system and to **verify whether the control signals stay within bounds and are not subjected to excessive oscillations** when making a step on the reference signal. The procedure to **validate the control algorithm before implementation** is as follows:

1. Start Matlab
2. In the Matlab command window, type in **pwd** and verify that you are indeed in your working directory under C:\labcourse\
3. Edit the file **parameters.m** by typing in **edit parameters** to specify the parameters of your model. Make sure to save the **parameters.m** file before you continue.
4. Run **maelab** script file by typing in **maelab** in the Matlab command window.
5. Specify the encoder output (1 or 2) you are interested
6. Specify the name of the filename that contains your model parameters (default **parameters.m**)
7. Specify degrees of freedom you would like to simulate. For validation of the control algorithm you must **specifying 2 to simulate the full 2DOF** mass/spring/damper system using m1, d1, k1 and m2, d2, k2 parameters.
8. Specify the sign of the encoder (1 or -1). This is necessary when a **positive step** on the system results in a **negative reading** (use -1) of the encoder output.
9. Use the menu option to "Design/evaluate feedback controller" and enter the numerical values for the **kp (proportional gain)**, **kd (derivative gain)** and **ki (integral gain)**. Keep in mind the bounds on the gains to avoid excessive control signals: **|kp| < 1, |kd| < 0.02 and |ki| < 1**.
10. Make sure to motivate and argument the choice of your kp, kd and ki values in your lab report based on the figures being created by **maelab** script file.
11. Use the menu option to "Simulate closed loop step response" and enter the values for the step response simulation. The values should be the same as done during an experiment for comparison purposes. Typically a closed-loop step should be in the order of 1000 counts.
12. NOTE: when asked for a filename, make sure to put the filename between quotes, e.g. **`myfile'.** Initially, for control validation purposes, no experimental data is required.
13. For final validation purposes at the end of the 3rd week, you should have a simulation and an experiment (based on a closed-loop step response) that show close resemblance, similar to the figure below.



Example of figure for validation of control algorithm based on closed-loop step data and simulation

CLOSED-LOOP EXPERIMENTS (2nd and 3rd week of lab)

A closed-loop experiment for the rectilinear or torsional system requires the specification of both a control algorithm and a closed-loop trajectory. For the specification of the control algorithms and the closed-loop trajectory, please follow these steps:

Setup of Control Algorithm

1. Select **Setup - Control Algorithm** from the main menu bar (see Figure 1) to specify the servo control algorithm. A window similar to Figure 4 will open.

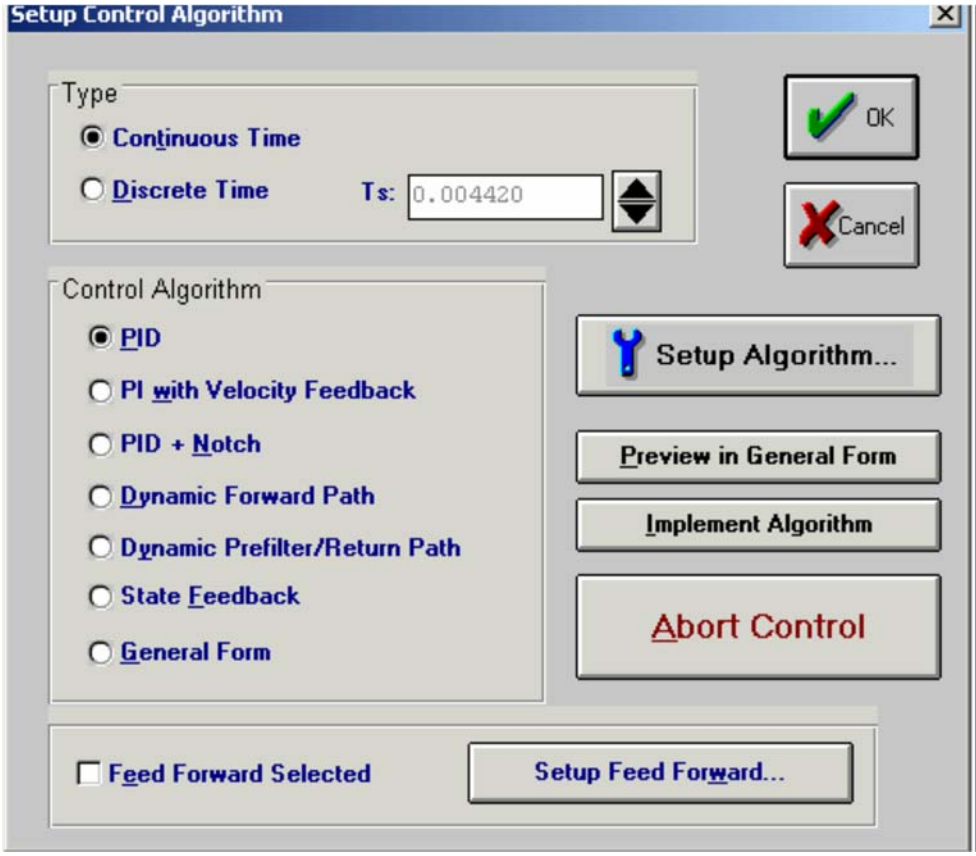


Figure 4: Setup Control Algorithm Window

2. Make sure *Type* is set to **Continuous Time** and the *Control Algorithm* is set to *PID* to be able to specify a PID controller.
3. Click on the **Setup Algorith**m button to set up the PID control algorithm and a window similar to Figure 5 will appear.

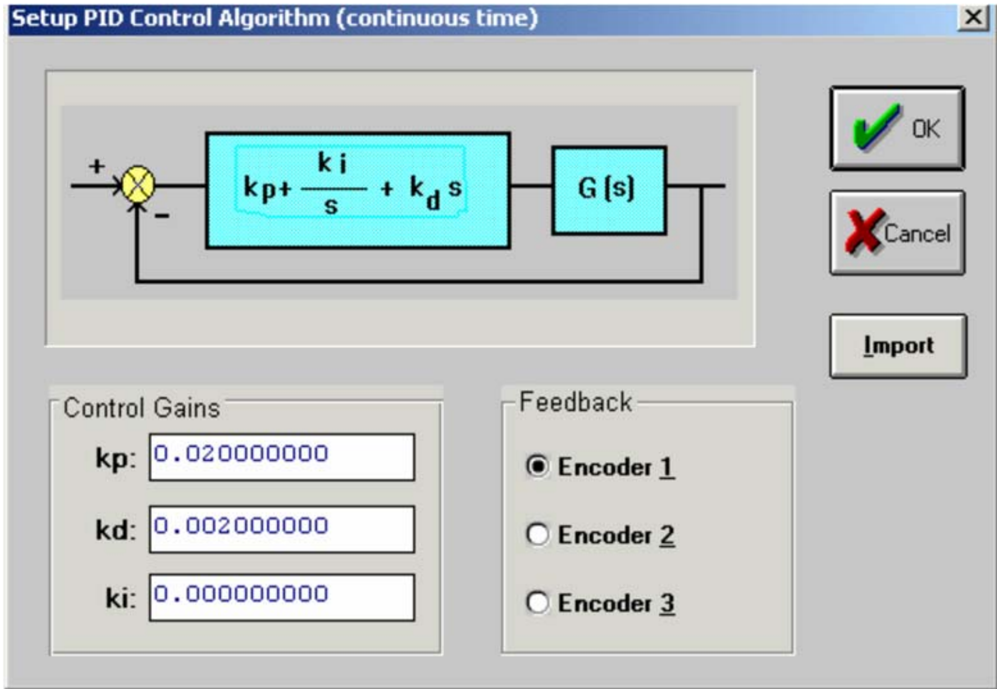


Figure 5: Setup PID Control Algorithm (continuous time) window

4. Make sure you select the correct **Encoder 1, 2 or 3** under *Feedback* and specify the value of the **Kp** (proportional gain), **Kd** (derivative gain) and the *Ki* (integral gain) in this window. Click on **OK** button to close the Setup PID Control Algorithm (continuous time) window.

NOTE: make sure Kp, Kd and Ki satisfy the following bounds to avoid excessive control signals: 0<Kp<1, 0<Kd<0.02 and 0<Ki<1. Especially the bound on Kd of 0.02 is important as large derivative gains cause strong vibrations in mechanical System that should be avoided at all times!

5. In the Setup Control Algorithm (see Figure 4) make sure to click the **Implement Algorithm** button to actually implement the control algorithm.

NOTE: keep the mouse hovering over the Abort Control button and give a small tap to the mechanical system. If this results in excessive movement of vibrations of the mechanical system your control algorithm may be destabilizing and a direct click on the Abort Control button shuts of the control algorithm

6. If the implementation of the control algorithm does not result in excessive movement of vibrations of the mechanical system click the **OK** button to close the Setup Control Algorithm (see Figure 4) window. You should see the control is being active by the **Control Loop Status: CLOSED** in the main window of ECP (see Figure 1).

Setup of closed-loop trajectory
1. Select **Command - Trajectory** from the main menu bar (see Figure 1) to specify the test signal (called trajectory) for the servo Motor. A window similar to Figure 2 will open, allowing you to specify various test signals (trajectories). Typically we will use Step and Sinusoidal inputs.
2. Select Step and click on Setup button to set up the step signal. A window similar to Figure 6 will open, allowing the specification of the Step Size, Dwell Time (how long is the step) and the number of repetitions.

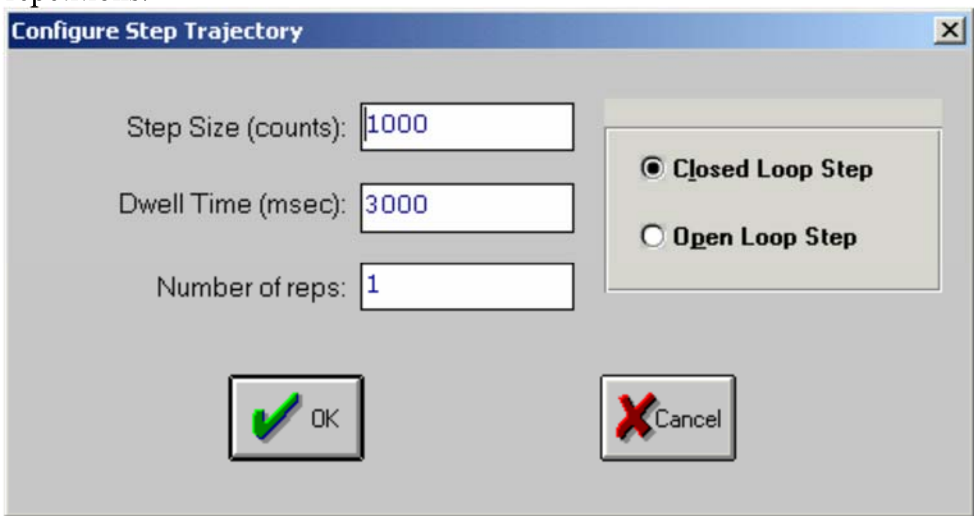


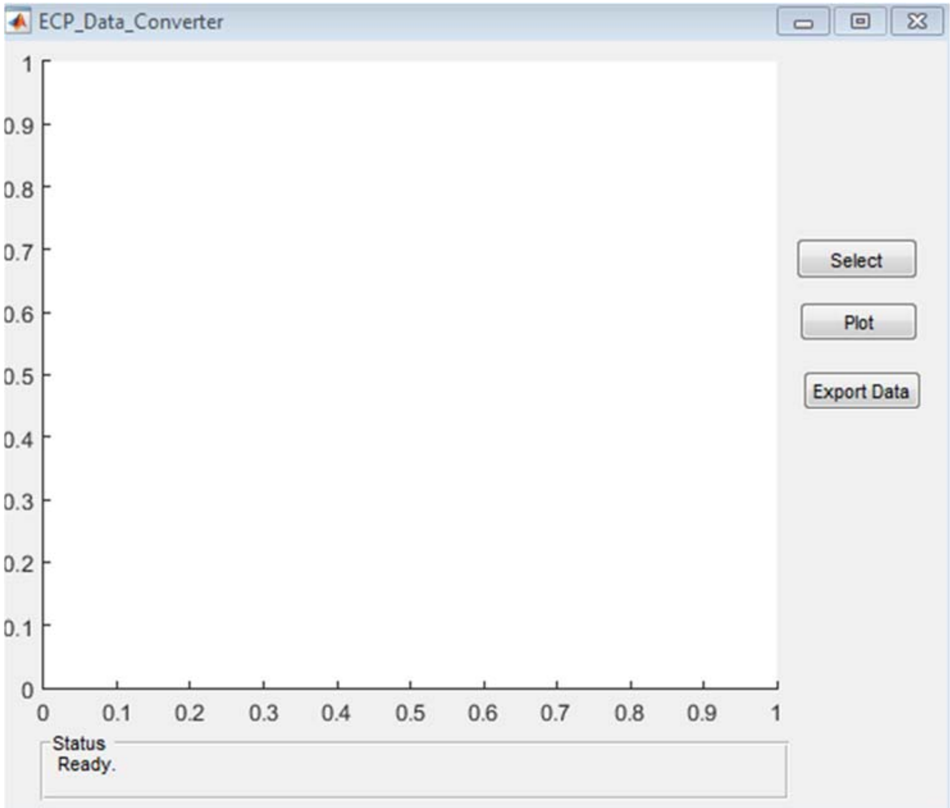
Figure 6: Configure Step Trajectory window for a closed-loop step

For a closed-loop experiment, make sure to select **Closed-Loop Step**. This causes the Step Size to be expressed in units of counts, indicating a reference signal on the Servo Motor control loop in Encoder counts.
3. Close the Configure Step window (Figure 6) by a click on the **OK** button and close the Trajectory Configuration window (Figure 2) also by a click on the **OK** button.
This file is to give you instructions on how to use ECP_Data_Converter.
The function of this program is to convert raw data from ECP programs to those acceptable for MATLAB. This will help relieve you from trivial work of copying data in a txt file and pasting them in a MATLAB script.

How to use MATLAB with ECP Data Converter, simply follow the steps:

Step 1: Decompressing the file, launch MATLAB, click 'Open' to select 'ECP_Data_Converter.m' and run this program (When asked if change the MATLAB current folder, choose 'Change Folder');

Step 2: A GUI, shown in Pic. 1, will pump out.



Pic. 1

Step 3: Click ‘Select’, a window will pop up, asking for ECP data file. After choose your data file, click ‘OK’. Once your file has been successfully loaded, ‘Status’ will show ‘You have chosen file: {*your file*}’, and then you can proceed to the next step;
Step 4: Click ‘Plot’, in the left blank space a plot containing the curves of all loaded data will appear. In addition, the Status will show ‘A plot has been generated’.
Step 5: Click ‘Export Data’, a window will pop up, asking for the address and name you would like to store your data. The data will be stored in a .mat file. If you use `load` in MATLAB, you will automatically get a bunch of vectors. These vectors are named after ‘Time’, ‘Encoder1Pos’, etc. And they store the corresponding data you just acquired from ECP. For example, you just obtained the counts of encoder 1 and encoder 2 and you save your data as ‘trial1.mat’ in the same folder with your MATLAB codes. In the beginning of your own MATLAB script, type `load trial1`. Then, you are free to use `Time, Encoder1Pos, Encoder2Pos` to do further data processing. If you drag trial1.mat directly to MATLAB command window, you will be able to see that the three variables will appear in workspace.

M205

Torsional Control System

Experiment#1

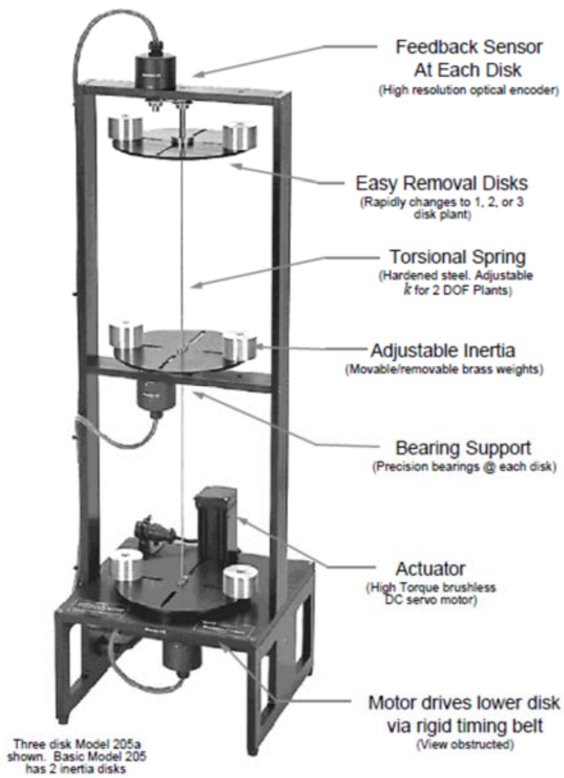
System Identification of a Torsional Control System

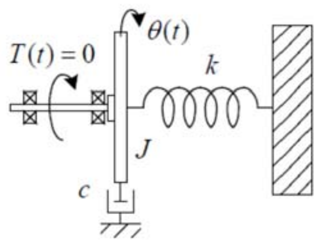
1. **Background Theory**

This experiment will give a procedure for identifying the plant parameters for a torsional dynamic system. The approach will be to directly measure the inertia, spring, and damping parameters and hardware gain by taking measurements of the plant while set up in a pair of classical spring mass configurations.

To become familiar with the operation of the equipment, it would be useful for any user to read section 1 of the manual in its entirety prior to undertaking the operations described here. Lack of time may preclude a detailed study of the manual, but, in any case, the safety portion of the manual, Section 1, must be read and understood prior to operating the equipment.

properties of lightly damped second order systems to indirectly measure the inertia, spring constant and damping coefficients of the plant by making measurements with the plant set up in a pair of classical spring-mass configurations.





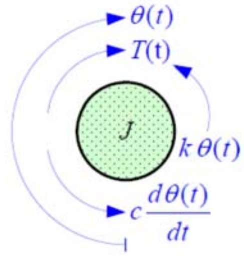


Figure 3: Single disk system

This type of system was discussed in lectures (see L3.9-3.13). Here, in the experiment we have no applied external torque, i.e. $T(t)=0$. The position of the disk is given by $\theta(t)$ (with $\theta=0$ being the equilibrium position, when $T=0$). k and c are the spring constant and the viscous damping coefficient respectively. The differential equation of motion is easily seen from the Free-body diagram to be

$$T(t)-c\frac{d\theta(t)}{dt}-k\,\theta(t)=J\frac{d^2\theta(t)}{dt^2} \tag{1}$$

We are exciting the system by displacing it from its equilibrium position, i.e. we have (nonzero) initial conditions $\theta(0)=\theta_0$ and $\left[\frac{d\theta(t)}{dt}\right]_{t=0}=0$. The motion for $t>0$ can be found by taking the Laplace transform (see lecture slide L1.18) of equation (1)

$$T(s)-c\left(s\,\Theta(s)-\theta(0)\right)-k\,\Theta(s)=J\left(s^2\Theta(s)-s\,\theta(0)-\left[\frac{d\theta(t)}{dt}\right]_{t=0}\right)$$

$$\Rightarrow \Theta(s)=\frac{(J\,s+c)\,\theta_0}{J\,s^2+c\,s+k}=\frac{(s+2\zeta\omega_n)\,\theta_0}{(s^2+2\zeta\omega_n s+\omega_n^2)} \tag{2}$$

$$\text{where } \omega_n^2=\frac{k}{J},\ \text{and } \zeta=\frac{c}{2J\,\omega_n} \tag{3}$$

We can rewrite equation (2) as (see lecture slide L4.12 for similar development)

$$\Rightarrow \Theta(s)=\frac{(s+\sigma_d)\,\theta_0}{(s+\sigma_d)^2+\omega_d^2}+\frac{\sigma_d\,\theta_0}{(s+\sigma_d)^2+\omega_d^2}\leftrightarrow \theta(t)=e^{-\sigma_d t}\cos(\omega_d t)\,\theta_0+\left(\frac{\sigma_d}{\omega_d}\right)e^{-\sigma_d t}\sin(\omega_d t)\,\theta_0 \tag{4}$$

$$\text{where } \sigma_d=\zeta\omega_n,\ \omega_d=\sqrt{1-\zeta^2}\,\omega_n \tag{5}$$

The peaks of the waveform and its locations can be found by solving for

$$\frac{d\theta(t)}{dt}=-\sigma_d\,e^{-\sigma_d t}\cos(\omega_d t)\,\theta_0-\omega_d\,e^{-\sigma_d t}\sin(\omega_d t)\,\theta_0-\left(\frac{\sigma_d^2}{\omega_d}\right)e^{-\sigma_d t}\sin(\omega_d t)\,\theta_0+\sigma_d\,e^{-\sigma_d t}\cos(\omega_d t)\,\theta_0=0$$

$$\frac{d\theta(t)}{dt}=0 \ \text{ whenever } \sin(\omega_d t)=0\leftrightarrow t=\frac{\pi}{\omega_d},\frac{2\pi}{\omega_d},\frac{3\pi}{\omega_d},\cdots$$

If $\theta_0>0$, then $t=\frac{2\pi}{\omega_d},\frac{4\pi}{\omega_d},\frac{6\pi}{\omega_d},\cdots$ correspond to maxima in the waveform (and $t=\frac{\pi}{\omega_d},\frac{3\pi}{\omega_d},\frac{5\pi}{\omega_d},\cdots$ correspond to minima). So the damped frequency ω_d is indeed the frequency of oscillation and can be obtained from the location of successive peaks in the waveform. Further, these peaks are given by

$$\theta\left(\frac{2n\pi}{\omega_d}\right)=e^{-2n\pi\sigma_d/\omega_d}\cos(2n\pi)\,\theta_0+\left(\frac{\sigma_d}{\omega_d}\right)e^{-2n\pi\sigma_d/\omega_d}\sin(2n\pi)\,\theta_0$$

$$\Rightarrow \theta\left(\frac{2n\pi}{\omega_d}\right) = e^{-2n\pi\sigma_d/\omega_d}\,\theta_0 \qquad (6)$$

So the ratio of 1[st] peak to the (n+1)[st] peak (which occurs $(2n\pi/\omega_d)$ seconds later) can be used to obtain the damping ratio as follows

$$\Rightarrow \frac{1^{\text{st}}\text{ peak}}{(\text{n+1})^{\text{st}}\text{ peak}} = \frac{e^{-2\pi\sigma_d/\omega_d}x_0}{e^{-2(n+1)\pi\sigma_d/\omega_d}x_0} = e^{2n\pi\sigma_d/\omega_d} = e^{2n\pi\zeta/\sqrt{1-\zeta^2}} \leftrightarrow \frac{\zeta}{\sqrt{1-\zeta^2}} = \frac{1}{2n\pi}\ln\left(\frac{1^{\text{st}}\text{ peak}}{(\text{n+1})^{\text{st}}\text{ peak}}\right) \qquad (7)$$

which for small ζ can be approximated as $\zeta = \frac{1}{2n\pi}\ln\left(\frac{1^{\text{st}}\text{ peak}}{(\text{n+1})^{\text{st}}\text{ peak}}\right)$.

Appendix B - Model for Part 3

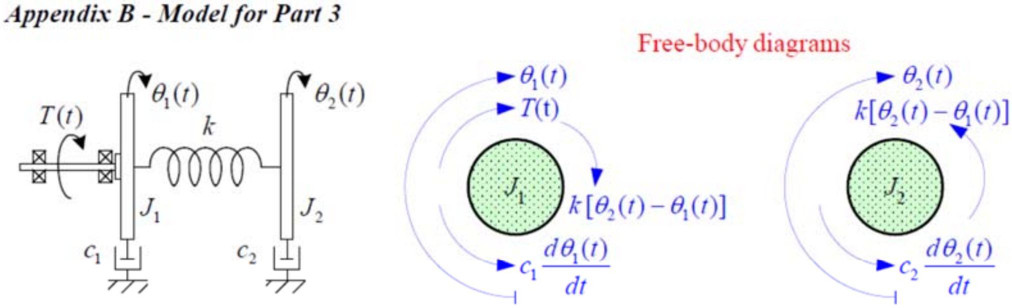


Figure 4: Two disk system

The differential equations of motion are easily seen from the Free-body diagrams (see lecture slides L3.9-L3.13 for similar development) to be

$$T(t) - c_1\frac{d\theta_1(t)}{dt} + k[\theta_2(t)-\theta_1(t)] = J_1\frac{d^2\theta_1(t)}{dt^2} \qquad (8)$$

$$-c_2\frac{d\theta_2(t)}{dt} - k[\theta_2(t)-\theta_1(t)] = J_2\frac{d^2\theta_2(t)}{dt^2} \qquad (9)$$

We can take their Laplace Transform (assuming zero initial conditions) and solve as follows

$$\begin{bmatrix} J_1s^2+c_1s+k & -k \\ -k & J_2s^2+c_2s+k \end{bmatrix}\begin{bmatrix}\Theta_1(s)\\ \Theta_2(s)\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix}T(s)$$

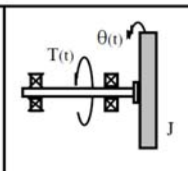
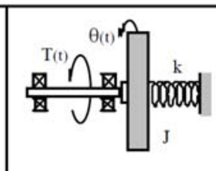
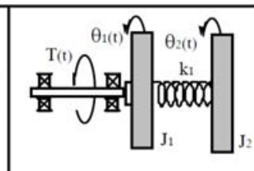
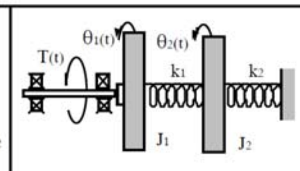
$$\Rightarrow \begin{bmatrix}\Theta_1(s)\\ \Theta_2(s)\end{bmatrix} = \begin{bmatrix} J_1s^2+c_1s+k & -k \\ -k & J_2s^2+c_2s+k \end{bmatrix}^{-1}\begin{bmatrix}1\\0\end{bmatrix}T(s) = \frac{\begin{bmatrix} J_2s^2+c_2s+k & k \\ k & J_1s^2+c_1s+k \end{bmatrix}}{(J_1s^2+c_1s+k)(J_2s^2+c_2s+k)-k^2}\begin{bmatrix}1\\0\end{bmatrix}T(s)$$

From this, we get the two transfer functions of interest given below

$$\Rightarrow G_1(s) = \frac{\Theta_1(s)}{T(s)} = \frac{J_2s^2+c_2s+k}{J_1J_2s^4+(c_1J_2+c_2J_1)s^3+(kJ_1+kJ_2+c_1c_2)s^2+(kc_1+kc_2)s} \qquad (10)$$

$$\Rightarrow G_2(s) = \frac{\Theta_2(s)}{T(s)} = \frac{k}{J_1J_2s^4+(c_1J_2+c_2J_1)s^3+(kJ_1+kJ_2+c_1c_2)s^2+(kc_1+kc_2)s} \qquad (11)$$

The transfer functions relating angular velocity to torque (if desired) are obtained by multiplying by s.

Plant Models	 Rigid body	 Free-clamped	 Free-free, 2 DOF	 Free-clamped, 2 DOF
Time Domain Equations	$J\ddot{\theta}^{(t)} = T^{(t)}$	$J\ddot{\theta}^{(t)} + k\theta^{(t)} = T^{(t)}$	$J_1\ddot{\theta}_1^{(t)} + k(\theta_1^{(t)} - \theta_2^{(t)}) = T^{(t)}$ $J_2\ddot{\theta}_2^{(t)} + k(\theta_2^{(t)} - \theta_1^{(t)}) = 0$	$J_1\ddot{\theta}_1^{(t)} + k_1(\theta_1^{(t)} - \theta_2^{(t)}) + k_2\theta_1^{(t)} = T^{(t)}$ $J_2\ddot{\theta}_2^{(t)} + k_1(\theta_2^{(t)} - \theta_1^{(t)}) + k_2\theta_2^{(t)} = 0$
S-Domain Equations	$\frac{\theta(s)}{T(s)} = \frac{1}{J s^2}$	$\frac{\theta(s)}{T(s)} = \frac{1}{J s^2 + k}$	$\frac{\theta_1(s)}{T(s)} = \frac{J_2 s^2 + k}{D(s)}, \quad \frac{\theta_2(s)}{T(s)} = \frac{k}{D(s)}$ $D(s) = s^2\left(J_1 J_2 s^2 + (J_1 + J_2)k\right)$	$\frac{\theta_1(s)}{T(s)} = \frac{J_2 s^2 + k_1 + k_2}{D(s)}, \quad \frac{\theta_2(s)}{T(s)} = \frac{k_1}{D(s)}$ $D(s) = J_1 J_2 s^4 + \left(J_1(k_1 + k_2) + J_2 k_1\right) s^2 + k_1 k_2$
Character-istics	• Rigid body model. • Type 2 system. • See page 5.	• Classic spring-mass oscillator • Type 0 system • Single vibration mode	• Rigid body plus 1 oscillatory mode. • Type 2 system. • θ_1/T: 2 imag zeros, pole excess = 2 • θ_2/T: no zeros, pole excess = 4	• 2 oscillatory modes. • Type 0 system. • θ_1/T: 2 imag zeros, pole excess = 2 • θ_2/T: no zeros, pole excess = 4

Experimental work

Obtain a set of masses, springs, tools and the lab manual from your tutor. (Make sure you return all of these at the end of the session)

The ECP equipment has three disks. This experiment will derive parameters for the bottom disk 1 and the top disk 3. There are three parts to the experiment (*Figure 1* shows the corresponding configurations):

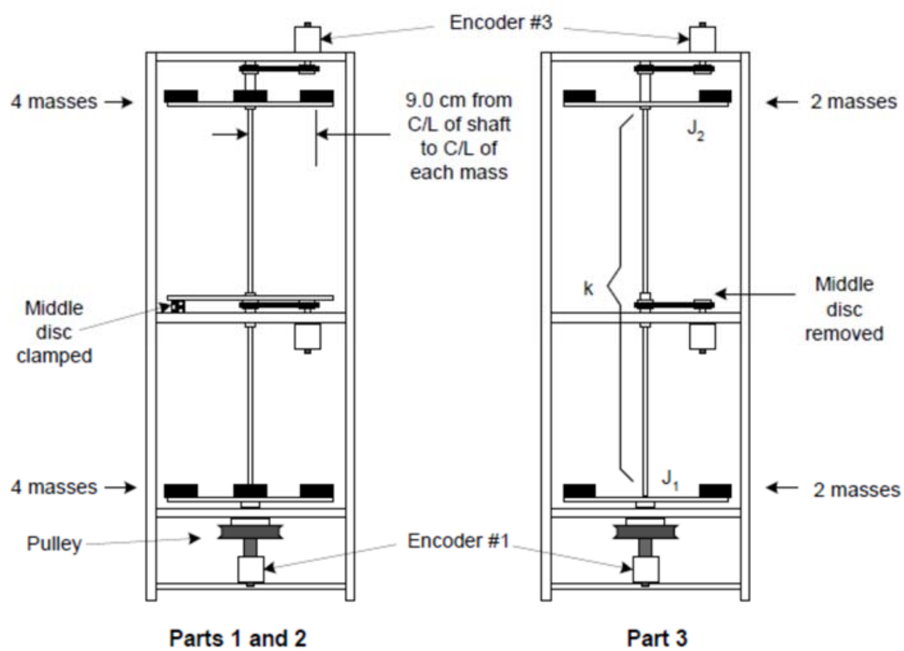


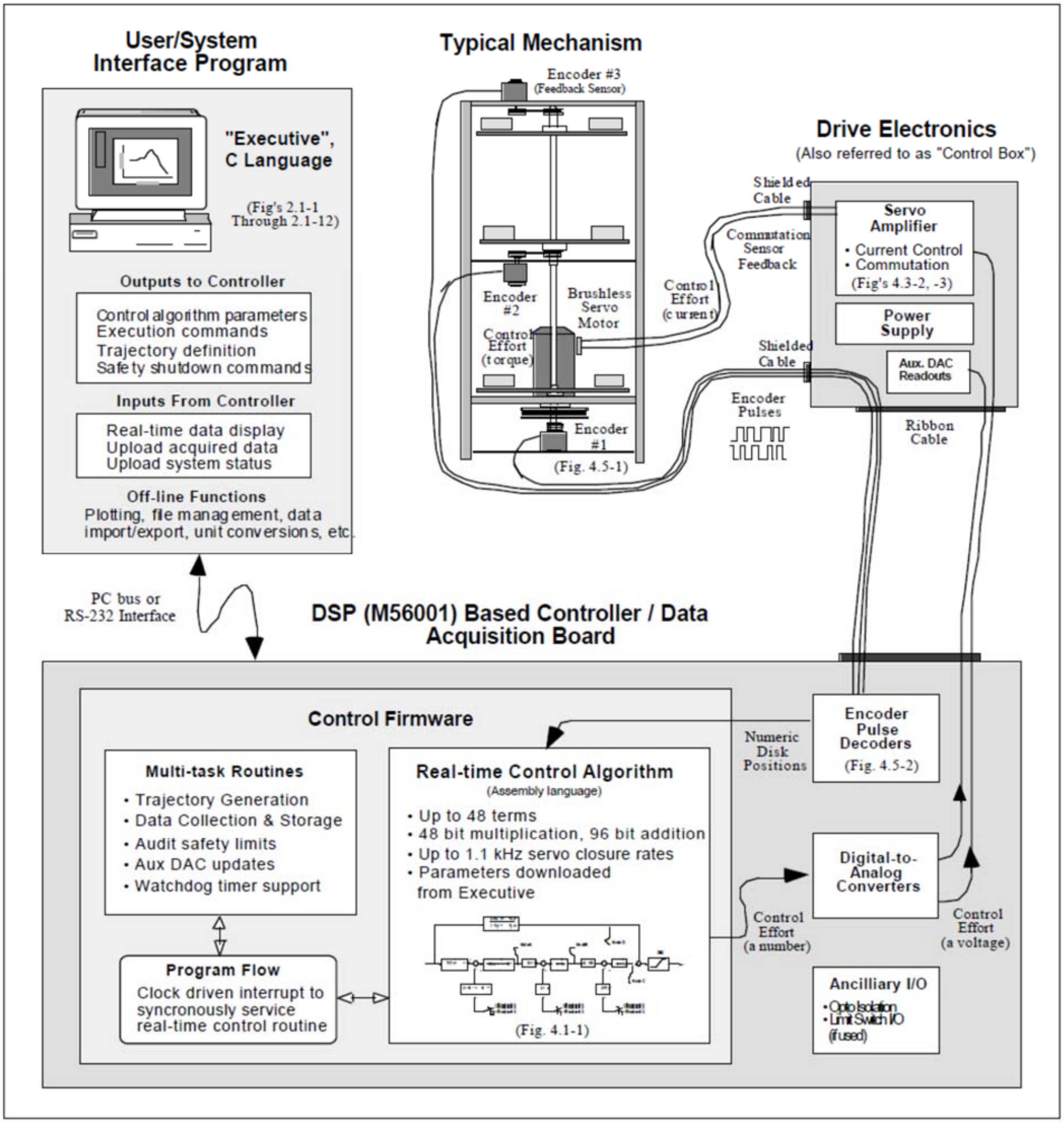
Figure 1: Test configurations

1. parameters for disk 1,
2. parameters for disk 3, and
3. test on a two disk system.

It is recommended that you save data and control configuration files regularly to avoid work loss should a system fault occur.

Real-Time Control Implementation

A functional overview of the control system is shown in Figure 4.0-1. The system is comprised of three susbsystems: The mechanism including motor and sensors, the real-time controller / drive electronics, and the user/system (“Executive”) interface software.



Overview of Real-time Control System.
This architecture is consistent with modern industrial control implementation.

An brief survey of the system architecture is afforded by tracing the data flow as the system is operated. The user specifies the control algorithm in the Executive program and downloads it (via "Implement Algorithm") to the DSP based real-time controller board. The DSP immediately executes the algorithm at the specified sample rate. This involves reading the reference input[1] and feedback sensor (optical encoders) values, computing the algorithm, and outputting the digital control effort signal to the digital to- analog converter (DAC).
The DAC converts the resulting stream of digital words to an analog voltage which is transformed to a current by the servo amplifier and then to a torque by the motor. The mechanism transforms the motor input to motion at the desired output according to the plant dynamics (i.e. equations of motion). These plant outputs are sensed by the encoders which output a stream of pulses[2]. The pulses are decoded by a counter on the DSP board and made available as a digital position word to the real-time control algorithm.
When the user specifies a trajectory and subsequently commands the system to "Execute" the maneuver, the trajectory parameters are downloaded to the controller board. The DSP generates corresponding reference input values for use by the real-time control algorithm. Throughout the maneuver, any data specified by the user is captured and stored in memory on the board. On completion of the maneuver, the data is uploaded to PC memory where it is available for plotting and storage.

Part 1 - Parameters for disk 1

Carry out the following steps to determine the inertia and coefficient of friction for the first disk and the torsional spring constant:

2. Clamp the centre disk to put the mechanism in the configuration shown in the configuration diagram (shown in *Figure 1*), using the 1/4" bolt, square nut, and clamp spacer. Only light torqueing on the bolt is necessary.
3. Secure four 500g masses on disk 1 as shown in *Figure 1*. Verify that the masses are secured and that each is at a centre distance of 9.0 cm from the shaft centre-line.
4. With the controller powered up, go to ***Set-up/Control Algorithm*** and set . Go to ***Command /Trajectory*** and select ***Step, Set-up***. Select ***Open Loop Step*** and input a step size of zero, duration of 4000 ms and 1 repetition. This puts the controller in a mode for acquiring 8 sec of data on command but without driving the actuator. Exit to the background screen by consecutively selecting ***OK***. This procedure may be repeated later to vary the data acquisition period. T_S = 0.00442. Go to ***Data/Set up Data Acquisition*** and ensure that ***Encoder #1*** is selected as data to acquire and specify data sampling every 2 servo cycles (i.e. every). Select ***OK*** to exit. T_S = 2. Select ***Utility /Zero Position*** to zero encoder positions.
5. Go to ***Command /Execute***. Prepare to manually displace the disk 1 approximately 20 degrees. Exercise caution in displacing the inertia disk; displacements beyond 40 degrees may damage and possibly break the flexible drive shaft. (Displacements beyond 25 degrees will trip a software limit that disables the controller indicated by "***Limit Exceeded***" in the ***Controller Status*** box in the Background Screen. To reset, simply reselect ***Execute*** from the ***Execute*** menu.) With the disk displaced approximately 20 degrees (≤1000 encoder counts as read on the Background Screen display) in either direction, select ***Run*** from the ***Execute*** box and release the disk approximately 1 second

later. The disk will oscillate and then settle while encoder data is collected to record the response. Select ***OK*** after the data is uploaded.
Export the data from ECP to MATLAB, and plot the Encoder 3 data using MATLAB. Be sure to clearly label the plots. To export the data to MATLAB, refer to Page 13

6. Go to ***Plotting /Set-up Plot*** and choose ***Encoder #1 Position*** then select ***Plot Data*** from the ***Plotting*** menu. You will see the time response.
   The damped frequency (see section 1) can be obtained by measuring the interva l between successive peaks of the waveform (This is valid for at least the first few large peaks. Smaller peaks later are dominated by nonlinear friction effects and do not reflect the salient system dynamics). Note that we are using "d11" as a subscript to denote disk #1, trial #1. You may "zoom" the plot via ***Axis Scaling*** for more precise measurement. Similarly, the damping ratio can be obtained by comparing heights of successive peaks (see section 1). Natural frequency ω_{nd11} can be obtained from ω_{dd11} and ζ_{d11}

$$\omega_{dd11} = \sqrt{1-\zeta_{d11}^2}\,\omega_{nd11}$$

   Make sure to export the data from ECP to MATLAB and create a new figure for this data, and clearly label the plot and the beginning and end points as you did in the step before

   How does this damping ratio compare with that for the upper disk? Why might it be different?
7. Close the graph window by clicking on the left button in the upper right hand corner of the graph. This will collapse the graph to an icon, from where it may later be brought back up.
8. Remove the four masses from the disk and repeat steps 3 through 7 to obtain , ω_{dd12} , ζ_{d12} , and ω_{nd12} for the unloaded disk. If necessary, in step 3, adjust the execution (data sampling only in this case) duration.
9. Each brass weight has a mass of 500 ± 5 g and a diameter of 5.00 ± 0.02 cm. Calling the inertia of the four masses combined J_m , use the following relationships (see section 1) to solve for the unloaded disk inertia J_{d1} , and the lower torsional spring constant k_{d1}
10. $$\frac{k_{d1}}{J_m + J_{d1}} = \omega_{nd11}^2 \quad \text{and} \quad \frac{k_{d1}}{J_{d1}} = \omega_{nd12}^2$$
11. Here J_m is the moment of inertia of the four 500 g masses about the disk axis and has a value of 0.0168 Kg-m^2. Note that the calculated inertia J_{d1} will include the reflected inertias of all connected elements such as motor, belts and pulley. Finally, the damping coefficient can be found by using

Transfer Function Calculation

The so-called hardware gain, k_{hw},[1] of the system is comprised of the product:

$$k_{hw} = k_c k_a k_t k_p k_e k_s \quad (12)$$

where:

k_c, the DAC gain, = 10V / 32,768 DAC counts
k_a, the Servo Amp gain, = approx. 2 (amp/V)
k_t, the Servo Motor Torque constant = approx. 0.1 (N-m/amp)
k_p, the Drive Pulley ratio = 3 (N-m @ disk / N-m @ Motor)
k_e, the Encoder gain, = 16,000 pulses / 2π radians
k_s, the Controller Software gain, = 32 (controller counts / encoder or ref input counts)[2]

The report is to include the following:
A diagram identifying the control elements and signals in the Torsion Experiment.
Sensor:
Actuator:
Controller:
Reference Input:
Actuator Output:
System Output:
MATLAB Plots, with Data Cursor Points on each plot, along with titles, labels and legends if necessary that clearly show which plot corresponds to which situation.
- Disk 3 Trial 1
- Disk 3 Trial 2
- Disk 1 Trial 1
- Disk 1 Trial 2
Calculations the following values, along with units
- Natural frequencies
- Damping ratios
- *Inertia of the masses (J_m)
- Inertia of Disk 1 (J_{d1})
- Inertia of Disk 3 (J_{d3})
- Damping constant on Disk 3 (c_{d3})
- Damping constant on Disk 1 (c_{d1})
- Spring constant on Disk 3 (k_{d3})
- Spring constant on Disk 1 (k_{d1})
For calculate the inertia of each weight about its center of gravity ($J_{cg}=0.5m*r^2$).
Then, use the parallel axis theorem to get the inertia about the center of rotation ($J=J_{cg}+mR^2$).
Then multiply four to get J_m

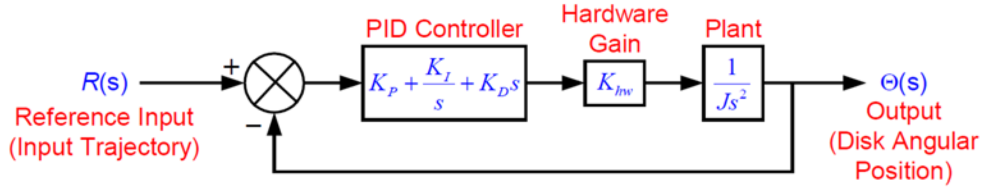
[1]It contains software gain also. This software gain, k_s is used to give higher controller-internal numerical resolution and improves encoder pulse period measurement for very low rate estimates.
[2]The "controller counts" are the counts that are actually operated on in the control algorithm. i.e. The system input (trajectory) counts and encoder counts are multiplied by 32 prior to control law execution.

Experiment#2

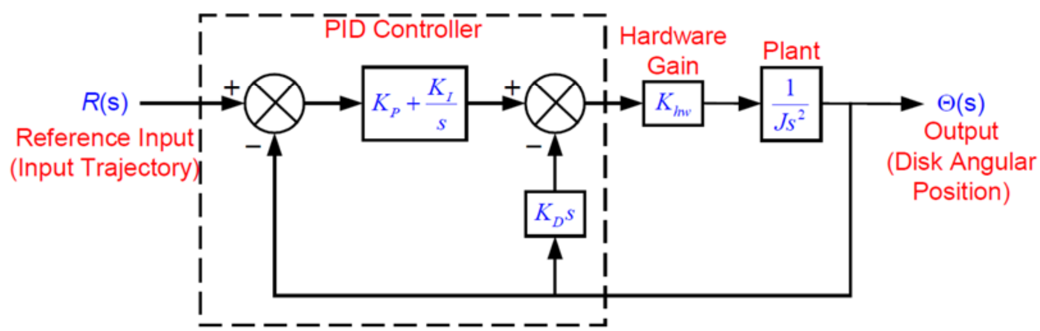
Rigid Body PD& PID A Torsional Control System

Introduction

This experiment demonstrates some key concepts associated with proportional plus derivative (PD) control and proportional plus integral plus derivative (PID) control. This type of control scheme finds broader application in industry than any other. It is employed in such diverse areas as machine tools, automobiles (cruise control), and spacecraft (attitude and gimbal control). The block diagram for forward path PID control of a rigid body is shown in *Figure 1(a)* where friction is neglected. *Figure 1(b)* shows the case where the derivative term is in the return path. Both implementations are found commonly in application and both give rise to the same characteristic roots and therefore have identical stability properties but vary in their response to dynamic inputs



(a) PID in the forward path



(b) PI in the forward path, D in the return path

Figure 1: Block diagram of rigid body control

The K_{hw} appearing in the block diagrams is referred to in the ECP manual as the "hardware gain" (in fact, it includes a software component). It accounts for DAC and amplifier gains, encoder gains, software gain factor, etc. It has units of N-m/rad; the plant gain has units of rad/N-m – position/torque, so the two combined have dimensionless gain.

For the first portion of this exercise we shall consider PD control and carry out a test to evaluate K_{hw} . We shall then include an integral term in the controller. The closed-loop transfer function for the forward path PID control (see *Figure 1(a)*) is:

$$H_{cl}(s) = \frac{\Theta(s)}{R(s)} = \frac{\left(K_P + \frac{K_I}{s} + K_D s\right) K_{hw} \left(\frac{1}{Js^2}\right)}{1 + \left(K_P + \frac{K_I}{s} + K_D s\right) K_{hw} \left(\frac{1}{Js^2}\right)} = \frac{(K_P s + K_I + K_D s^2)\left(\frac{K_{hw}}{J}\right)}{s^3 + (K_P s + K_I + K_D s^2)\left(\frac{K_{hw}}{J}\right)}$$

In this experiment, we will work with the derivative term in the return path. The closed-loop transfer function in this case (see *Figure 1(b)*) is:

$$H_{cl}(s)=\frac{\Theta(s)}{R(s)}=\frac{\left(K_P+\frac{K_I}{s}\right)K_{hw}\left(\frac{1}{Js^2}\right)}{1+\left(K_P+\frac{K_I}{s}\right)K_{hw}\left(\frac{1}{ms^2}\right)+K_{hw}\left(\frac{1}{Js^2}\right)(K_Ds)}=\frac{(K_Ps+K_I)\left(\frac{K_{hw}}{J}\right)}{s^3+(K_Ps+K_I+K_Ds^2)\left(\frac{K_{hw}}{J}\right)} \qquad (2)$$

Note that the characteristic polynomials are identical in the two cases. In the absence of the integral term (i.e. $K_I = 0$), the transfer function reduces to

$$H_{cl}(s)=\frac{\Theta(s)}{R(s)}=\frac{K_PK_{hw}\left(\frac{1}{Js^2}\right)}{1+K_PK_{hw}\left(\frac{1}{ms^2}\right)+K_{hw}\left(\frac{1}{Js^2}\right)(K_Ds)}=\frac{\left(\frac{K_PK_{hw}}{J}\right)}{s^2+\left(\frac{K_DK_{hw}}{J}\right)s+\left(\frac{K_PK_{hw}}{J}\right)} \qquad (3)$$

$$\text{i.e. } \omega_n^2=\frac{K_PK_{hw}}{J},\ \text{and } 2\zeta\omega_n=\frac{K_DK_{hw}}{J} \iff \omega_n=\sqrt{\frac{K_PK_{hw}}{J}},\ \ \zeta=\frac{K_D}{2}\sqrt{\frac{K_{hw}}{K_PJ}} \qquad (4)$$

Experimental

• **Plant set-up**
Set up the plant with two mass pieces at 9.0 cm radial center distance on the bottom disk and with the other two disks removed.

• **Proportional control and evaluation of** K_{hw}

For proportional control, $K_I = 0$ and $K_D = 0$. As equation (4) shows, ζ= 0 so no damping is provided by the control action (although, as found in Lab #1b, there will be a small amount of frictional damping which is not shown in the block diagram). From equation (4), determine the value of $K_{hw}K_p$ so that the system behaves like a 1 Hz spring-inertia oscillator (i.e., rad/sec). has an approximate value of 17.0 N-m/rad but its value varies significantly from one set of equipment to another. Assuming a value of 17.0 N-m/rad, determine the value of K_p . Now carry out a test and measure the actual frequency of oscillation using the following steps:

1. Go to ***Data/ Set-up Data Acquisition*** and set up to collect ***Encoder #1*** position and ***Commanded Position*** information. Go to ***Command/Trajectory*** and set up a closed-loop step size of 0 counts, dwell time of 5000 ms and 1 repetition.

2. Go to ***Set-up/Control Algorithm*** and set Ts = 0.0044 and select ***Continuous Time Control***. Select ***PI + Velocity Feedback*** (this is the return path derivative form shown in *Figure 1(b)*) and ***Set-up Algorithm***. Enter the Kp value determined above for 1 Hz oscillation and set K_I and K_D to zero. Do not input values greater than Kp = 0.08 . Select ***OK*** to exit.

IMPORTANT: In this and all future work, be sure to stay clear of the mechanism before doing the next step. Selecting Implement Algorithm immediately implements the specified controller; if there is an instability or large control signal, the plant may react violently. If the system appears stable after implementing the controller, first displace it with a light, non-sharp object (e.g. a plastic ruler) to verify stability prior to touching plant Select ***Implement Algorithm***, then ***OK***.

3. Go to ***Command/Execute***. Prepare to manually rotate the lower disk roughly 60 degrees. Select ***Run***, rotate the disk about 60 degrees and release it. Do not hold the rotated disk position for longer than 1-2 seconds as this may cause the motor drive thermal protection to open the control loop.

Export the data to MATLAB. Plot the encoder 1 data Calculate the frequency by using the Data Cursor Tool in the MATLAB Figure. Be sure to show the calculations and units. For system stability, do not input values greater than $k_p = 0.16$).

4. Go to ***Plotting/Set-up Plot*** and plot ***Encoder #1*** position. Determine the frequency of oscillation.

Use the measured value of frequency to find an accurate value of for the system that you are working on. Use this value for the rest of the experiment. K_{hw}
Now adjust to the correct value to give an oscillation frequency of 1 Hz and repeat the above steps to check the frequency. Kp
What do you expect to happen when the proportional gain is doubled? Verify your predictions experimentally. Again, for system stability, do not input values greater than . PK $0.08PK=$

• **Derivative control**

Determine the value of the $K_D = 0.1$ derivative gain to achieve $K_D K_{hw} = 0.1$N-m/(rad/s) . Repeat step 2 above, except input the value for K_D computed above, and set Kp as well as to zero. Do not input values greater than K_D=0.1.

After checking the system for stability by displacing it with a ruler, manually rotate the disk back and forth to feel the effect of viscous damping provided by K_D . Do not excessively coerce the disk, as this will again cause the motor drive thermal protection to open the control loop. Repeat for a value of K_D five times as large (again make sure that $K_D < 0.1$). Can you feel the increased damping?

• **PD control design**

Using equation (4), design controllers (i.e. find K_P and K_D) for three systems with the same natural frequency of 1 Hz (i.e. $\omega_n = 2\pi$ rad/sec), and damping ratios of (i) $\zeta = 0.2$ (i.e. underdamped), (ii) $\zeta = 1.0$ (i.e. critically damped), and (iii) $\zeta = 2.0$ (i.e. overdamped).

$$c(s) = \frac{k_p k_{hw}/J}{s^2 + (k_{hw}/J)(k_d s + k_p)}$$

$$\omega_n \triangleq \sqrt{\frac{k_p k_{hw}}{J}}$$

$$\zeta \triangleq \frac{k_d k_{hw}}{2J\,\omega_n} = \frac{k_d k_{hw}}{2\sqrt{J k_p k_{hw}}}$$

Step Response:

Implement the underdamped controller (via ***PI + Velocity Feedback***) and set up a trajectory for a 2500 count closed-loop ***Step*** with 2000 ms dwell time and 1 repetition (which has the effect of setting up an input shown in *Figure 2*). Execute this trajectory and plot the ***Commanded Position*** and ***Encoder #1*** position. Plot them both on the same vertical axis so that there is no graphical bias.
Repeat for the critically damped and over-damped cases. Save your plots for later comparison.

Frequency Response:

Implement the underdamped controller. Set up a trajectory for a 400 count closed-loop ***Sine Sweep*** from 0.1 Hz to 20 Hz of 60 seconds duration with ***Logarithmic Sweep*** checked (which has the effect of setting up an input shown in *Figure 2*; the signal is actually a "chirp" signal of the form $\sin(2\pi f_i)$ with $fi = f_o + 10^{\beta t}$). (You may wish to specify ***Encoder #1*** data only via ***Set-up/Data Acquisition***. This will reduce the acquired data size.) Execute the trajectory and plot the ***Encoder #1*** frequency response using ***Linear Time*** and ***Linear Amplitude*** for the horizontal and vertical axes. The data will reflect the system motion seen as the sine sweep was performed. Plot the Encoder 1 frequency response using Linear Time and Linear amplitude for the horizontal and vertical axes in MATLAB. The data will reflect the system motion seen as the sine sweep was performed.Now plot the same data using ***Logarithmic Frequency*** and ***dB Amplitude***. By considering the amplitude (the upper most portion of the data curve) you will see the data in Bode magnitude format. Can you easily identify the resonance frequency, and gain slopes in dB/decade?
Repeat for the critically damped and overdamped cases (can you easily identify the high frequency (>5 Hz) and low frequency (< 0.8 Hz)?). Save your plots for later comparison.

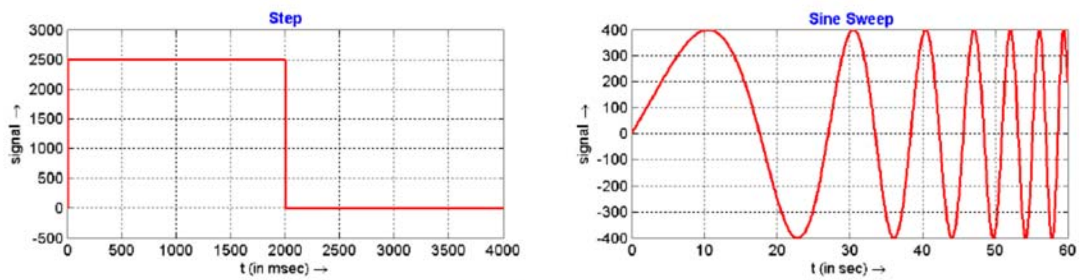


Figure 2: Signal shapes

• **Adding integral action**

Now compute K_I such that $K_I K_{hw}$ =3.0N-m/(rad-sec) . Implement a controller with this value of K_I , and values of Kp and K_D that achieved critical damping under the PD control design above. Do not input . Be certain that the following error seen in the background window is within 20 counts prior to implementing (if not, choose ***Zero Position*** from the ***Utility*** menu). Execute a 2500 count closed-loop ***Step*** of 2000 ms duration (1 repetition). Plot the ***Encoder #1*** position and the ***Commanded Position***.
Increase K_I by a factor of two, implement your controller (again make sure that K_I <0.4) and plot its step response. Manually displace the disk by roughly 5 degrees. Can you feel the integral action increasing the restoring control torque with time? (Do not hold for more than about 5 seconds to avoid excessive force build-up and hence triggering of the motor thermal protection.) What happens when you let go?

• **Modeling static friction**

Static or Coulomb friction may be modeled as some disturbance torque acting on the plant, as shown in *Figure 3*. Assume that this torque is a step function, and use the Final Value Theorem to explain the effect of such an input on the PD controlled system with and without the addition of an integral term.

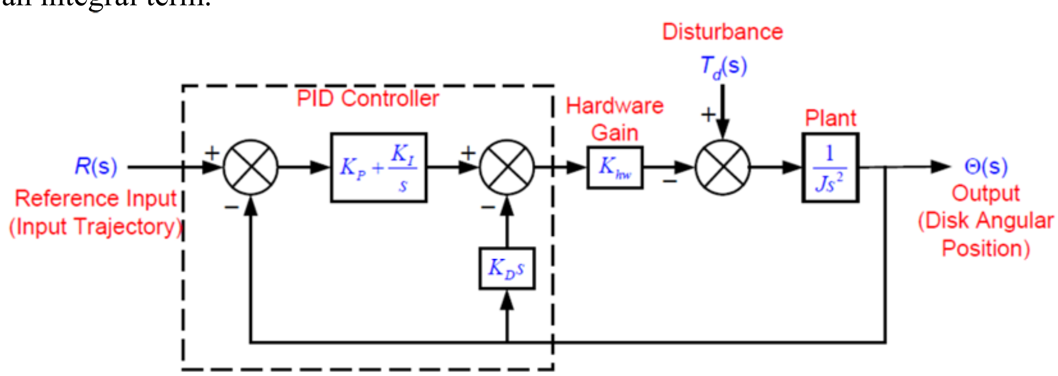


Figure 3: Friction modeled as disturbance torque on the plant

The final report have to include:

Two (6) MATLAB Plots, with two (2) Data Cursor Points on each plot, along with titles, labels and legends if necessary that clearly show which plot corresponds to which situation.
- Plot of kp
- Plot of 2kp
Calculations
- Inertia of the System J
- Calculation for kp
- Experimental frequency calculations using the kp value calculated.
- Calculations for kd
- Under-damped step response
- Critically damped step response
- Over-damped step response
Calculations .
- K_p
- Under-damped k_d
- Critically damped k_d
- Over-damped k_d
- Calculated k_i
- Experimentally "better" k_i

In your final report observe these results and explain the effects of adding integral action to the controller. Is this what you expect?

Experiment#3

Collocated PD Control With 2 DOF Plant

In this experiment we consider PD control of a 2-disk system where the controlled output, θ_1, is of the lower disk. Such a scheme is referred to as *collocated* since the sensor output is rigidly coupled to the actuator input.

The addition of the spring and second inertia increases the plant order by two and adds an oscillatory mode to the plant dynamics. This may be thought of, in a sense, as a dynamic disturbance to the rigid body plant studied in experiment 2. The collocated PD control implemented here is the approach most commonly used in industry. It may be practically employed when there is flexibility between the actuator and some inertia, and the location of objective control being near the actuator. If the location of objective control is at the distant inertia, however, this method has its limitations.

The approach in this experiment will be to design the controller by interactively changing the PD gains and observing their effect on the physical system.

Procedure :

1. Set-up the system with two masses on the upper and lower disk as shown in experiment 1, Figure 1b.
2. Implement the critically damped controller of experiment 2 being sure that encoder #1 is selected for control. Set-up data acquisition for encoders 1 & 3 (Model 205) and for commanded position and gather data every 5 servo cycles. Execute a 1000 count step response and plot the result for commanded position and encoder #1.
3. Now iteratively adjust the gains k_p & k_d and plot results to obtain an improved response. Make your gain adjustments gradually (not more than 50% at a time) and note the effects of increasing or reducing each of them. Do not input $k_p > 1$ or $0.02 < k_d > 0.2$. Attempt to achieve performance goals for the lower disk of $\leq$ 400 ms rise time (0-90% amplitude) and $\leq$ 10% overshoot in the lower disk (encoder #1) without excessive oscillation. Save your best step response plot. Manually displace the upper and lower disks and note their relative stiffness. (The lower disk stiffness is entirely due to the control system.
4. For your last iteration in Step 3, plot and save the step response of the two disks. What is the predominant characteristic of the top disk motion? Can you give an explanation for the difference in the responses of the two disks in terms of their closed loop transfer functions?
5. Now using the existing values of k_p & k_d as starting points, iteratively reduce gains and plot θ_2 results to provide a well-behaved step response with $\leq$ 10% overshoot, without excessive oscillation, and as fast a rise time as possible. Save your final plot and record the corresponding gains. Manually displace the lower and upper disks and note their stiffness. Are they generally more or less

stiff than for the controller of Step 3? How does the steady state error compare with the high gain controller from Step 3?

Questions:

A. Calculate the poles of the closed-loop transfer functions: $\theta_1(s)/r(s)$ and $\theta_2(s)/r(s)$ for your final controllers in Steps 3 & 5 respectively. How close to the imaginary axis (& right half plane) are the most lightly damped poles in each case? How close are the complex poles of $\theta_1(s)/r(s)$ to its zeros in each case? Explain your answer in terms of the root loci for this system for gain ratios of $k_d/k_p = 0.05, 0.10, 0.17,$ and 0.25.

B. Calculate the closed loop transfer function in the form:

$$\frac{\theta_2(s)}{r(s)} = \frac{(N(s))_{forward\,path} \,/\, D_{ol}(s)}{1 + N_{ol}(s)/D_{ol}(s)} \qquad (3\text{-}1)$$

Use N_{ol} and D_{ol} to obtain the open loop Nyquist or Bode responses resulting from your high and low gain controllers from Steps 3 & 5 respectively. What are the associated phase and gain stability margins? What are these margins for $\theta_1(s)/r(s)$? Explain

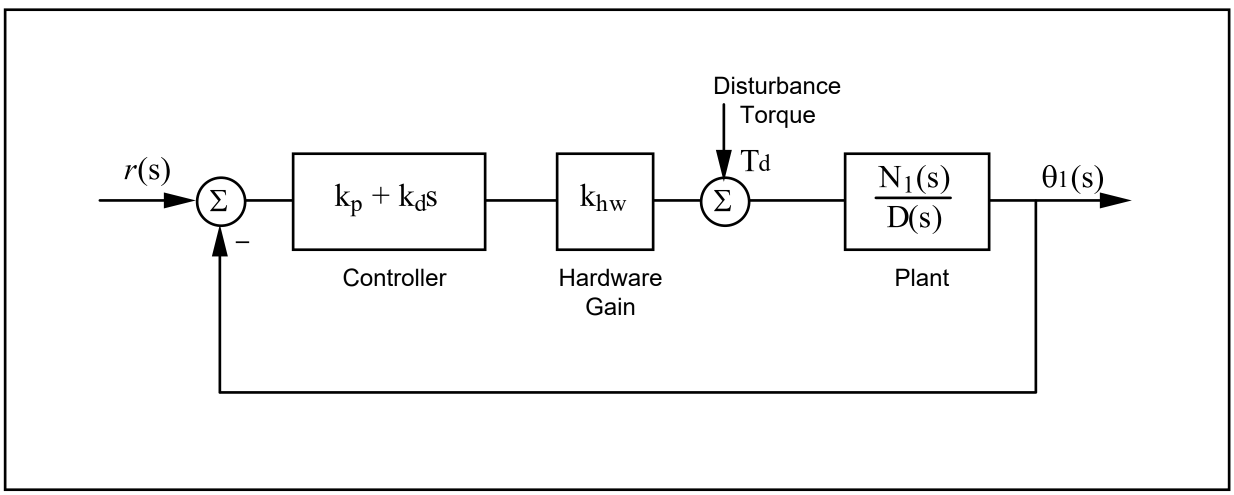


Figure 3-1. Disturbance Torques On PD Controlled 2-DOF Plant

3.5 Noncollocated PD Plus Notch Filter Control

In this experiment a control scheme is implemented which first closes the collocated loop with simple rate feedback to dampen the oscillatory mode. Then a notch filter is designed to further attenuate the transmission of signals at the damped mode frequency (i.e. nearly canceling the poles with zeros). Finally, PD control is used to achieve certain performance goals. As in the previous section, these gains are found through interactively changing their values and observing the resulting closed loop behavior.

Procedure :

1. The plant studied here is that of the two previous exercises, i.e. the one shown in Figure 3.1-1b. Consider the block diagram of Figure 3.5-1. Use root locus techniques to find the rate feedback gain, k_v, that causes the greatest damping in the complex roots of the inner loop $\theta_1(s)/r^*(s)$.
2. Implement this gain as the f1 coefficient in the General Form controller specification box. Be sure that you select encoder #1 for feedback loop #3 before exiting the box and implementing. Attempt to manually excite the oscillatory mode via the upper disk and notice the damping effect of velocity feedback.

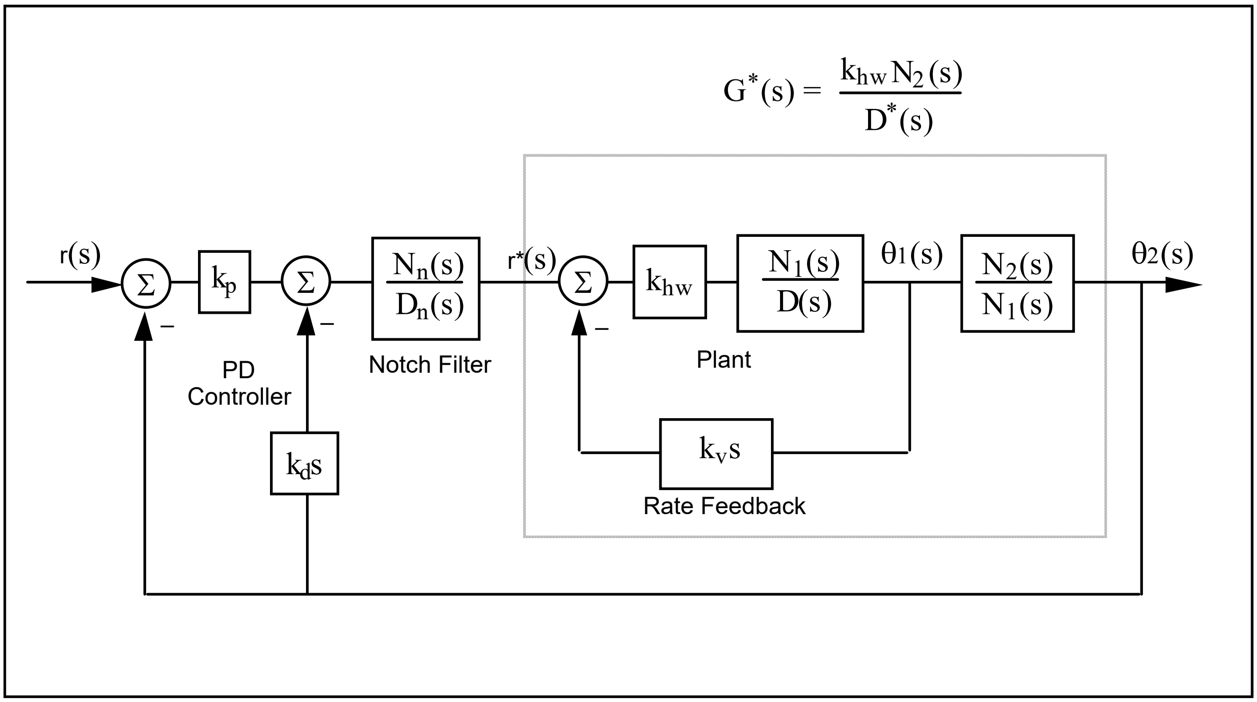


Figure 3.5-1. Control Structure

3. We may now design for the new "plant" $G^*(s)$ where $N_2(s)$ is as before, and:

$$D^*(s) = D(s) + k_v s\, N_1(s) \qquad (3.5\text{-}1)$$

Design a notch filter, $N_n(s)/D_n(s)$ with two poles at 10.0 Hz and 70.7% damping ($\zeta=\sqrt{2}$), two poles at 40.0 Hz and 70.7% damping and with two zeros at the poles of $D^*(s)$. Utilize a monic form of the denominator (highest order term in $D_n(s)$ has coefficient of 1) and normalize the notch filter transfer function to have unity DC gain.[3]

4. Select some initial PD gains for control of θ_2. Use values of $k_p = 0.01$, $k_d = 0.001$ initially. Do not exceed $k_p = 0.1$ and $k_d = 0.02$ in any subsequent run. Since this control scheme uses multiple loop feedback, the Executive program requires that it be implemented via the general control algorithm form. A short-cut to controller coefficient entry is as follows: Input your notch filter coefficients and proportional controller gain, k_p, via the dialog box PID + Notch. Be certain to enter these coefficients in the proper order and to high numerical precision (e.g. 8 decimal places). Exit and select Preview In General Form. You will see the P + notch controller in the form that it is implemented in the generalized controller form.[4]

   In the General Form window, enter the f1 coefficient calculated in Step 1 (i.e. k_v), and enter as i1 the desired derivative gain, k_d. Make sure that Encoder #1 is selected for Loop #3 and Encoder #3 is selected for Loops #1&2 (the above for Model 205a, for Model 205 select Encoder #2 for Loop #1). You should verify that these entries appropriately represent the control structure of the controller in Figure 6.5-1. Exit the box and make sure that General Form is selected; then Implement. Now check performance of the control using step trajectories of 1000 counts.

5. Iteratively change PD gains by repeating Step 4 to obtain performance goals of 0.4 sec rise time (0-90% of final amplitude) and overshoot less than 15%. Do not exceed k_p=0.10 or k_d=0.01. You may notice a trend here that for relatively high values of k_d the system transmits excessive noise and may appear "twitchy". As k_p becomes large, the system becomes oscillatory and further increases lead to instability. Does increase in noncollocated derivative action necessarily reduce the step response oscillations?

6. Record your best performance step response. Manually displace the disks and note the relative static stiffness of the upper disk under this control.

---

[3]For discrete time design, convert your notch filter design to the z domain using the Tustin (bilinear) transformation:

$$s = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}$$

[4]$r(s)$ will be the same as $D_n(s)$ and s(s) & $t(s)$ will be identical and equal to $k_pN_n(s)$. Upon selecting Preview In General Form the algorithm is also converted to discrete time form (by the Tustin transform for N_n and D_n, backwards difference for any derivative terms). In this way the discrete time equivalent controller may be viewed in the discrete time General Form controller box.

Questions:

A. Report your calculated values for k_v, $N_n(s)$, $D_n(s)$, and your selected values of k_p, k_d. Submit your step response plot. Does it meet the performance goals of Step 5?

B. Calculate the closed-loop transfer function $\theta_2(s)/r(s)$ including all elements in the block diagram of Figure 6.5-1. You may express this in terms of the polynomials $D(s)$, $N_1(s)$, $N_n(s)$, etc. rather than expanding each term fully. Use the equation

$$\frac{\theta_2(s)}{r(s)} = \frac{(N(s))_{forward\,path} \,/\, D_{ol}(s)}{1 + N_{ol}(s)/D_{ol}(s)} \qquad (3.5\text{-}2)$$

and determine the phase and gain margins of your system design via Bode or Nyquist plots. What would happen if you doubled your gains k_p and k_d?. What if you quadrupled them?

Experiment 4

Successive Loop Closure / Pole Placement Design For 2 DOF Plant

In this experiment we first close a position loop about the collocated (θ_1) position with a relatively high bandwidth (close tracking) PD control. We then make the assumption that the lower disk closely follows its internal demand $r^*(s)$ so that for designing a controller for θ_2, the "plant" is approximated by the transfer function $\theta_2(s)/\theta_1(s)$ (i.e. $N_2/N_1(s)$). The block diagram for this approach is given in Figure -1.
The design and control implementation in this section proceeds as follows

1. High bandwidth PD control of θ_1
2. Low pass filter augmentation to attenuate signal noise due to high PD gains
3. Outer loop control of θ_2 via pole placement methodology

4.6.1 PD Control Of The Lower Disk

1. Setup the plant in the configuration of Figure 1. in experiment 1.

PD Control Design & Implementation

1. design the PD control gains such that ω_n = 10 Hz, and ζ = 0.707 when considering only J_1 acting as a rigid body.
2. Set the sampling rate to T_s = 0.002652 seconds and implement your gains via the PID control algorithm box. You should notice some system noise associated with the high derivative gain term. Discontinue the control via Abort Control.

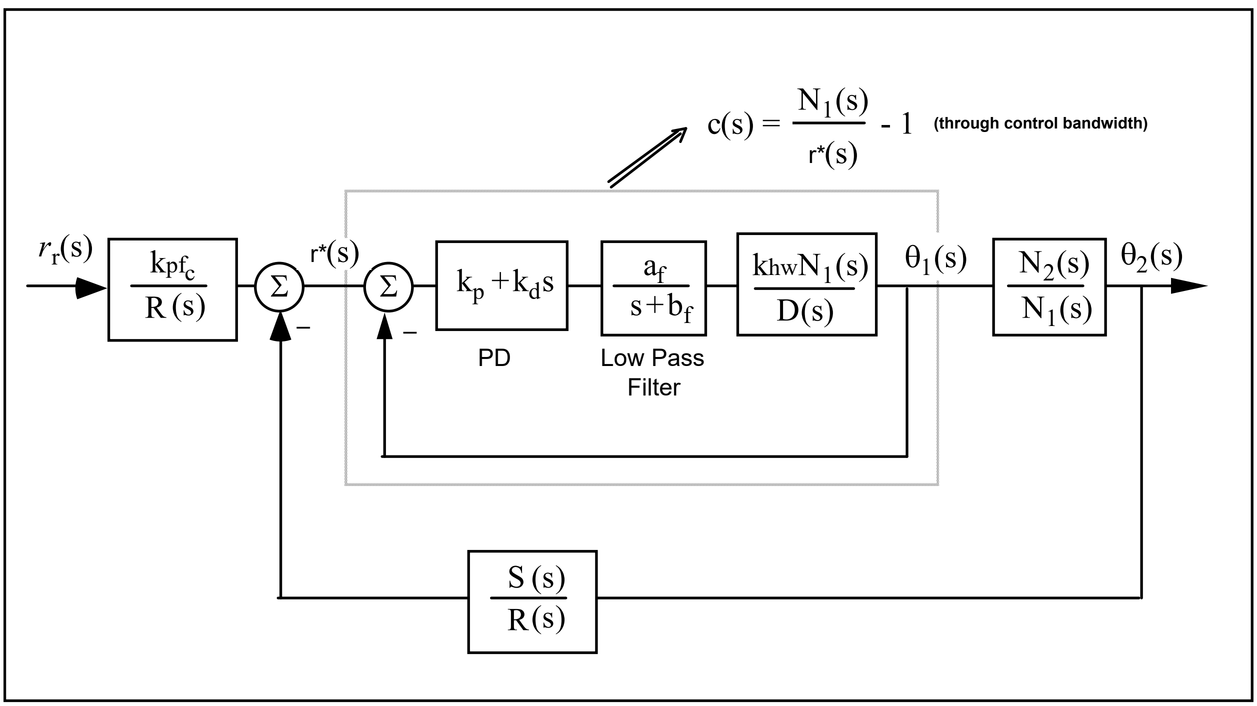


Figure -1. Control Structure For Successive Loop Closure with High Gain Inner Loop

Low Pass Filter Design & Implementation

3. Solve for the constants a_f, and b_f such the filter has a pole at s = -240 (approximately 40 Hz) and has unity DC gain.
4. Calculate the numerator and denominator of the controller associated with the cascade of the PD and low pass filter blocks. These will have the form

$$\text{PD*Filter} = \frac{e_0 + e_0 s}{g_0 + g_1 s} \qquad (1)$$

5. In the Generalized Form box, enter the following.
   a. Enter your calculated e_i's and g_i's
   b. Select Encoder #1 for Loop #2. (the other loops may have any encoder selected at this point)
   c. Set the following equal to 1: t_0, h_0, i_0,
   d. Verify that r_1 = 0.0000002[5]
   e. Verify that all other coefficients = 0

   It is important to take care and assure that all parameters are properly set before leaving the Generalized Form dialog box and implementing control.
6. Exit the dialog box, verify that the Generalized Form button is selected, and Implement control. You should notice a reduction in audible signal noise. Safety check the controller and manually displace the disks. They should behave as before in Step 2.

4.2 Pole Placement Control of $\theta_2(s)/\theta_1(s)$

Having closed a relatively high bandwidth (≈10 Hz) loop about the first disk, we utilize the fact that the transfer function of Eq.

$$c(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

has near unity input/output gain (and near zero phase) through the bandwidth (≈ 2.5 Hz) that we will attempt to attain in the overall control of θ_2. Thus for the control of θ_2 we consider the outer loop in the block diagram of Figure -1.

[5] This small value is needed to provide a proper transfer function for bilinear transformation and subsequent discrete control implementation whenever T/R and S/R are used to implement a differentiator. Its small value results in a pole many decades beyond the system bandwidth and is of no practical implication to system modeling or performance. This coefficient may be set to zero here, but should generally remain.

Now the plant to be controlled is:

$$\frac{\theta_2(s)}{\theta_1(s)} = \frac{N_2}{N_1} \triangleq \frac{N^*(s)}{D^*(s)} \tag{2}$$

The numerical values of the parameters in this expression were determined in Experiment #1.

We now seek to find a controller *S(s)/R(s)* which will result in a prescribed set of closed loop poles. The closed loop denominator will have the form:

$$D_{cl}(s) = D^*(s)R(s) + N^*(s)S(s) \tag{3}$$

which may be expressed as[6]

$$D_{cl}(s) = (d_2 s^2 + d_1 s + d_0)(r_1 s + r_0) + (n_0)(s_1 s + s_0) \tag{4}$$

where the d_i's and n_i's are the respective coefficients of the denominator and numerator of the right hand side of Eq. (2).

By linear system theory, for coprime $N^*(s)$, $D^*(s)$ with $N^*(s)/D^*(s)$ proper, there exists an $(n-1)^{th}$ order *S(s)*, *R(s)* which when convolved as form an arbitrary $(2n-1)^{th}$ order $D_{cl}(s)$ where n is the order of $D^*(s)$.

Here we shall solve for the desired denominator:

$$D_{cl}(s) = \left(s + 5\pi\frac{(1+j)}{\sqrt{2}}\right)\left(s + 5\pi\frac{(1-j)}{\sqrt{2}}\right)(s + 5\pi) \tag{5}$$

I.e. closed loop poles at -2.5, and $-1.77 \pm j1.77$Hz.[7]

Pole Placement Design

7. Determine the coefficients of the controller polynomials *S(s)* and *R(s)* by equating coefficients in the expanded forms of Eq's 4 and 5.

---

[6]The notation here is the obvious one.

[7] This has poles of magnitude |s| = 2.5 Hz that lie at 135, 180, and 225 deg. It is similar to a third order Butterworth polynomial but somewhat more damped.

8. Calculate the scalar prefilter gain k_{pf} by referring to Figure 1. The goal is to have the output $\theta_2(s)$ scaled equal to the input $r_r(s)$. Hint: Consider the system in static equilibrium. Set $\theta_2 = 1$ and $r_r = 1$ and solve for k_{pf} using only the constant terms in all control blocks.

Control Implementation

9. In the Executive program, set-up to collect Encoder #1, Encoder #3 and Commanded Position information[8] via Set-up Data Acquisition with data sampling every two sample periods. Setup a closed loop step trajectory of 1000 counts, 2000 ms duration and 1 repetition.

10. Return to the General Form Algorithm box and verify that the PD controller, low pass filter and all other coefficients are entered as given in Step 5 above. Enter the coefficients for $S(s)$ and $R(s)$ determined in Step 1. Enter the value k_{pf} calculated in Step 8 as to. Make sure that Encoder #3 is selected for Loop #1 (for Model 205a, Encoder #2 for Model 205) and Encoder #1 for Loop #2. Select OK to exit to the controller selection dialog box.

11. <u>While staying clear of the mechanism</u> select Implement Algorithm. If the mechanism reacts violently you have implemented an unstable controller or otherwise improperly entered the control coefficients and you will need to repeat the above steps as appropriate. You should first Reset Controller (Utilities menu) before attempting to re-implement control. If the system is well behaved, and <u>after safety checking the controller</u>, you may disturb the upper and lower disks lightly. Notice the relative stiffness of the two disks and how the lower disk moves in opposition to (i.e. regulation of) disturbances of the upper disk.

    You may notice some "twitching" or buzzing due to noise in the system. This often occurs in such high gain systems, but has been mitigated via the low pass filter. <u>If the noise is excessive or there is any possibility that the equipment is at risk discontinue control immediately.</u>

12. Execute the Step input previously programmed, and plot the Encoder 1, Encoder #2, and Commanded Position data. Save your plot. How does the response at θ_2 compare with designs previously tested? Describe the motion of θ_1 and how it shapes the response at θ_2.

A. Report your calculated values for k_p, k_d, a_f, b_f, e_0, e_1, g_0, g_1, s_0, s_1, r_0, r_1, and k_{pf}

---

[8] You may also select Control Effort if you wish to later observe this value .

Experiment 5

LQR Control Design For 2 DOF Plant

Dr. Rad
Revise 6/2016

In this experiment a linear quadratic regulator is implemented using full state feedback. The plant used is again that of Figure 6.1-1b. The states chosen are the disk angles and rates according to the model of Eq (5.1-3) with the output taken as θ_2, i.e.

$$C = [\,0 \;\; 0 \;\; 1 \;\; 0\,] \tag{1}$$

<u>LQR Design:</u>

1. Construct a state space model of the plant using the realization of Eq (4.1-3) and measured parameter values
2. The following notation shall be used for LQ optimization:

   Feedback law:

   $$u = -Kx \tag{2}$$

   where

   $$K = [\; K_1 \quad K_2 \quad K_3 \quad K_4 \;] \tag{3}$$

   Perform LQR synthesis via the Riccati equation solution[9] or numerical synthesis algorithms to find the controller K which minimizes the cost function (scalar control effort):

   $$J = \int \left(x' Qx + u^2 r\right) dt \tag{4}$$

   In this synthesis choose Q=C'C so that the error at the intended output, θ_2, is minimized subject to the control effort cost. Perform synthesis for control effort weight values: r = 100, 10, 1.0, 0.01, and 0.001. Calculate the closed loop poles for each case as the eigenvalues of [A–BK]

3) From this data, select a control effort weight to put the lowest pole frequency between 2.75 and 3.25 Hz. Use one of the above obtained K values if it meets this criteria, or interpolate between the appropriate r values and perform one last synthesis iteration. Do not use K_1 or K_3 values greater than 1, or K_2 or K_4 values greater than 0.12.[10]

---

[9]See for example Kwakernaak and Sivan, "Linear Optimal Control Systems", Wiley & Sons, 1972.

[10]K_1 and K_3 scale control effort proportional to position errors, K_2 and K_4 scale control effort proportional to the respective velocities. Excessive values of K_1 or K_3 can lead to low stability margin and in the presence of time delays, instability. Large K_2 or K_4 cause excessive noise propagation and lead to "twitching" of the system. - see Section 6.8

Control Implementation

Important Note: For Model 205a, (3 encoders), θ_2 is measured via Encoder #3. The design gains $K_2 + K_3$ are therefore input as k_5 and k_6 in the Executive's State Feedback box.

4) Implement your controller via the State Feedback box under Control Algorithm.[11] For tracking, the prefilter gain K_{pf} must be set equal to $K_1 + K_3$. You may wish to select Preview In General Form to see how these parameters are mapped into the generalized algorithm.

5) Execute a 1000 count step and plot the result. How do the rise time, overshoot, servo stiffness, and steady state errors compare with previous controllers?

Questions / Exercises:

A. Report your calculated values of the closed loop poles for the various values of r in Step 2 and for your final design. Report the values of K, for your final design.

B. Calculate the closed-loop transfer function $\theta_2(s)/r_r(s)$ including all elements in the block diagram of Figure 1b (lower subfigure). You may express this in terms of the polynomials $D(s)$, $N_1(s)$, N_2, and K_i (i = 1,2,3,4). Determine the phase and gain margins for the system by considering the open loop "numerator" to be all terms in the closed loop denominator except $D(s)$[12].

Questions / Exercises:

A. Report your calculated values of the closed loop poles for the various values of r in Step 2 and for your final design. Report the values of K, for your final design.

B. Calculate the closed-loop transfer function $\theta_2(s)/r_r(s)$ including all elements in the block diagram of Figure 1b (lower subfigure). You may express this in terms of the polynomials $D(s)$, $N_1(s)$, N_2, and K_i (i = 1,2,3,4). Determine the phase and gain margins for the system by considering the open loop "numerator" to be all terms in the closed loop denominator except $D(s)$[13].

[11] If using discrete time implementation, be sure to divide your K_2 and K_4 values by T_s before entering them.

[12] This yields the margins as if the additional phase lag or gain were to occur in the control input (i.e. k_{hw} block), or uniformly among the outputs. This is not necessarily the case in practice, but does provide a general measure of stability margin.

[13] This yields the margins as if the additional phase lag or gain were to occur in the control input (i.e. k_{hw} block), or uniformly among the outputs. This is not necessarily the case in practice, but does provide a general measure of stability margin.

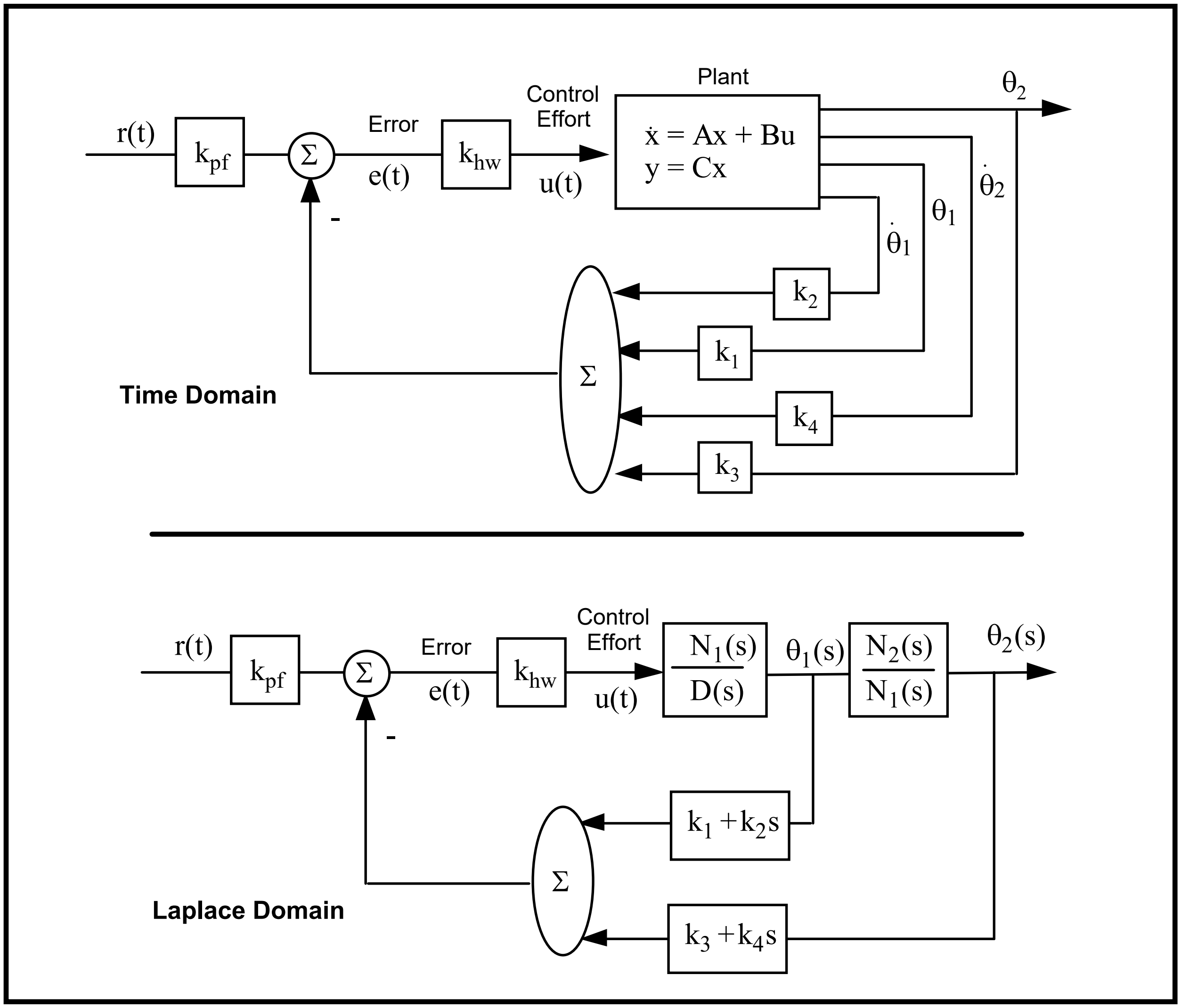
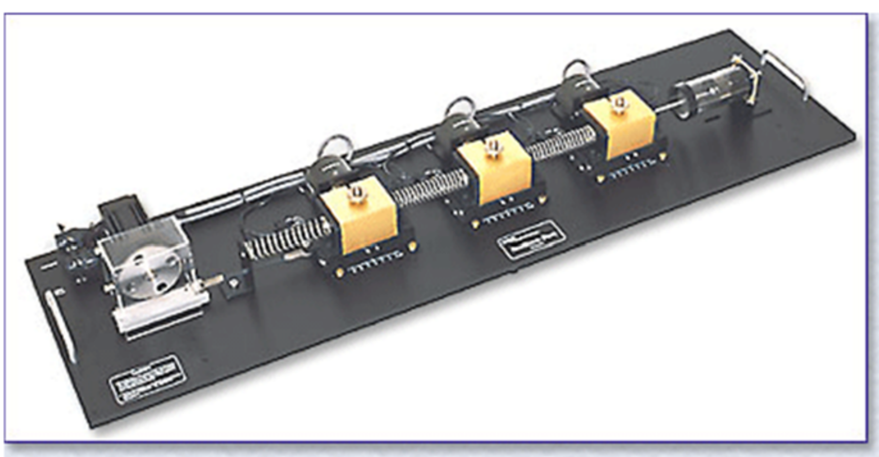


Figure -1. Representations of the Full State Feedback System

M210

Rectilinear Control System



Dr. Rad
Revise 6/2016

Experiment#6

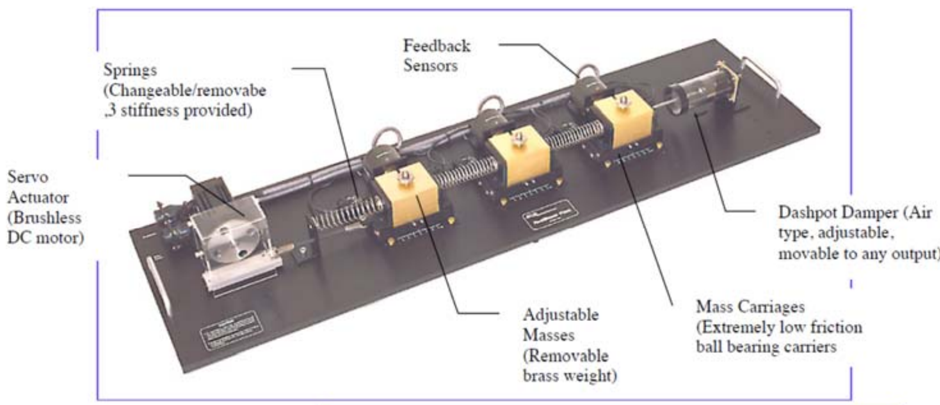
System Identification of a Linear Plant Control System

Introduction

This document describes experiments to be carried out on the ECP Systems Model 210a Linear Plant. Much of the work is concerned with ***system identification***, i.e. determination of parameter values of models of the system. Part of the work involves using the values obtained in a Matlab simulation.

To become familiar with the operation of the equipment, it would be useful for any user to read chapter 1 of the manual in its entirety prior to undertaking the operations described here. Lack of time may preclude a detailed study of the manual, but, in any case, the safety portion of the manual, Section 1.3, must be read and understood prior to operating the equipment.

The procedures described here for identifying plant parameters use certain fundamental properties of lightly damped second order systems to indirectly measure the mass, spring constant and damping coefficients of the plant by making measurements with the plant set up in a pair of classical spring-mass configurations.



Model 210A Rectilinear Plant

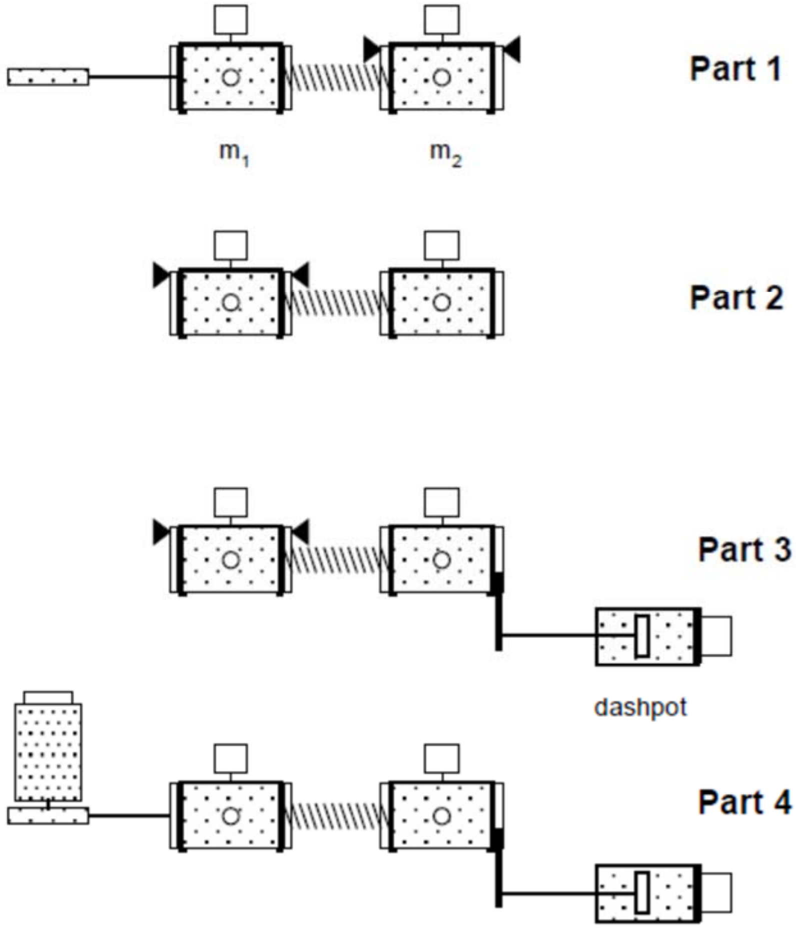


Figure 1: Test configurations

1. Background Theory

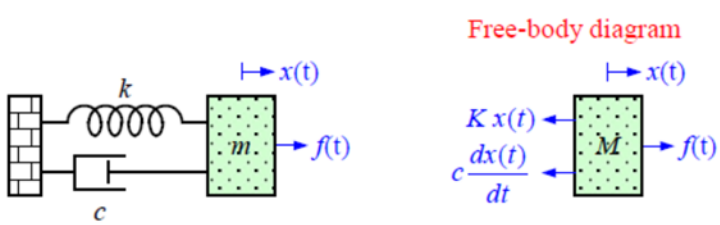


Figure 3: Single mass system

This Experiment has no

external force, i.e. $f(t)=0$. The position of the mass is given by $x(t)$ (with $x=0$ being the equilibrium position, when $f=0$). k and c are the spring constant and the viscous damping coefficient respectively. The differential equation of motion is easily seen from the Free-body diagram to be

$$f(t)-c\frac{dx(t)}{dt}-k\,x(t)=m\frac{d^2x(t)}{dt^2} \tag{1}$$

We are exciting the system by displacing it from its equilibrium position, i.e. we have (nonzero) initial conditions $x(0)=x_0$ and $\left[\frac{dx(t)}{dt}\right]_{t=0}=0$. The motion for $t>0$ can be found by taking the Laplace transform (see lecture slide L1.18) of equation (1)

$$F(s)-c\left(sX(s)-x(0)\right)-k\,X(s)=m\left(s^2X(s)-s\,x(0)-\left[\frac{dx(t)}{dt}\right]_{t=0}\right)$$

$$\Rightarrow X(s)=\frac{(m\,s+c)x_0}{m\,s^2+c\,s+k}=\frac{(s+2\zeta\omega_n)x_0}{(s^2+2\zeta\omega_n s+\omega_n^2)} \tag{2}$$

$$\text{where } \omega_n^2=\frac{k}{m},\ \text{and } \zeta=\frac{c}{2m\,\omega_n} \tag{3}$$

$$\Rightarrow X(s)=\frac{(s+\sigma_d)\,x_0}{(s+\sigma_d)^2+\omega_d^2}+\frac{\sigma_d\,x_0}{(s+\sigma_d)^2+\omega_d^2}\leftrightarrow x(t)=e^{-\sigma_d t}\cos(\omega_d t)\,x_0+\left(\frac{\sigma_d}{\omega_d}\right)e^{-\sigma_d t}\sin(\omega_d t)\,x_0 \tag{4}$$

$$\text{where } \quad \sigma_d=\zeta\omega_n,\omega_d=\sqrt{1-\zeta^2}\,\omega_n \tag{5}$$

The peaks of the waveform and its locations can be found by solving for

$$\frac{dx(t)}{dt}=-\sigma_d\,e^{-\sigma_d t}\cos(\omega_d t)\,x_0-\omega_d\,e^{-\sigma_d t}\sin(\omega_d t)\,x_0-\left(\frac{\sigma_d^2}{\omega_d}\right)e^{-\sigma_d t}\sin(\omega_d t)\,x_0+\sigma_d\,e^{-\sigma_d t}\cos(\omega_d t)\,x_0=0$$

$$\frac{dx(t)}{dt}=0 \ \text{ whenever } \sin(\omega_d t)=0\leftrightarrow t=\frac{\pi}{\omega_d},\frac{2\pi}{\omega_d},\frac{3\pi}{\omega_d},\cdots$$

If $x_0>0$, then $t=\frac{2\pi}{\omega_d},\frac{4\pi}{\omega_d},\frac{6\pi}{\omega_d},\cdots$ correspond to maxima in the waveform (and $t=\frac{\pi}{\omega_d},\frac{3\pi}{\omega_d},\frac{5\pi}{\omega_d},\cdots$ correspond to minima). So the damped frequency ω_d is indeed the frequency of oscillation and can be obtained from the location of successive peaks in the waveform. Further, these peaks are given by

$$x\left(\frac{2n\pi}{\omega_d}\right)=e^{-2n\pi\sigma_d/\omega_d}\cos(2n\pi)\,x_0+\left(\frac{\sigma_d}{\omega_d}\right)e^{-2n\pi\sigma_d/\omega_d}\sin(2n\pi)\,x_0$$

$$\Rightarrow x\left(\frac{2n\pi}{\omega_d}\right)=e^{-2n\pi\sigma_d/\omega_d}\,x_0 \tag{6}$$

So the ratio of 1st peak to the (n+1)st peak (which occurs $(2n\pi/\omega_d)$ seconds later) can be used to obtain the damping ratio as follows

$$\Rightarrow\frac{1^{\text{st}}\text{ peak}}{(\text{n+1})^{\text{st}}\text{ peak}}=\frac{e^{-2\pi\sigma_d/\omega_d}\,x_0}{e^{-2(n+1)\pi\sigma_d/\omega_d}\,x_0}=e^{2n\pi\sigma_d/\omega_d}=e^{2n\pi\zeta/\sqrt{1-\zeta^2}}\leftrightarrow\frac{\zeta}{\sqrt{1-\zeta^2}}=\frac{1}{2n\pi}\ln\left(\frac{1^{\text{st}}\text{ peak}}{(\text{n+1})^{\text{st}}\text{ peak}}\right) \tag{7}$$

which for small ζ can be approximated as $\zeta=\frac{1}{2n\pi}\ln\left(\frac{1^{\text{st}}\text{ peak}}{(\text{n+1})^{\text{st}}\text{ peak}}\right)$.

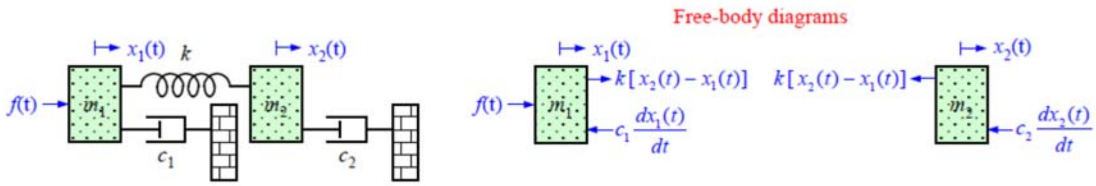


Figure 4: Two mass system

$$f(t)-c_1\frac{dx_1(t)}{dt}+k[x_2(t)-x_1(t)]=m_1\frac{d^2x_1(t)}{dt^2} \tag{8}$$

$$-c_2\frac{dx_2(t)}{dt}-k[x_2(t)-x_1(t)]=m_2\frac{d^2x_2(t)}{dt^2} \tag{9}$$

We can take their Laplace Transform (assuming zero initial conditions) and solve as follows

$$\begin{bmatrix} m_1s^2+c_1s+k & -k \\ -k & m_2s^2+c_2s+k \end{bmatrix}\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix}=\begin{bmatrix} 1 \\ 0 \end{bmatrix}F(s)$$

$$\Rightarrow\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix}=\begin{bmatrix} m_1s^2+c_1s+k & -k \\ -k & m_2s^2+c_2s+k \end{bmatrix}^{-1}\begin{bmatrix} 1 \\ 0 \end{bmatrix}F(s)=\frac{\begin{bmatrix} m_2s^2+c_2s+k & k \\ k & m_1s^2+c_1s+k \end{bmatrix}}{(m_1s^2+c_1s+k)(m_2s^2+c_2s+k)-k^2}\begin{bmatrix} 1 \\ 0 \end{bmatrix}F(s)$$

From this, we get the two transfer functions of interest given below

$$\Rightarrow G_1(s)=\frac{X_1(s)}{F(s)}=\frac{m_2s^2+c_2s+k}{m_1m_2s^4+(c_1m_2+c_2m_1)s^3+(k\,m_1+k\,m_2+c_1c_2)s^2+(k\,c_1+k\,c_2)s} \tag{10}$$

$$\Rightarrow G_2(s)=\frac{X_2(s)}{F(s)}=\frac{k}{m_1m_2s^4+(c_1m_2+c_2m_1)s^3+(k\,m_1+k\,m_2+c_1c_2)s^2+(k\,c_1+k\,c_2)s} \tag{11}$$

The transfer functions relating velocity to force (if desired) are obtained by multiplying by s.

Experimental work

Obtain a set of masses, springs, tools and the lab manual from your tutor. (Make sure you return all of these at the end of the session)
The ECP equipment has three mass carriages. For this experiment, only the first two (closest to the drive) are used. Ensure that the third mass carriage is clamped securely and will not interfere with the other equipment. Check also that the medium stiffness spring (nominally 400 N/m) is connecting the first and second mass carriages.
There are four parts to the experiment (*Figure 1* shows the corresponding configurations):
1. parameters for mass 1,
2. parameters for mass 2,
3. parameters for mass 2 with dashpot connected, and
4. test on a two mass system.

It is recommended that you save data and control configuration files regularly to avoid work loss should a system fault occur.

• **Part 1 - Parameters for mass 1**

Carry out the following steps to determine the mass and coefficient of friction for the first mass carriage and the spring constant:

1. Clamp the second mass to put the mechanism in the first of the configurations in *Figure 1* using a shim (e.g. 1/4 inch nut) between the stop tab and stop bumper so as not to engage the limit switch
2. Secure four 500g masses on the first mass carriage.

3. With the controller powered up, go to ***Set-up/Control Algorithm*** and set Ts =0044 . Go to ***Command /Trajectory*** and select ***Step, Set-up***. Select ***Open Loop Step*** and input a step size of zero, duration of 3000 ms and 1 repetition. This puts the controller in a mode for acquiring 6 sec of data on command but without driving the actuator. Exit to the background screen by consecutively selecting ***OK***. This procedure may be repeated later to vary the data acquisition period.

4. Go to ***Data/Set up Data Acquisition*** and ensure that ***Encoder #1*** is selected as data to acquire and specify data sampling every 2 servo cycles (i.e. every 2Ts). Select ***OK*** to exit.

   Select ***Utility /Zero Position*** to zero encoder positions.

5. Go to ***Command /Execute***. Prepare to manually displace the first mass carriage approximately 2.5 cm. Exercise caution in displacing the carriage so as not to engage the travel limit switch. With the first mass displaced approximately 2.5 cm in either direction, select ***Run*** from the ***Execute*** box and release the mass approximately 1 second later. The mass will oscillate and then settle while encoder data is collected to record the response. Select ***OK*** after the data is uploaded.

6. Go to ***Plotting /Set-up Plot*** and choose ***Encoder #1 Position*** then select ***Plot Data*** from the ***Plotting*** menu. You will see the time response.

7. The damped frequency can be obtained by measuring the interval between successive peaks of the waveform (This is valid for at least the first few large peaks. Smaller peaks later are dominated by nonlinear friction effects and do not reflect the salient system dynamics). Note that we are using "m11" as a subscript to denote mass #1, trial #1. You may “zoom” the plot via ***Axis Scaling*** for more precise measurement. Similarly, the damping ratio ζ_{m11}can be obtained by comparing heights of successive peaks. Natural frequency ω_{nm11}can be obtained fromω_{dm11} and ζ_{m11}

$$\omega_{dm11} = \sqrt{1 - \zeta_{m11}^2}\ \omega_{nm11}$$

   Close the graph window by clicking on the left button in the upper right hand corner of the graph. This will collapse the graph to an icon, from where it may later be brought back up.

8. Remove the four masses from the first mass carriage and repeat steps 3 through 7 to obtain ω_{nm12} , ζ_{m12} , and ω_{nm12} for the unloaded carriage. If necessary, in step 3, adjust the execution (data sampling only in this case)

9. Each brass weight has a mass of 500 ± 10 g. Calling the mass of the four weights combined m_w , use the following relationships to solve for the unloaded carriage mass m_{c1} , and spring constant : k

$$\frac{k}{m_w + m_{c1}} = \omega_{nm11}^2 \quad \text{and} \quad \frac{k}{m_{c1}} = \omega_{nm12}^2$$

Note that the calculated mass will include the reflected inertias of all connected elements such as motor pinion and armature. Finally, the damping coefficient can be found by using .

$$\zeta_{m12} = \frac{c_{m1}}{2 m_{c1}\, \omega_{nm12}} \, .$$

• **Part 2 - Parameters for mass 2**

The same steps as above should now be carried out for mass carriage 2, with the necessary adjustments. Mass carriage 2 will be unclamped and carriage 1 will be clamped as shown in the configuration diagram (see *Figure 1*). Disconnect the dashpot extension bracket from carriage 2. Select encoder 2 for data acquisition instead of encoder 1, etc.

Again by measuring ω_{dm21}, ζ_{m21}, ω_{dm22}, and, ζ_{m22}, obtain m_{c2},C_{m2} and k . The value obtained for should be close to that obtained in Part 1. In later work, you may use the average of the two for your identified value.

• **Part 3 – Coefficient of friction for mass 2 with dashpot**

Connect the mass carriage extension bracket and dashpot to the second mass as shown in the configuration diagram (see *Figure 1*). Open the damping (air flow) adjustment knob 2 turns (anti-clockwise) from the fully closed position.
Repeat the relevant steps of Part 2 with four 500 g masses on the second carriage. You may find it convenient to reduce the step duration to, say, 1000 ms.
Hence obtain ζ_d where the "d" subscript denotes "dashpot". Note that the damping ratio that you measure is the sum of ζ_d and ζ_{m2} that you obtained in Part 2.

• **Part 4 – Test of two mass system**

1. Secure four weights to each of carriages 1 and 2 and unclamp both carriages. The dashpot should be connected to carriage 2 at the same setting as in Part 3. Check that the drive mechanism is connected.

2. In the ***Command /Trajectory*** window deselect ***Unidirectional moves*** (thereby enabling bi-directional inputs) and select ***Step, Set-up***. Choose ***Open Loop Step***, and input 2.00 Volts*,* 75 ms*,* 2 reps. ("Step" actually means a square pulse, see p19 of the manual). This has the effect of setting up the following input:

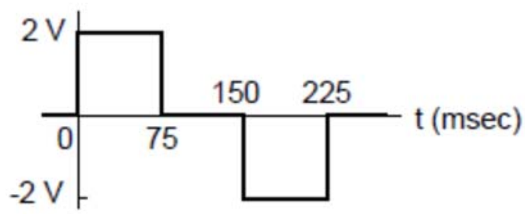


Figure 2: Test input

3. Carry out a run via ***Command /Execute*** (remember to tick ***Extended Data Sampling***). This move may trip a software speed limit or motion travel limit that disables the controller, indicated by ***"Limit Exceeded"*** in the ***Controller Status*** box in the "desk top". To reset, simply reselect ***Execute*** from the ***Execute*** menu.

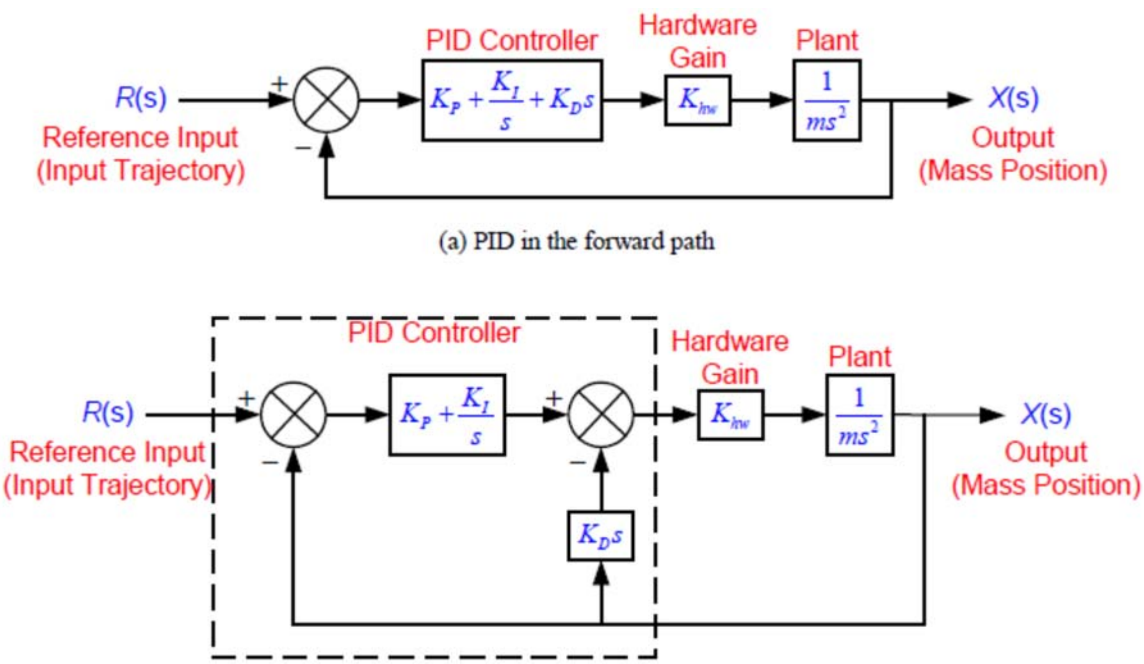
Experiment#7

PD and PID- Linear Plant

Introduction

This experiment demonstrates some key concepts associated with proportional plus derivative (PD) control and proportional plus integral plus derivative (PID) control. This type of control scheme finds broader application in industry than any other. It is employed in such diverse areas as machine tools, automobiles (cruise control), and spacecraft (attitude and gimbal control).

The block diagram for forward path PID control of a rigid body is shown in *Figure 1(a)* where friction is neglected. *Figure 1(b)* shows the case where the derivative term is in the return path. Both implementations are found commonly in application and both give rise to the same characteristic roots and therefore have identical stability properties but vary in their response to dynamic inputs



(a) PID in the forward path

(b) PI in the forward path, D in the return path

Figure 1: Block diagram of rigid body control

The K_{hw} appearing in the block diagrams is referred to in the ECP manual as the "hardware gain" (in fact, it includes a software component). It accounts for DAC and amplifier gains, encoder gains, software gain factor, etc. It has units of N/m; the plant gain has units of m/N – position/force, so the two combined have dimensionless gain

For the first portion of this exercise we shall consider PD control and carry out a test to evaluate K_{hw}. We shall then include an integral term in the controller. The closed-loop transfer function for the forward path PID control (see *Figure 1(a)*) is:

$$H_{cl}(s) = \frac{X(s)}{R(s)} = \frac{\left(K_P + \frac{K_I}{s} + K_D s\right) K_{hw} \left(\frac{1}{m s^2}\right)}{1 + \left(K_P + \frac{K_I}{s} + K_D s\right) K_{hw} \left(\frac{1}{m s^2}\right)} = \frac{(K_P s + K_I + K_D s^2)\left(\frac{K_{hw}}{m}\right)}{s^3 + (K_P s + K_I + K_D s^2)\left(\frac{K_{hw}}{m}\right)} \quad (1)$$

In this experiment, we will work with the derivative term in the return path. The closed-loop transfer function in this case
(see *Figure 1(b)*) is:

$$H_{cl}(s)=\frac{X(s)}{R(s)}=\frac{\left(K_P+\frac{K_I}{s}\right)K_{hw}\left(\frac{1}{m\,s^2}\right)}{1+\left(K_P+\frac{K_I}{s}\right)K_{hw}\left(\frac{1}{m\,s^2}\right)+K_{hw}\left(\frac{1}{m\,s^2}\right)(K_Ds)}=\frac{(K_Ps+K_I)\left(\frac{K_{hw}}{m}\right)}{s^3+(K_Ps+K_I+K_Ds^2)\left(\frac{K_{hw}}{m}\right)} \qquad (2)$$

Note that the characteristic polynomials are identical in the two cases. In the absence of the integral term (i.e. $K_I = 0$), the transfer function reduces to

$$H_{cl}(s)=\frac{X(s)}{R(s)}=\frac{K_PK_{hw}\left(\frac{1}{m\,s^2}\right)}{1+K_PK_{hw}\left(\frac{1}{m\,s^2}\right)+K_{hw}\left(\frac{1}{m\,s^2}\right)(K_Ds)}=\frac{\left(\frac{K_PK_{hw}}{m}\right)}{s^2+\left(\frac{K_DK_{hw}}{m}\right)s+\left(\frac{K_PK_{hw}}{m}\right)} \qquad (3)$$

$$\text{i.e. } \omega_n^2=\frac{K_PK_{hw}}{m},\ \text{and } 2\zeta\omega_n=\frac{K_DK_{hw}}{m} \quad \Leftrightarrow \quad \omega_n=\sqrt{\frac{K_PK_{hw}}{m}},\quad \zeta=\frac{K_D}{2}\sqrt{\frac{K_{hw}}{K_Pm}} \qquad (4)$$

.

Experimental work

- **Plant set-up**

Set up the plant with four 500g mass pieces on carriage #1 and with no springs or damper attached. The other carriages should be secured away from the range of motion of the first carriage.

- **Proportional control and evaluation of K_{hw}**

For proportional control, $K_I = 0$ and $K_D = 0$. As equation (4) shows, $\zeta = 0$ so no damping is provided by the control action (although, as found in Lab #1a, there will be a small amount of frictional damping which is not shown in the block diagram). From equation (4), determine the value of K_PK_{hw} so that the system behaves like a $\sqrt{2}$ Hz spring-mass oscillator (i.e., $\omega_n = 2\pi\sqrt{2}$ rad/sec). K_{hw} has an approximate value of 12,000 N/m but its value varies significantly from one set of equipment to another. Assuming a value of 12,000 N/m, determine the value of K_P. Now carry out a test and measure the actual frequency of oscillation using the following steps:

1. Go to ***Data/ Set-up Data Acquisition*** and set up to collect ***Encoder #1*** position and ***Commanded Position*** information. Go to ***Command/Trajectory*** and set up a closed-loop step size of 0 counts, dwell time of 3000 ms and 1 repetition.
2. Go to ***Set-up/Control Algorithm*** and set $T_s = 0.00442$ and select ***Continuous Time Control***. Select ***PI + Velocity Feedback*** (this is the return path derivative form shown in *Figure 1(b)*) and ***Set-up Algorithm***. Enter the K_P value determined above for a $\sqrt{2}$ Hz oscillation and set K_I and K_D to zero. Do not input values greater than $K_P = 0.08$. Select ***OK*** to exit.
   IMPORTANT: In this and all future work, be sure to stay clear of the mechanism before doing the next step. Selecting Implement Algorithm immediately implements the specified controller; if there is an instability or large control signal, the plant may react violently. If the system appears stable after implementing the controller, first displace it with a light, non-sharp object (e.g. a plastic ruler) to verify stability prior to touching plant.
   Select ***Implement Algorithm***, then ***OK***.
3. Go to ***Command/Execute***. Prepare to manually displace the mass carriage roughly 2.5 cm. Select ***Run***, displace the mass approximately 2.5 cm and release it. Do not hold the mass position for longer than about 1 second as this may cause the motor drive thermal protection to open the control loop.

4. Go to ***Plotting/Set-up Plot*** and plot ***Encoder #1*** position. Determine the frequency of oscillation.

Use the measured value of frequency to find an accurate value of K_{hw} for the system that you are working on. Use this value for the rest of the experiment.

Now adjust K_P to the correct value to give an oscillation frequency of $\sqrt{2}$ Hz and repeat the above steps to check the frequency.

What do you expect to happen when the proportional gain K_P is doubled? Verify your predictions experimentally. Again, for system stability, do not input values greater than $K_P = 0.08$.

- **Derivative control**

Determine the value of the derivative gain K_D to achieve $K_D K_{hw} = 50$ N/(m/s). Repeat step 2 above, except input the value for K_D computed above, and set K_P as well as K_I to zero. Do not input values greater than $K_D = 0.04$.

After checking the system for stability by displacing it with a ruler, manually move the mass back and forth to feel the effect of viscous damping provided by K_D. Do not excessively coerce the mass, as this will again cause the motor drive thermal protection to open the control loop.

Repeat for a value of K_D five times as large (again make sure that $K_D < 0.04$). Can you feel the increased damping?

- **PD control design**

Using equation (4), design controllers (i.e. find K_P and K_D) for three systems with the same natural frequency of 4 Hz (i.e. $\omega_n = 8\pi$ rad/sec), and damping ratios of (i) $\zeta = 0.2$ (i.e. underdamped), (ii) $\zeta = 1.0$ (i.e. critically damped), and (iii) $\zeta = 2.0$ (i.e. overdamped).

Step Response:
Implement the underdamped controller (via ***PI + Velocity Feedback***) and set up a trajectory for a 2500 count closed-loop ***Step*** with 1000 ms dwell time and 1 repetition (which has the effect of setting up an input shown in *Figure 2*). Execute this trajectory and plot the ***Commanded Position*** and ***Encoder #1*** position. Plot them both on the same vertical axis so that there is no graphical bias.

Repeat for the critically damped and over-damped cases. Save your plots for later comparison.

Frequency Response:
Implement the underdamped controller. Set up a trajectory for a 600 count closed-loop ***Sine Sweep*** from 0.1 Hz to 20 Hz of 60 seconds duration with ***Logarithmic Sweep*** checked (which has the effect of setting up an input shown in *Figure 2*; the signal is actually a "chirp" signal of the form $\sin(2\pi f_i)$ with $f_i = f_0 + 10^{\beta t}$). (You may wish to specify ***Encoder #1*** data only via ***Set-up/Data Acquisition***. This will reduce the acquired data size.) Execute the trajectory and plot the ***Encoder #1*** frequency response using ***Linear Time*** and ***Linear Amplitude*** for the horizontal and vertical axes. The data will reflect the system motion seen as the sine sweep was performed. Now plot the same data using ***Logarithmic Frequency*** and ***dB Amplitude***. By considering the amplitude (the upper most portion of the data curve) you will see the data in Bode magnitude format. Can you easily identify the resonance frequency, and gain slopes in dB/decade?

Repeat for the critically damped and overdamped cases (can you easily identify the high frequency (>5 Hz) and low frequency (< 0.8 Hz)?). Save your plots for later comparison.

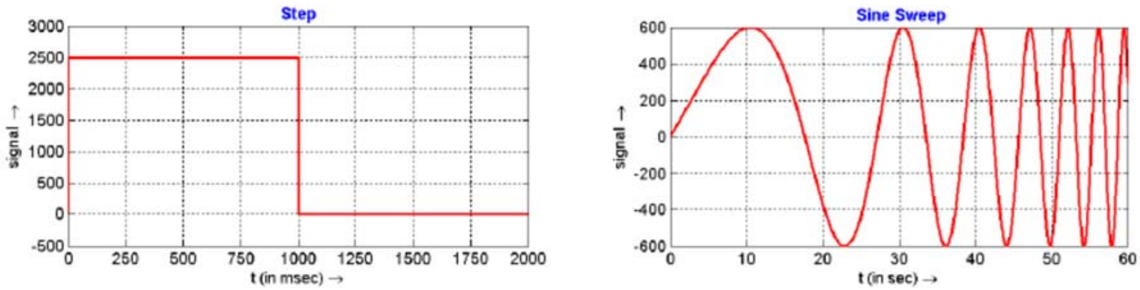


Figure 2: Signal shapes

- **Adding integral action**

Now compute K_I such that $K_I K_{hw} = 7500\ \text{N/(m-sec)}$. Implement a controller with this value of K_I, and values of K_P and K_D that achieved critical damping under the PD control design above. Do not input $K_I > 3.0$. Be certain that the following error seen in the background window is within 20 counts prior to implementing (if not, choose ***Zero Position*** from the ***Utility*** menu). Execute a 2500 count closed-loop ***Step*** of 2000 ms duration (1 repetition). Plot the ***Encoder #1*** position and the ***Commanded Position***.

Increase K_I by a factor of two, implement your controller (again make sure that $K_I < 3.0$) and plot its step response. Manually displace the mass by roughly 5 mm. Can you feel the integral action increasing the restoring control force with time? (Do not hold for more than about 2 seconds to avoid excessive force build-up and hence triggering of the motor thermal protection.) What happens when you let go?

- **Modeling static friction**

Static or Coulomb friction may be modeled as some disturbance force acting on the plant, as shown in *Figure 3*. Assume that this force is a step function, and use the Final Value Theorem to explain the effect of such an input on the PD controlled system with and without the addition of an integral term.

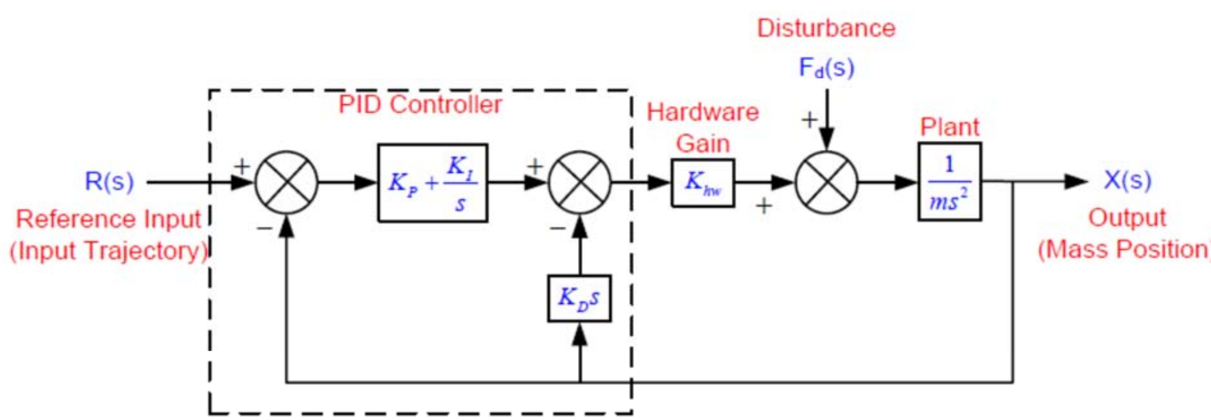


Figure 3: Friction modeled as disturbance force on the plant

Experiment#8

Collocated PD Control With 2 DOF Plant

Dr. Rad

In this experiment we consider PD control of a 2-mass system where the controlled output, x_1, is of the first mass. Such a scheme is referred to as collocated since the sensor output is rigidly coupled to the actuator input.

The addition of the spring and second mass increases the plant order by two and adds an oscillatory mode to the plant dynamics. This may be thought of, in a sense, as a dynamic disturbance to rigid body plant studied in other Experiment. The collocated control implemented here may be practically employed when there is flexibility between the actuator and some inertia, with the location of objective control being near the actuator. If the location of objective control is at the distant inertia, however, this method has its limitations.

The approach in this experiment will be to design the controller by interactively changing the PD gains and observing their effect on the physical system.

<u>Procedure :</u>

1. Set-up the system with four masses on the first and second carriage and with a medium stiffness spring (nominally 400 N/m) connecting the two as shown in Figure 1d. I.e. in the same configuration (using the same spring) as was used in identifying the transfer function at the end of Experiment #1 without using the added dashpot damping.
2. Implement the critically damped controller in Section before being sure that encoder #1 is selected for control. Set-up data acquisition for encoders 1 & 2 and for commanded position and gather data every 5 servo cycles. Execute a 2000 count step response and plot the result for commanded position and encoder #1. Export the data from ECP to MATLAB, and plot the Encoder 1 data using MATLAB. Be sure to clearly label the plots. To export the data

   to MATLAB,
3. Now iteratively adjust the gains k_p & k_d and plot results to obtain an improved response. Make your gain adjustments gradually (not more than 50% at a time) and note the effects of increasing or reducing each of them. Do not input k_p >1.2 or $0.01 < k_d > 0.05$. Attempt to achieve performance goals for the first mass of $\leq$ 200 ms rise time (0-90% amplitude) and $\leq$ 5% overshoot without excessive oscillation. Save your best step response plot. Gently displace the first mass (manually) and note the relative stiffness of the servo system at the first mass.

4. For your last iteration in Step 3, plot the step response of the second mass. (There is no need to rerun the step, simply re-setup the plot for "encoder #2" & "commanded position" and plot data.) What is the predominant characteristic of the second mass motion? Can you give an explanation for the difference in

the responses of the two masses in terms of their closed loop transfer functions?

5. Now using the existing values of k_p & k_d as starting points, iteratively reduce gains and plot x_2 results to provide a well-behaved step response with ≤ 10% overshoot, without excessive oscillation, and as fast a rise time as possible. Save your final plot and record the corresponding gains. Manually displace the first and second masses and note their relative stiffness. Are they generally more or less stiff than for the controller of Step 3? How does the steady state error compare with the high gain controller from Step 3? Could this error be reduced by feedback about x_2?

Questions:

A. Calculate the poles of the closed-loop transfer functions: $x_1(s)/r(s)$ and $x_2(s)/r(s)$ for your final controllers in Steps 3 & 5 respectively. How close to the imaginary axis (& right half plane) are the most lightly damped poles in each case? How close are the complex poles of $x_1(s)/r(s)$ to its zeros in each case? Explain your answer in terms of the root loci for this system for gain ratios of k_d/k_p = 0.05, 0.10, 0.17, and 0.25.

B. Calculate the closed loop transfer function in the form:

$$\frac{x_2(s)}{r(s)} = \frac{(N(s))_{forward\,path} \,/\, D_{ol}(s)}{1 + N_{ol}(s)/D_{ol}(s)} \tag{1}$$

Use N_{ol} and D_{ol} to obtain the open loop Nyquist or Bode responses resulting from your high and low gain controllers from Steps 3 & 5 respectively. What are the associated phase and gain stability margins? What are these margins for $x_1(s)/r(s)$? Explain.

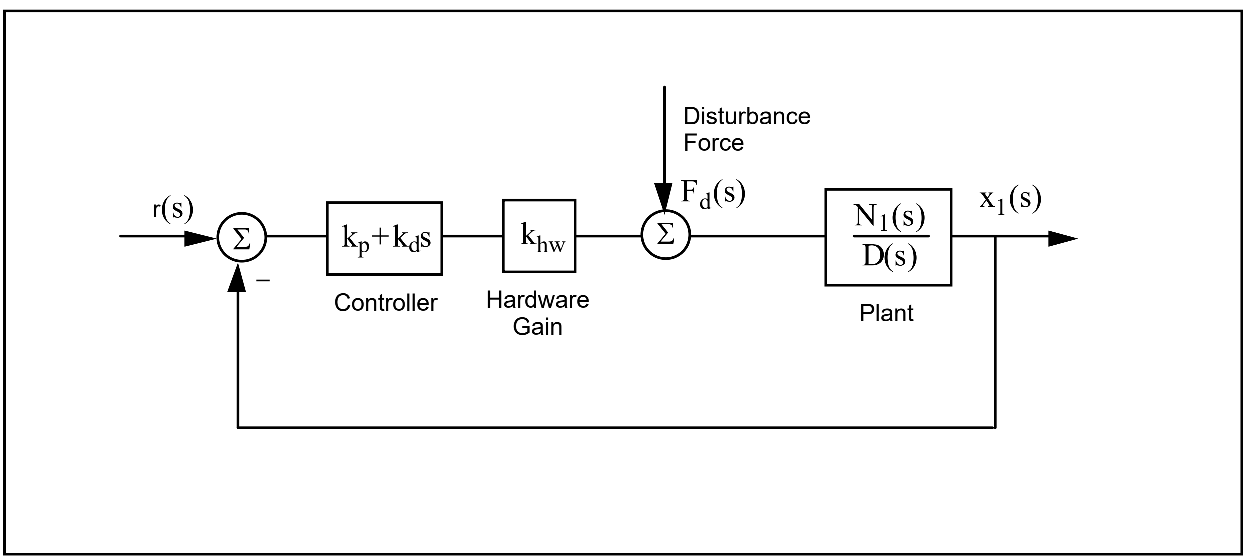


Figure 8.4-1. Disturbance Forces On PD Controlled 2-DOF Plant

Experiment#9

PD Plus Notch Filter Control

In this experiment a control scheme is implemented which first closes the collocated loop with simple rate feedback to dampen the oscillatory mode. Then a notch filter is designed to further attenuate the transmission of signals at the damped mode frequency (i.e. nearly canceling the poles with a zeros). Finally, PD control is used to achieve certain performance goals..

Procedure :

1. The plant studied here is that of the previous exercise, i.e. the one shown in Figure 1d but without the dashpot. Consider the block diagram of Figure 1. Use root locus techniques to find the rate feedback gain, k_v, that provides the satisfactory damping of the complex roots of the inner loop $x_1(s)/r^*(s)$.
2. Implement this gain as the f_1 coefficient in the General Form controller specification box using T_s = 2.652 ms. Be sure that you select Encoder #1 for Feedback Loop #3 before exiting the box and implementing. Attempt to manually excite the oscillatory mode via the second mass and notice the damping effect of velocity feedback.
3. We may now design for the new "plant" $G^*(s)$ where $N_2(s)$ is as before, and:

$$D^*(s) = D(s) + k_v s N_1(s) \tag{1}$$

Design a notch filter, $N_n(s)/D_n(s)$ with two poles at 5.0±5.0i Hz and two additional higher frequency poles at 8.0±8.0i Hz[14] and with two zeros at the poles of $D^*(s)$. Normalize the notch filter transfer function to have unity DC gain.[15]

---

[14]I.e. with pairs of poles at 7.07 and 11.3 Hz and damping of 70.7% ($\zeta=\sqrt{2}$). The first pole pair is necessary to make the filter proper and is placed sufficiently beyond the zero pair in $N_n(s)$ so as not to adversely effect its effectiveness in canceling the corresponding poles in $D_n(s)$. The second pole acts as a cascaded low pass filter to attenuate higher frequency noise.

[15]For discrete time design, convert your notch filter design to the z domain using the Tustin (bilinear) transformation:

$$s = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}$$

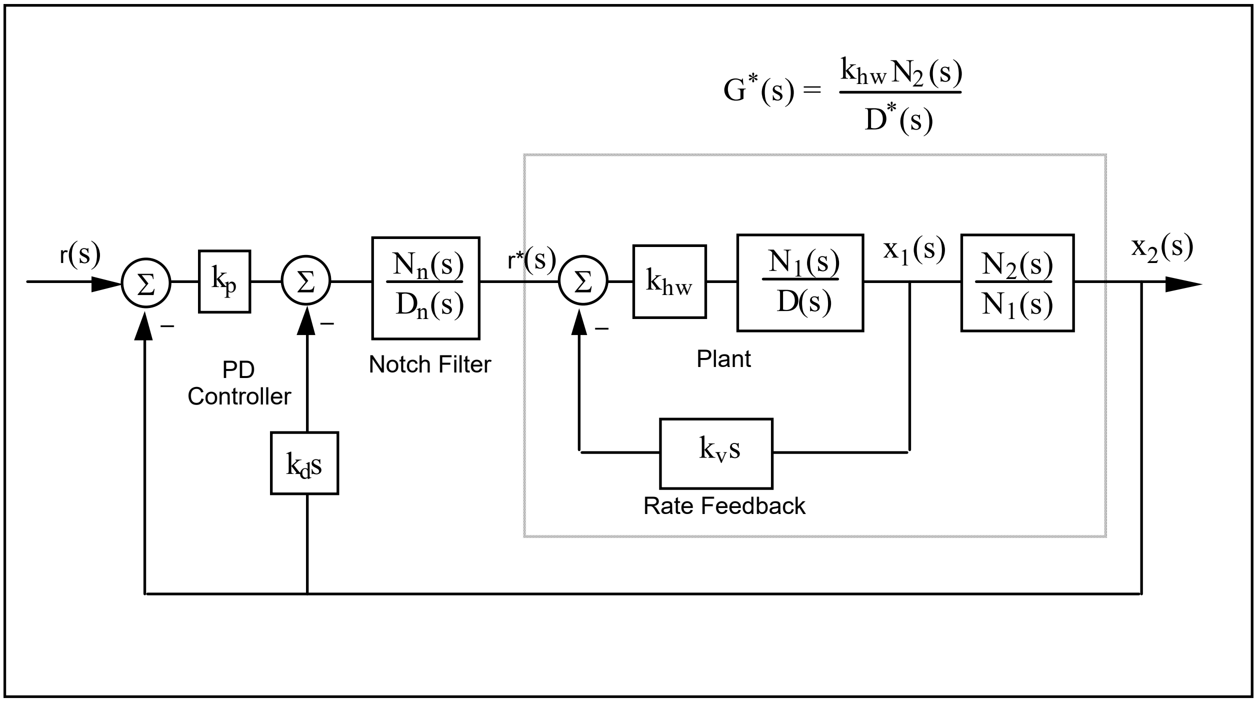


Figure 1. Control Structure

4. Select some initial PD gains for control of x_2. Use values of $k_p = 0.01$, $k_d = 0.001$ initially. Do not exceed $k_p = 0.05$ and $k_d = 0.05$ in any subsequent run. Since this control scheme uses multiple loop feedback, the Executive program requires that it be implemented via the general control algorithm form. A short-cut to controller coefficient entry is as follows: Input your notch filter coefficients and proportional controller gain, k_p, via the dialog box PID + Notch. Be certain to enter these coefficients in the proper order and to high numerical precision (e.g. 8 decimal places). Exit and select Preview In General Form. You will see the P + notch controller in the form that it is implemented in the generalized controller form.[16]

   In the General Form window, enter the f1 coefficient calculated in Step 1 (i.e. k_v), and enter as i1 the desired derivative gain, k_d. Make sure that Encoder #1 is selected for Loop #3 and Encoder #2 is selected for Loops #1&2. You should verify that these entries appropriately represent the control structure of the controller in Figure 1. Exit the box and make sure that General Form is

[16] *r(s)* will be the same as $D_n(s)$ and s(s) & *t(s)* will be identical and equal to $k_p N_n(s)$. Upon selecting Preview In General Form the algorithm is also converted to discrete time form (by the Tustin transform for N_n and D_n, backwards difference for any derivative terms). In this way the discrete time equivalent controller may be viewed in the discrete time General Form controller box.

selected; then Implement. Now check performance of the control using step trajectories of 500 - 1000 counts.[17]

5. Iteratively change PD gains by repeating Step 4 to obtain performance goals of 0.4 sec rise time (0-90% of final amplitude) and overshoot less than 15%. Do not exceed k_p=0.10 or k_d=0.01. You may notice a trend here that for relatively high values of k_d the system transmits excessive noise and may appear "twitchy". As k_p becomes large, the system becomes oscillatory and further increases lead to instability. Does increase in noncollocated derivative action necessarily reduce the step response oscillations?
6. Record your best performance step response. Manually displace the masses and note the relative static stiffness of the noncollocated (second) carriage under this control.

Questions:

A. Report your calculated values for k_v, $N_n(s)$, $D_n(s)$, and your selected values of k_p, k_d. Submit your step response plot. Does it meet the performance goals of Step 5?

B. Calculate the closed-loop transfer function $x_2(s)/r(s)$ including all elements in the block diagram of Figure 1. You may express this in terms of the polynomials $D(s)$, $N_1(s)$, $N_n(s)$, etc. rather than expanding each term fully. Use the equation

$$\frac{x_2(s)}{r(s)} = \frac{(N(s))_{forward\,path} \,/\, D_{ol}(s)}{1 + N_{ol}(s)/D_{ol}(s)} \qquad (2)$$

and determine the phase and gain margins of your system design via Bode or Nyquist plots. What would happen if you doubled your gains k_p and k_d?. What if you quadrupled them?

[17]Relatively small step sizes may be necessary here to prevent DAC saturation during the initial step response transient. The student may wish to view the control effort during these trials. Saturation occurs at the maximum DAC output of 5.0 V.

Experiment#10

Successive Loop Closure / Pole Placement Design For 2 DOF Plant

Dr. Rad
Revise 6/2016

In this experiment we first close a position loop about the collocated (x_1) position with a relatively high bandwidth (close tracking) PD control. We then make the assumption that the first carriage closely follows its internal demand $r^*(s)$ so that for designing a controller for x_2, the "plant" is approximated by the transfer function $x_2(s)/x_1(s)$ (i.e. $N_2/N_1(s)$). The block diagram for this approach is given in Figure 1

The design and control implementation in this section proceeds as follows

1. High bandwidth PD control of x_1
2. Low pass filter augmentation to attenuate signal noise due to high PD gains
3. Outer loop control of x_2 via pole placement methodology

10.1 PD Control Of The First Mass Carriage

Set-up the system with four masses on the first and second carriage and with a medium stiffness spring (nominally 400 N/m) connecting the two. I.e. in the same configuration (using the same spring) as the previous experiment.

PD Control Design & Implementation

1. Design the PD control gains such that ω_n = 10 Hz, and ζ = 0.707 when considering only m_1 acting as a rigid body.
2. Set the sampling rate to T_s = 0.002652 seconds and implement your gains via the PID control algorithm box. You may notice some system noise associated with the high derivative gain term. After safety checking the controller displace the first and second mass carriages and notice their relative stiffness. Discontinue the control via Abort Control.

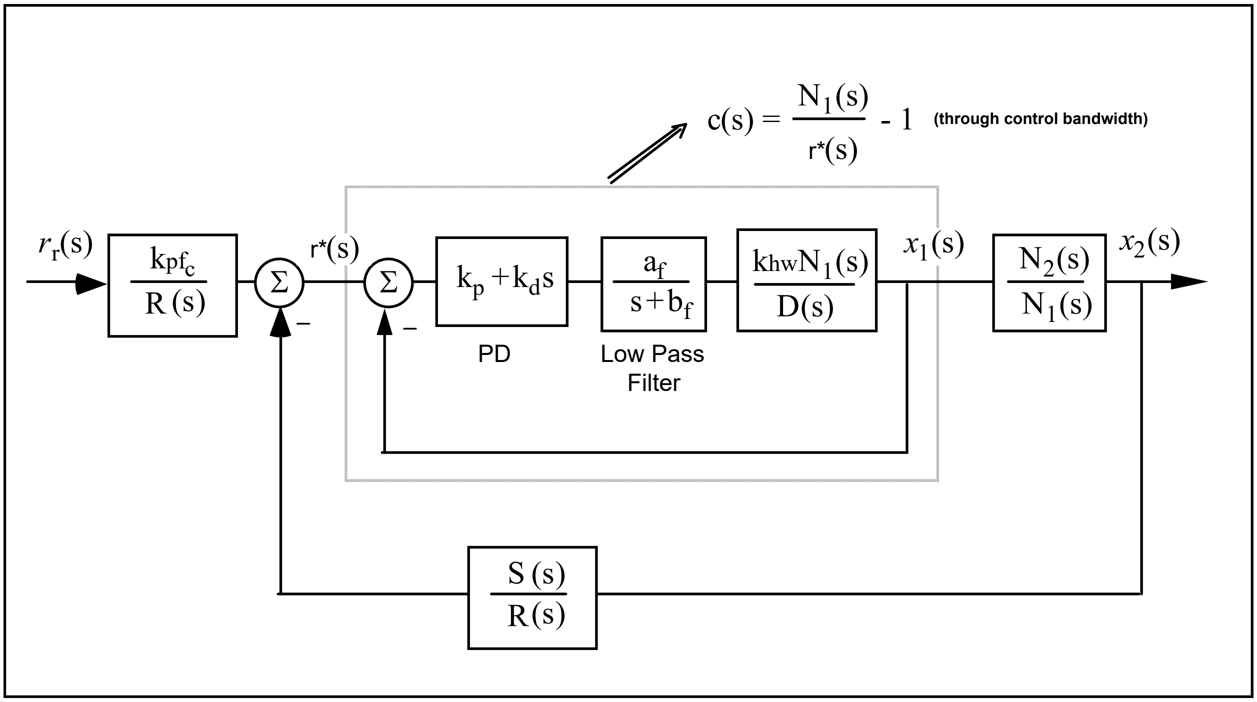


Figure 1. Control Structure For Successive Loop Closure with High Gain Inner Loop

<u>Low Pass Filter Design & Implementation</u>

3. Solve for the constants a_f, and b_f such the filter has a pole at s = -240 (approximately 40 Hz) and has unity DC gain.
4. Calculate the numerator and denominator of the controller associated with the cascade of the PD and low pass filter blocks. These will have the form

$$\text{PD*Filter} = \frac{e_0 + e_0 s}{g_0 + g_1 s} \tag{1}$$

5. In the Generalized Form box, enter the following.
   a. Enter your calculated e_i's and g_i's
   b. Select Encoder #1 for Loop #2. (the other loops may have any encoder selected at this point)
   c. Set the following equal to 1: t_0, h_0, i_0, j_0
   d. Verify that r_1 = 0.0000002[18]
   e. Verify that all other coefficients = 0

[18] This small value is needed to provide a proper transfer function for bilinear transformation and subsequent discrete control implementation whenever T/R and S/R are used to implement a differentiator. Its small value results in a pole many decades beyond the system bandwidth and is of no practical implication to system modeling or performance. This coefficient may be set to zero here, but should generally remain.

It is important to take care and assure that all parameters are properly set before leaving the Generalized Form dialog box and implementing control.

6. Exit the dialog box, verify that the Generalized Form button is selected, and Implement control. You should notice a reduction in any audible signal noise that may have been present without the low pass filter. Safety check the controller and manually displace the carriages. They should behave as before in Step 2.

10. 2 Pole Placement Control of $x_2(s)/x_1(s)$

Having closed a relatively high bandwidth (≈10 Hz) loop about the first mass, we utilize the fact that the transfer function of Eq. 6.2-5 has near unity input/output gain (and near zero phase) through the bandwidth (≈ 2.5 Hz) that we will attempt to attain in the overall control of x_2. Thus for the control of x_2 we consider the outer loop in the block diagram of Figure 6.6-1.

Now the plant to be controlled is:

$$\frac{x_2(s)}{x_1(s)} = \frac{N_2}{N_1} \triangleq \frac{N^*(s)}{D^*(s)} \tag{2}$$

The numerical values of the parameters in this expression were determined in Experiment #1.

We now seek to find a controller $S(s)/R(s)$ which will result in a prescribed set of closed loop poles. The closed loop denominator will have the form:

$$D_{cl}(s) = D^*(s)R(s) + N^*(s)S(s) \tag{3}$$

which may be expressed as[19]

$$D_{cl}(s) = (d_2 s^2 + d_1 s + d_0)(r_1 s + r_0) + (n_0)(s_1 s + s_0) \tag{4}$$

where the d_i's and n_i's are the respective coefficients of the denominator and numerator of the right hand side of Eq. (2). Their values are known from the plant model

[19]The notation here is the obvious one.

By linear system theory, for coprime N*(s), D*(s) with N*(s)/D*(s) proper, there exists an (n-1)th order $S(s)$, $R(s)$ which when convolved as per Eq. (6.6-3) form an arbitrary (2n-1)th order $D_{cl}(s)$ where n is the order of $D^*(s)$.

Here we shall solve for the desired denominator:

$$D_{cl}(s) = \left(s + 5\pi\frac{(1+j)}{\sqrt{2}}\right)\left(s + 5\pi\frac{(1-j)}{\sqrt{2}}\right)(s + 5\pi) \tag{5}$$

I.e. closed loop poles at -2.5, and -1.77$\pm j$1.77 Hz.[20]

Pole Placement Design

7. Determine the coefficients of the controller polynomials $S(s)$ and $R(s)$ by equating coefficients in the expanded forms of Eq's 4 and 5.
8. Calculate the scalar prefilter gain k_{pf} by referring to Figure 1. The goal is to have the output $x_2(s)$ scaled equal to the input $r_r(s)$. Hint: Consider the system in static equilibrium. Set x_2 =1 and r_r = 1 and solve for k_{pf} using only the constant terms in all control blocks.

Control Implementation

9. In the Executive program, set-up to collect Encoder #1', Encoder #2 and Commanded Position information[21] via Set-up Data Acquisition with data sampling every two sample periods. Setup a closed loop step trajectory of 2500 counts, 2000 ms duration and 1 repetition.
10. Return to the General Form Algorithm box and verify that the PD controller, low pass filter and all other coefficients are entered as given in Step 5 above. Enter the coefficients for $S(s)$ and $R(s)$ determined in Step 7. Enter the value k_{pf} calculated in Step 8 as to. Make sure that Encoder #2 is selected for Loop #1 and Encoder #1 for Loop #2. Select OK to exit to the controller selection dialog box.
11. While staying clear of the mechanism select Implement Algorithm. If the mechanism reacts violently you have implemented an unstable controller or otherwise improperly entered the control coefficients and you will need to repeat the above steps as appropriate. You should first Reset Controller (Utilities menu) before attempting to re-implement control. If the system is well behaved, and after safety checking the controller, you may disturb the mass carriages lightly. Notice the relative stiffness of the two carriages and how the

[20] This has poles of magnitude |s| = 2.5 Hz that lie at 135, 180, and 225 deg. It is similar to a third order Butterworth polynomial but somewhat more damped.
[21] You may also select Control Effort if you wish to later observe this value .

first carriage moves in opposition to (i.e. regulation of) disturbances of the second carriage.

You may notice some "twitching" or buzzing due to noise in the system. This often occurs in such high gain systems, but has been mitigated via the low pass filter. <u>If the noise is excessive or there is any possibility that the equipment is at risk discontinue control immediately.</u>

12. Execute the Step input previously programmed, and plot the Encoder 1, Encoder #2, and Commanded Position data. Save your plot. How does the response at x_2 compare with designs previously tested? Describe the motion of x_1 and how it shapes the response at x_2.

Questions:

A. Report your calculated values for k_p, k_d, a_f, b_f, e_0, e_1, g_0, g_1, s_0, s_1 , r_0, r_1, & k_{pf}

B. Determine an expression for the closed-loop transfer function $x_2(s)/r_r(s)$ including all elements in the block diagram of Figure 6.6-1. You may express this in terms of the polynomials D*(s)*, $N_1(s)$, *R(s)*, etc. rather than expanding each term fully. Determine $x_2(s)/r_r(s)$ using the assumption c*(s)*=1. Compare the simulated frequency response of the full and reduced order transfer functions. In which regions are the two similar in magnitude and phase and in which are they different? Are they similar throughout the final system closed loop bandwidth? Is the assumption of unity gain in c*(s)* valid for the purposes here?

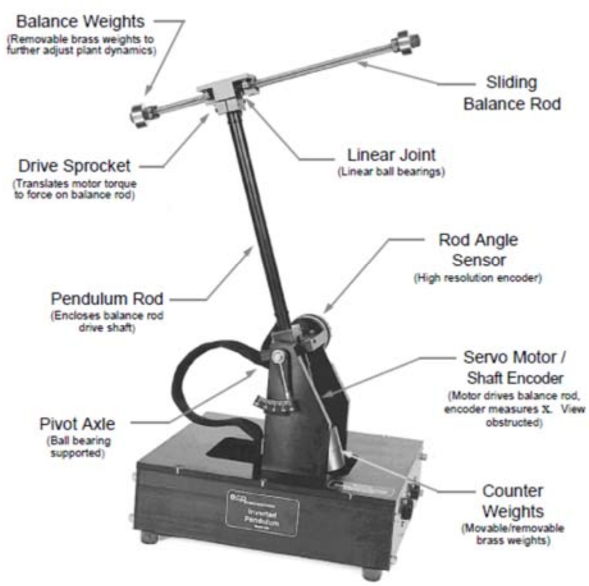
M505

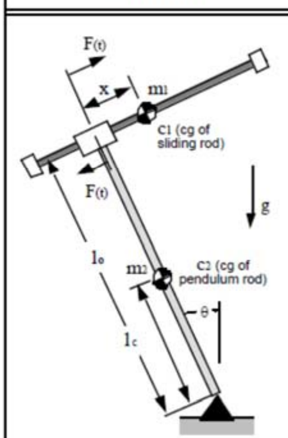
Pendulum Control System

Experiment#11

Inverted Pendulum

This unique ECP design vividly demonstrates the need for and effectiveness of closed loop control. It is *not* the conventional rod-on-cart inverted pendulum, but rather, it steers a horizontal balancing rod in the presence of gravity to control the vertical pendulum rod. As detailed in the manual, the plant has both right half plane poles and zeros as well as kinematic and gravitationally coupled nonlinearities. By adjusting mass properties, these roots may be varied to make the control problem range from being relatively simple to theoretically impossible! The system includes removable and adjustable moment arm counterweights on the vertical and horizontal rods for quick adjustment of the plant dynamics. It features
linear and rotary ball bearings at the joints for low friction and consistent dynamic properties.



Plant Model	*Dynamic Equations*		*Characteristics*
	*"Exact"*	$m_1\ddot{x}(t)+m_1 l_o\ddot{\theta}(t) - m_1 x(t)\dot{\theta}(t)^2 - m_1 g\sin\theta(t) = F(t)$ $m_1 l_o\ddot{x}(t)+J_o(x)\ddot{\theta}(t)+2m_1 x(t)\dot{x}(t)\dot{\theta}(t) - (m_1 l_0+m_2 l_c)g\sin\theta(t) - m_1 g x(t)\cos\theta(t) = 0$ $J_o(x) = J_1+m_1(l_o^2+x^2)+J_1+m_2 l_c^2$	• Nonlinearities in kinematic constraints and coordinate dependent mass properties.
	Linearized Time Domain	$m_1\ddot{x}(t)+m_1 l_o\ddot{\theta}(t) - mg\,\theta(t) = F(t)$ $m_1 l_o\ddot{x}(t)+J_o^*\ddot{\theta}(t) - (m_1 l_c+m_2 l_o)g\theta(t) - m_2 g x(t) = 0$ $J_o^* = J_o\|_{x=0}$	• Linearization about x = 0, θ = 0 shown to be valid for many control schemes.
	S-Domain	$\frac{\theta(s)}{F(t)} = \frac{-(l_o s^2 - g)}{(J_o^* - m_1 l_o^2)s^4 + (m_2 l_o - m_1 l_c)g s^2 - m_2 g^2}$	• One RHP, 2 oscillatory poles. • Nonmin phase (RHP zero). • Attainable bandwidth bounded from above and below by RHP roots.

An **inverted pendulum** is a classic control problem. The process is non linear and unstable with one input signal and several output signals. The aim is to balance a pendulum vertically on a motor driven wagon. The following figure shows an inverted pendulum. The aim is to move the wagon along the x direction to a desired point without the pendulum falling. The wagon is driven by DC motor, which is controlled by a controller (analog in our implementation). The wagons x position (not in our case) and the pendulum angle è are measured and supplied to the control system. A disturbance force, $F_{DISTURBANCE}$, can be applied on top of the pendulum.

Back ground

This Experiment provides time domain expressions which are useful for nonlinear plant modeling and Laplace domain expressions useful for linear control implementation. These are used in the experiments described later in this manual.

11.1 Equations of Motion

11.1.1 Nonlinear Expressions

Neglecting friction effects, the plant may be modeled as shown in Fig. 7.1-1[22]. From the figure we have that the system kinetic energy, **T**(s) is:

$$\mathbf{T} = \frac{1}{2}m_1 v_{cg_1}^2 + \frac{1}{2}J_1\dot{\theta}^2 + \frac{1}{2}m_2 v_{cg_2}^2 + \frac{1}{2}J_2\dot{\theta}^2 \qquad (11.1\text{-}1)$$

[22]The force F(t) shown in the figure is equivalent to the motor torque divided by the drive pulley / belt contact radius.

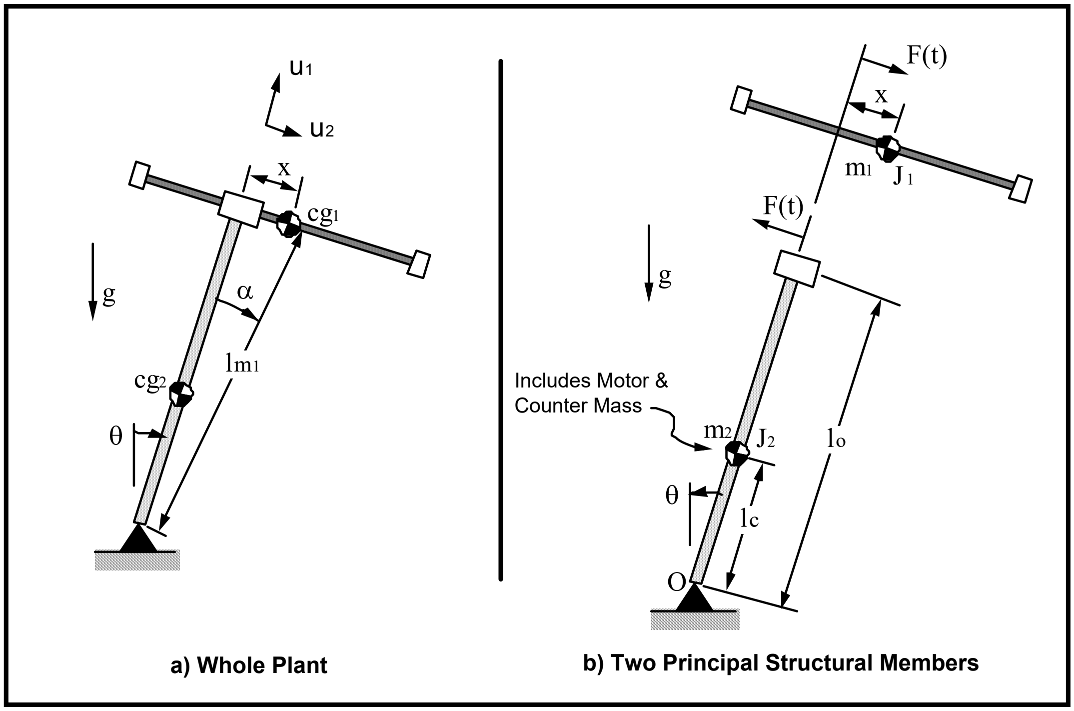


Figure 11.1-1 Plant Model Descriptions

where the v_{cg}'s are the inertial velocities of the respective members' centers of gravity (c.g's) (see Fig. 11.1-1b) and the J_i's are the polar moments of inertia about the respective c.g's. The members are kinematically constrained so that:

$$v_{cg_1} = \left| \underline{\dot{x}} + \underline{\dot{\theta}} \times \underline{l_{m1}} \right| \tag{11.1-2}$$

$$v_{cg_2} = l_c \dot{\theta} \tag{11.1-3}$$

where the underscored quantities are treated as vectors and "x" is the vector cross product. By resolving unto the $\underline{u_1}$ and $\underline{u_2}$ components and squaring, it is readily shown that:

$$v_{cg_1}^2 = \dot{x}^2 + \left(l_{m1}\dot{\theta}\right)^2 + 2\left(l_{m1}\dot{x}\dot{\theta}\cos\alpha\right) \tag{11.1-4}$$

Substituting Eq's 7.1-3,-4 into 7.1-1 and using: $l_{m1}^2 = l_o^2 + x^2$ and $\cos\alpha = l_o/l_{m1}$ yields:

$$\begin{aligned} \mathbf{T} &= \tfrac{1}{2}\left(J_1 + J_2 + m_1\left(l_o^2 + x^2\right) + m_2 l_c^2\right)\dot{\theta}^2 + \tfrac{1}{2}m_1\dot{x}^2 + m_1 l_o \dot{x}\dot{\theta} \\ &= \tfrac{1}{2}J_o(x)\dot{\theta}^2 + \tfrac{1}{2}m_1\dot{x}^2 + m_1 l_o \dot{x}\dot{\theta} \end{aligned} \tag{11.1-5}$$

where

$$J_o(x) = J_1 + J_2 + m_1\left(l_o^2 + x^2\right) + m_2 l_c^2 \tag{11.1-6}$$

is the system moment of inertia about O.

The potential energy, **V**, taken with the datum at {θ=90º, x=0} is:

$$\begin{aligned} \mathbf{V} &= m_1 g l_{m1}\cos(\theta + \alpha) + m_2 g l_c \cos\theta \\ &= m_1 g l_{m1}\left(\cos\theta\cos\alpha - \sin\theta\sin\alpha\right) + m_2 g l_c \cos\theta \\ &= m_1 g\left(l_o \cos\theta - x\sin\theta\right) + m_2 g l_c \cos\theta \end{aligned} \tag{11.1-7}$$

To obtain the equations of motion, we use Lagrange's equation in the form:

$$\frac{\partial}{\partial t}\left(\frac{\partial \mathbf{T}}{\partial \dot{q}_i}\right) - \frac{\partial \mathbf{T}}{\partial q_i} + \frac{\partial \mathbf{V}}{\partial q_i} = Q_i \tag{11.1-8}$$

where q_i is the i^{th} generalized coordinate, and Q_i is the associated generalized force[23]. Choosing q_1=x, q_2=θ, we have:

$$m_1\ddot{x}+m_1 l_o\ddot{\theta} - m_1 x\dot{\theta}^2 - m_1 g\sin\theta = F(t) \qquad (11.1\text{-}9)$$

$$m_1 l_o\ddot{x}+J_o\ddot{\theta}+2m_1 x\dot{x}\dot{\theta} - (m_1 l_0+ m_2 l_c)g\sin\theta - m_1 g x\cos\theta = 0 \qquad (11.1\text{-}10)$$

These equations are useful for nonlinear system design and analysis. For example, when linear control is implemented based on a linearized plant model, the system time response with the full-term (nonlinear) and linearized plants may be simulated to evaluate the range of validity of the linear approximation. The simulation with nonlinear plant model may then be compared with actual test results on the ECP system to see the effects of further unmodeled and non-ideal dynamic behavior.[24]

7.1.2 Linearization About Equilibrium

The equilibrium points $x=x_e$, $\theta=\theta_e$, may be found from Eq's 11-8 & -9 by solving for the motionless system ($\dot{x}=\ddot{x}=\dot{\theta}=\ddot{\theta}=0$) when F(t) = 0. These are readily obtained as:[25]

$$\theta_e = 0, x_e = 0 \qquad (11.1\text{-}11)$$

A linearized approximation of the system may be found via the first two (zeroeth and first order) terms of the Taylor's series expansion about the equilibrium points[26]. This results in:

---

[23]The Q_i's may be found using the definition of virtual work, δW_i: i.e. $\delta W_i = Q_i\ \delta q_i$. In this case, the only external force acting on the system is F(t) which acts linearly in the positive x direction and is independent of x and θ, i.e. $\delta W_1 = F(t)\delta x$, $\delta W_2 = 0$.

[24]While ECP systems follow their dynamic models quite closely, virtually any physical systems' behavior is more complex than our mathematical models of it. Characteristics not accounted for in this analysis include Coulomb friction at the belt drive, bearings, and motor brushes; motor torque ripple; quantization of the encoder signals and signal processing; and saturation of the digital-to-analog converter when the control effort signal becomes excessive. Useful models may be built however without accounting for all such effects. It is the essence of control modeling to establish a plant model of minimal order (complexity) to describe the salient dynamics through the intended control bandwidth.

[25]An additional solution exists at $\theta_e = 180^o$, but as the physical system is constrained to preclude this position, it shall not be considered here.

$$m_1\ddot{x}+m_1 l_o\ddot{\theta} - m_1 g\theta = F(t) \tag{11.1-12}$$

$$m_1 l_o\ddot{x}+J_{o_e}\ddot{\theta} - (m_1 l_0+ m_2 l_c)g\theta - m_1 g x = 0 \tag{11.1-13}$$

where:

$$J_{o_e} = J_o\Big|_{\substack{x=x_e\\ \theta=\theta_e}}$$

7.1.3 State Space Realization

Isolating the second derivative terms in equations 7.1-12,-13 we have:

$$\left(J_{o_e}-m_1 l_o^2\right)\ddot{x} + m_1 l_o g x - \left(J_{o_e}- m_1 l_o^2 -m_2 l_o l_c\right)g\theta = \frac{F(t)J_{o_e}}{m_1} \tag{11.1-14}$$

$$\left(J_{o_e}-m_1 l_o^2\right)\ddot{\theta} - m_1 g x -m_2 l_c g\theta = -F(t)l_o \tag{11.1-15}$$

By inspection, a state space realization of the linearized plant is:

$$\begin{aligned}\dot{\boldsymbol{x}} &= A\boldsymbol{x} + BF(t)\\ Y &= C\boldsymbol{x}\end{aligned} \tag{11.1-16}$$

where:

$$\boldsymbol{x} = \begin{bmatrix}\theta\\ \dot{\theta}\\ x\\ \dot{x}\end{bmatrix}, \quad A = \begin{bmatrix}0 & 1 & 0 & 0\\ m_2 l_c g/J^* & 0 & m_1 g/J^* & 0\\ 0 & 0 & 0 & 1\\ \left(J^* -m_2 l_o l_c\right) g/J^* & 0 & -m_1 l_o g/J^* & 0\end{bmatrix},$$

$$B = \frac{1}{J^*}\begin{bmatrix}0\\ -l_o\\ 0\\ J_{o_e}/m_1\end{bmatrix}, \qquad C = \begin{bmatrix}C_1 & 0 & 0 & 0\\ 0 & C_2 & 0 & 0\\ 0 & 0 & C_3 & 0\\ 0 & 0 & 0 & C_4\end{bmatrix}$$

and

$$J^* = \left[J_{o_e}-m_1 l_o^2\right] \tag{11.1-17}$$

---

[26]I.e. for our variable set: $(\bullet)\Big|_{\substack{x=x_e\\ \theta=\theta_e}} + \left(x\left(\frac{\partial}{\partial x}(\bullet)\Big|_{x=x_e}\right)+\theta\left(\frac{\partial}{\partial \theta}(\bullet)\Big|_{\theta=\theta_e}\right)\right)$ where (•) is Eq. 5.1-9 or -10. In this case, the result is the same as setting all squared and cross terms equal to zero, sinθ equal to θ, and cosθ equal to 1.

.

11.1.4 Transfer Function

By Laplace transform of Eq. (7.1-13) and assuming zero valued initial conditions it is straightforward to express

$$\frac{\theta(s)}{x(s)} = \frac{m_1 l_o}{J_{o_e}} \frac{-s^2 + g/l_o}{s^2 - (m_1 l_o + m_2 l_c) g / J_{o_e}} \qquad (11.1\text{-}18)$$

which when substituted into Eq. (7.1-12) leads to:

$$\frac{\theta(s)}{F(s)} = \frac{l_o}{J^*} \frac{-s^2 + g/l_o}{s^4 + \left((m_1 l_o - m_2 l_c) g / J^*\right) s^2 - m_1 g^2 / J^*} \qquad (11.1\text{-}19)$$

The first of the above transfer functions relates the motion of the nominally vertical (or pendulum) rod to the motion of the sliding rod and the second relates pendulum rod motion to the force acting on the sliding rod via the drive belt. It is the motion of the pendulum rod (i.e. θ) that is to be controlled in the experiments that follow.

The linearized relationship between the applied force and the sliding rod position follows from the product of Eq.(7.1-19) and the inverse of Eq. (7.1-18):

$$\frac{x(s)}{F(s)} = \frac{J_{o_e}}{m_1 J^*} \frac{s^2 - (m_1 l_o + m_2 l_c) g / J_{o_e}}{s^4 + \left((m_1 l_o - m_2 l_c) g / J^*\right) s^2 - m_1 g^2 / J^*} \qquad (11.1\text{-}20)$$

7.2 Effect Of Mass Properties & Geometry On Plant Dynamics

From inspection of Eq. 11.1-19, it is readily seen that the transfer function of the linearized plant has one right half plane (RHP) zero and at least one RHP pole. Thus the open loop system is both nonminimum phase and unstable. As will be discussed later, these characteristics have strong implications to the attainable performance and stability of any subsequent closed loop design. In particular, the nonminimum phase zero sets an upper limit to the achievable system bandwidth, and the unstable pole sets a lower limit. These properties are discussed in more detail in Appendix A. The effects of various dynamic parameters on the values of the plant transfer function roots are described in this section.

11.2.1 Distribution And Effect Of Zeros

From Eq 7.1-19, the zeros lie at:

$$z = \pm\sqrt{g/l_o} \tag{11.2-1}$$

Thus for the experimental plant the zeros are fixed since the pendulum rod length, l_o, is not changeable. From the discussions of Appendix A, the lower the "frequency" (magnitude) of the RHP zero, the lower the maximum attainable closed loop bandwidth, and hence performance. From the Appendix, for practical control implementation, we would not expect to attain closed loop bandwidths greater than about $(g/l_o)^{1/2}$ or 1.2 Hz for our particular geometry.

11.2.2 Distribution And Effect Of Poles

In order to gain insight as to the effect of mass properties on the distribution of the poles, consider the following notation:

$$a \triangleq (m_1 l_o - m_2 l_c) / J^* \tag{11.2-2}$$

$$b \triangleq m_1 / J^* \tag{11.2-3}$$

$$k \triangleq l_o / J^* \tag{11.2-4}$$

Note that since:

$$J^* = J_{o_e} - m_1 l_o^2 = J_1 + J_2 + m_2 l_c^2 \ (>0) \tag{11.2-5}$$

b>0 and k>0. Equation 11.1-19 may then be expressed as:

$$\frac{\theta(s)}{F(s)} = \frac{-k\left(s^2 + g/l_o\right)}{s^4 + ags^2 - bg^2} \tag{11.2-6}$$

or:

$$\frac{\theta(s)}{F(s)} = \frac{-k\left(s^2 + g/l_o\right)}{\left(s^2 - p_1 g\right)\left(s^2 - p_2 g\right)} \tag{11.2-7}$$

where:

$$p_1 = \frac{-a + \sqrt{a^2 + 4b}}{2} \quad (>0) \tag{11.2-8}$$

$$p_2 = \frac{-a - \sqrt{a^2+4b}}{2} \quad (<0) \qquad (11.2\text{-}9)$$

Equations 7.2-7,8,&9 show that their are two real poles at $\pm\sqrt{p_1 g}$ and two imaginary poles at $\pm i\sqrt{|p_2|g}$. Two possible pole-zero distributions are shown in Figure 7.2-1.

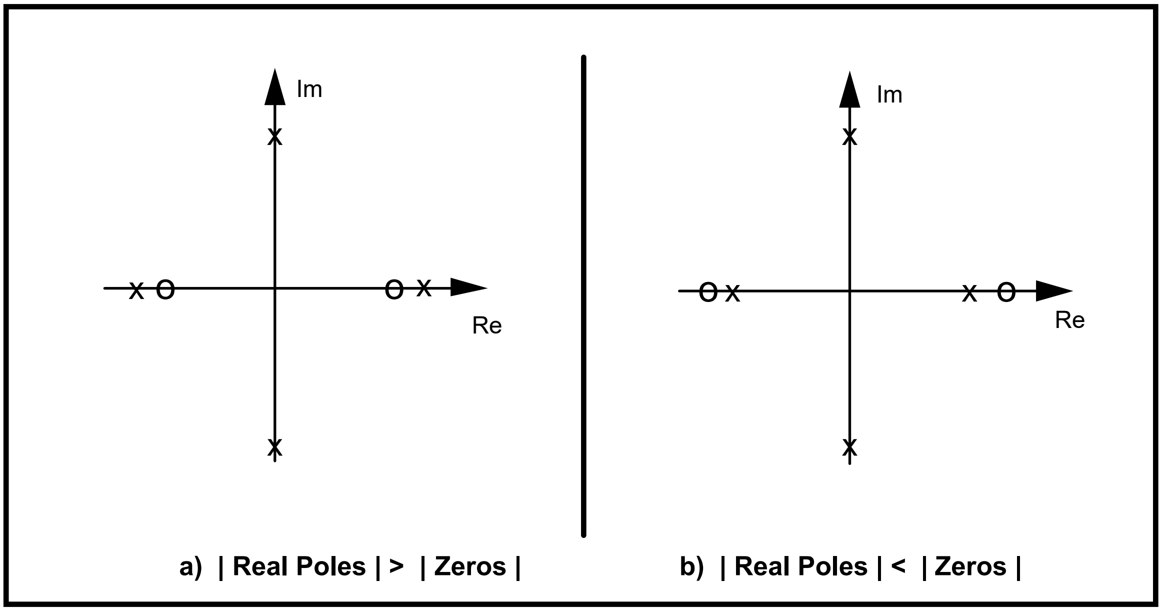


Figure 11.2-1. Possible Pole-Zero Distributions

According to Eq's 7.2-1,-7,&-8, the distribution of Figure 7.2-1a implies that:

$$\frac{-a + \sqrt{a^2+4b}}{2} > \frac{1}{l_o} \qquad (11.2\text{-}10)$$

It is relatively straightforward to show that this holds if and only if:

$$J^* < m_2 l_c l_o \qquad (11.2\text{-}11)$$

Similarly, the distribution of Fig. 7.2-1b implies that:

$$J^* > m_2 l_c l_o \qquad (11.2\text{-}12)$$

Eq. 7.2-11 is equivalent to the condition that $J_1+J_2+m_2(l_c)^2 < m_2 l_c l_o$. This in turn requires that J_1 and J_2 be small and l_c be small relative to l_o. It can be shown that for the range of available parameters, this condition is never realized in the experimental system. Therefore, all pole-zero distributions are as shown in Figure 7.2-1b. Note also from Eq's 7.2-8,-9 that the magnitude (frequency) of the imaginary poles is always greater than the real poles.

For the experimental system, the masses m_1 and m_2 may be changed discretely by adding or removing the appropriate brass weights and by changing the distance of the balance weight along the lead screw from the pivot. The effect of these changes is shown in Figure 6.2-2. From the discussions of Appendix A, the lower the frequency of the unstable pole, the less difficult it is to stabilize the system under closed loop control. As seen in the Figure, stabilizing the system becomes less difficult when the balance masses are added, when they are moved as far downward as possible,[27] and when the "donut" weights on the sliding rod are removed.[28]

The case where both balance masses are removed altogether is nearly identical to the left most points on the curves of Figure 6.2-2a.

[27]This result is rather intuitive – as the system cg. moves downward and below the pivot, the system would appear to become similar to an ordinary (non inverted) pendulum. Note from the transfer function denominator that the system *never* becomes stable even for large negative values of the product m_2l_c. This somewhat non intuitive result may be visualized by considering small static perturbations about the equilibrium. If for example the pendulum is moved slightly in the clockwise direction, the sliding rod will tend to move toward the right thereby causing further clockwise pendulum displacement. Thus the equilibrium is unstable. It may be demonstrated on the experimental system however that in such cases, relatively small amounts of friction result in a stable equilibrium.

Note that in the $\theta(s)/x(s)$ (Eq. 5.1-18) case however the transfer function becomes stable for sufficiently large negative values of m_2l_c.

[28]Removing the "donut" weights however reduces the available control authority (e.g. smaller attainable range in θ).

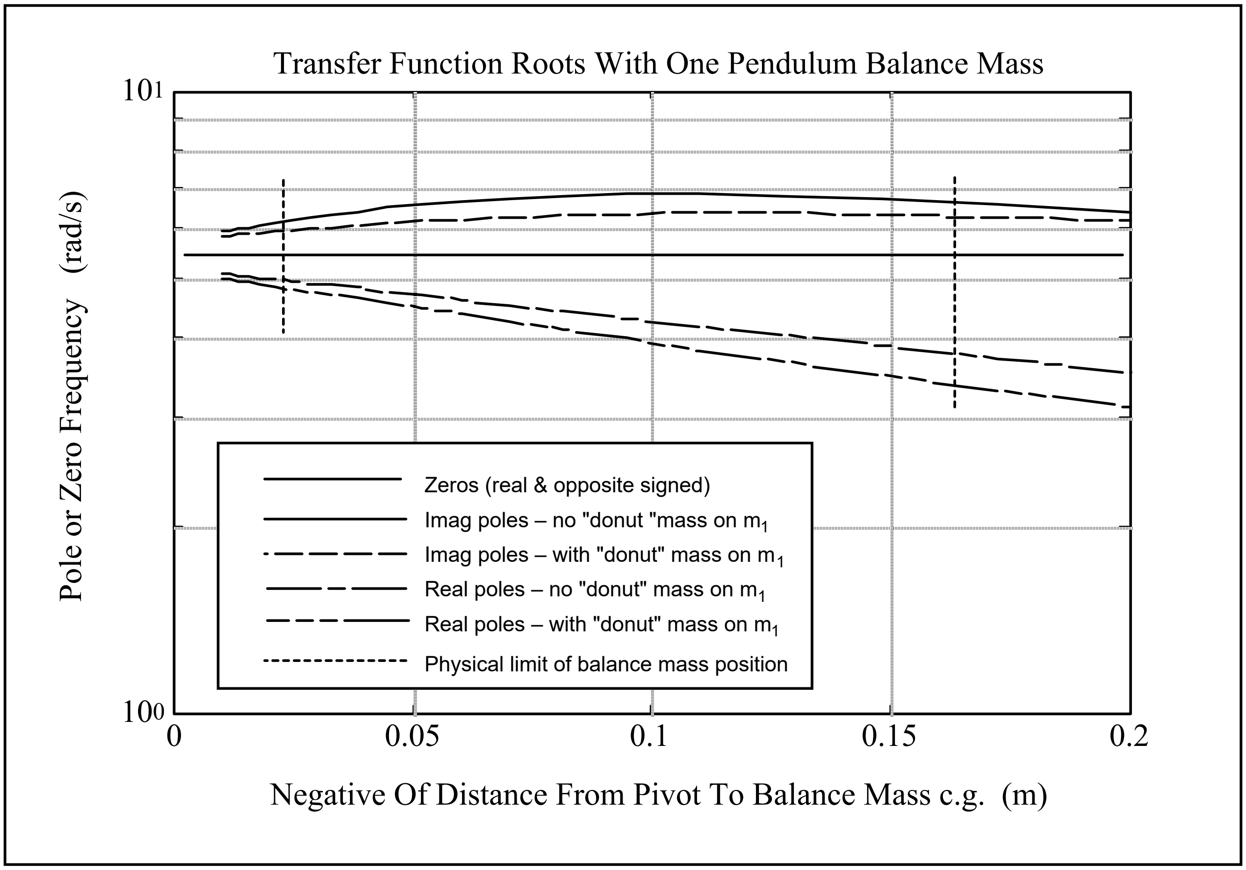


Figure 11.2-2a. Effect of Changing Plant Mass Properties - Single Balance Mass

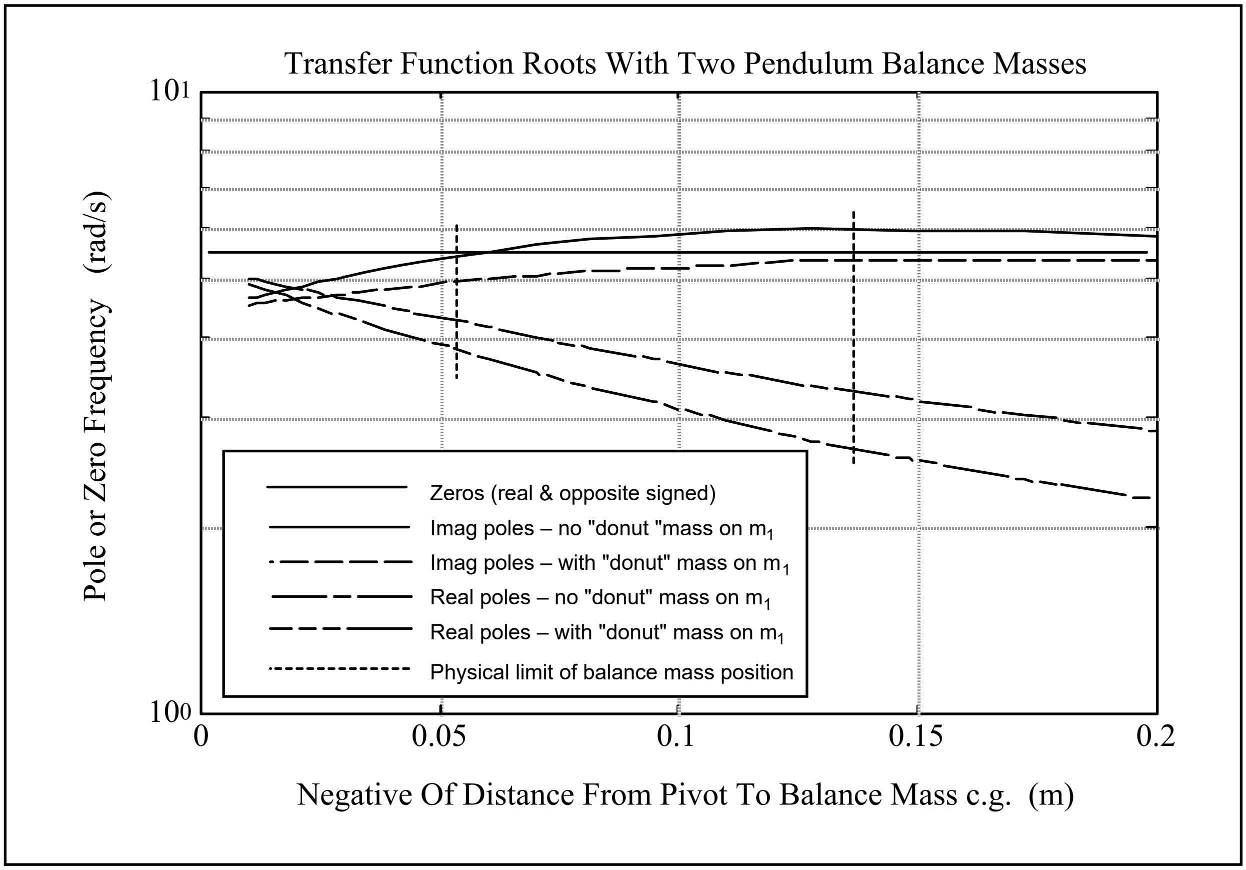


Figure 11.2-2a. Effect of Changing Plant Mass Properties - Two Balance Masses

11.2.3 Mass Property Determination.

Both the nonlinear and linearized time domain equations of motion and the transfer function are fully specified by the parameters m_1, m_2, l_o, l_c, and J^*. For the experimental pendulum (and for a given set of attached brass weights), the parameters m_1, m_2 and l_o are fixed and known. The parameters l_c, and J^* however depend on the location of the movable brass counter-mass. Define l_{c_o} and m_{2_o} to be the value of l_c and m_2 respectively when the counter-mass, m_{w2}, (both brass weights) is removed. Then the parameter l_c is found from a simple center of mass calculation, i.e.:

$$l_c = \frac{m_{w2} l_{w2} + m_{2_0} l_{c_o}}{m_2} \qquad (11.2\text{-}13)$$

where l_{w2} is the distance (signed) from the pivot to the center of m_{w2}.
In our case, the parameters l_{c_o} and m_{2_o} are also fixed and known so that determination of l_c is straightforward. The remaining parameter J^* is found experimentally[29] as described in Chapter 6

[29]More precisely, we will find J_{O_e} and then obtain J_o^* via Eq. 5.1-17.

This chapter outlines experiments which identify the plant parameters, implement several control schemes, and demonstrate important control principles. The versatility of this software / hardware system allows for a broader range of experimental uses than will be described here and the user is encouraged to explore whatever topics and methodologies may be of interest – subject of course to your school and laboratory guidelines and the safety notations of this manual. The safety portion of this manual, experiment 1, must be read and understood by any user prior to operating this equipment.

The instructions in this chapter begin at a high level of detail so that they may be followed without a great deal of familiarity with the PC system interface and become more abbreviated in details of system operation as the chapter progresses. To become more familiar with these operations, it is strongly recommended that the user read Chapter 2 prior to undertaking the operations described here. Remember here, as always, it is recommended to save data and control configuration files regularly to avoid undue work loss should a system fault occur.

System Identification

It is necessary to identify pertinent dynamic and scaling parameters in order to design and implement controllers which stabilize the inverted pendulum plant and allow it to track set-points in angular displacement. In Section 6.1 the nonlinear dynamical equations of the plant were derived. A linearized model of the plant dynamics was also presented in both state-space form and transfer function form. The linearized model can be shown to be an adequate representation in many cases for the purpose of linear control design and implementation (e.g. the controller used as the default controller in Section 3.2 uses a linear controller designed based upon this linearized model). It is recognized that not all parameters may be measured with a fully assembled pendulum. As result, Table 11.1-1 below shows the parameters which are fixed. Their nominal values have been measured prior to assembly:

Table 11.1-1. Mass Property Values

Parameter	Value	Description
L_o	0.330 (m)	Length of pendulum rod from pivot to the sliding rod T section
m_1	TBD† (kg)	Mass of the complete sliding rod including all attached elements.
m_{1o}	0.103 (kg)	Mass of the sliding rod with belt, belt clamps, and rubber end guards (but without sliding rod brass "donut" weights)
m_{w1}	0.110 (kg)	Combined mass of both of the sliding rod brass "donut" weights (=0 if not used)
m_2	TBD (kg)	Mass of the complete assembly minus m_1
m_{w2}	1.000 (kg) (÷2 if only one weight used)	Mass of brass balance weight
m_{2o}	0.785 (kg)	Mass of the complete moving assembly minus m_1 and m_{w2}
L_{co}	0.071 (m)	Position of c.g. of the complete pendulum assembly with the sliding rod and balance weight removed
J_o^*	0.0246 (kg-m²)	$[J_{oe}-m_1 l_o^2]$ evaluated when m_{w2}=0.

†To be determined

From the definitions in the table we have:

$$m_1 = m_{1o} + m_{w1} \qquad (11.1\text{-}1)$$

$$m_2 = m_{2o} + m_{w2} \qquad (11.1\text{-}2)$$

The parameter

$$l_{w2} \triangleq \text{ signed distance from pivot to c.g. of balance mass } (m_{w2})$$

is changeable by the user and is readily measured. That is, referring to Figure 7.1-1,

$$l_{w2} = -\frac{(t+l_t+l_b)}{2} \qquad (11.1\text{-}3)$$

The three remaining parameters – J^*, J_{oe}, and l_c – are derived from the above as follows:

$$l_c = \frac{m_{w2} l_{w2} + m_{2_0} l_{c_o}}{m_2} \quad (11.2\text{-}13)$$

$$J_{o_e} = J_o^* + m_l l_o^2 + m_{w2}(l_{w2})^2 \quad (11.1\text{-}4)$$

$$J^* = \left[J_{o_e} - m_1 l_o^2\right] \quad (11.1\text{-}17)$$

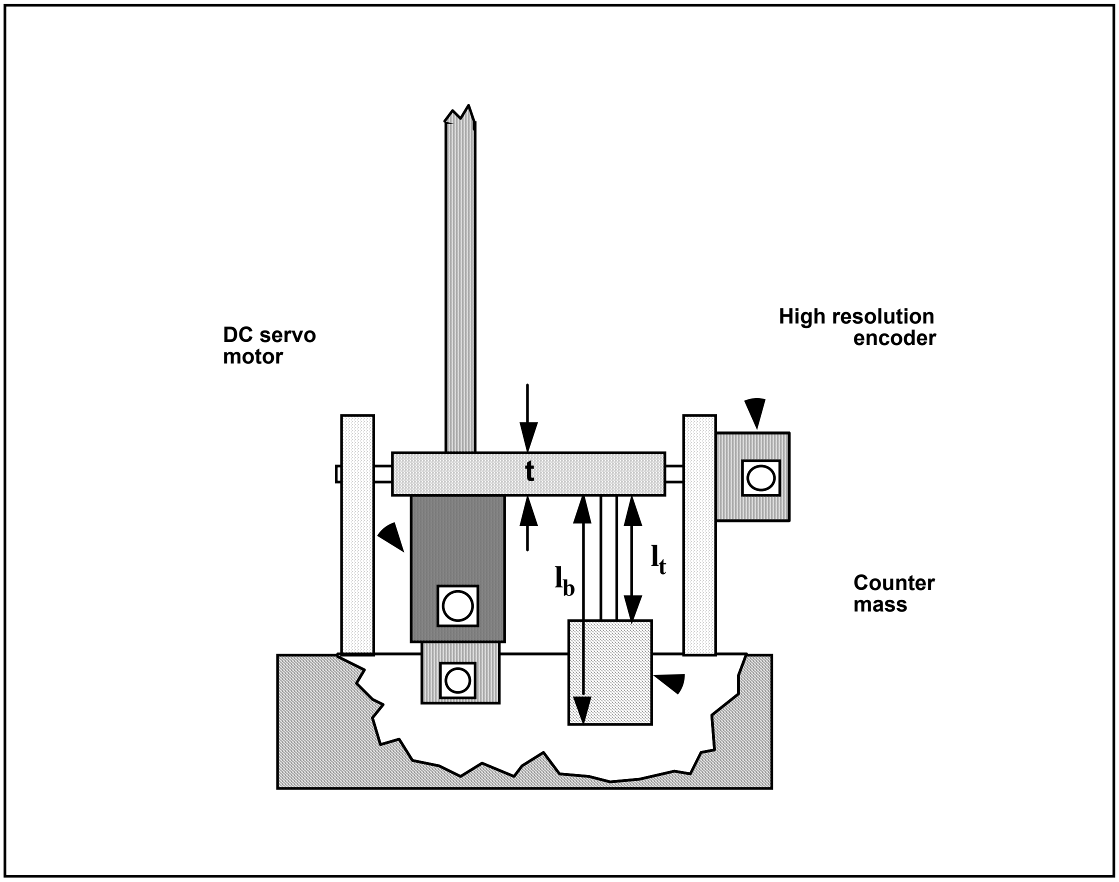


Figure 11.3-1 Measurement of l_{w2}

Plant Configurations For Control Experimentation.

The experiments that follow in this manual use two distinct physical setups and corresponding parameter sets for their models. In each case both "donut" weights are used for the sliding rod and both balance masses are used . For the first case, "Plant #1", l_t = 10.0 cm, and for the second ("Plant #2") l_t = 7.0 cm, .

Based upon the above information you should now be able to calculate the numerical values of the linearized plant parameters as given by Eq's 11.1-18, -19, and -20 . Calculate these values for each plant case. Calculate the roots of the transfer function Eq 11.1-19 and show that the real poles and zeros are mirror images about the imaginary axis on the s-plane and that there is also an imaginary pole pair. Show that the poles of Eq. 11.1-18 are imaginary for plant #1 and are real and opposite for plant #2.[30]

Now we can use the numerical value of plant transfer function or state space model to design controllers. However to actually implement them we need four scaling parameters which are associated with the amplifier/actuator, the two sensors (encoders) and the controller firmware. With reference to the block diagram of Figure 11.3-2, these four scale factors (gains) are as shown in Table 11.1-2 where θ_c and x_c are the respective values of θ and x in units of encoder counts.[31]

Now, all of the above parameters are fixed. The value of k_f is factory set and will not change unless the amplifier analog gains are changed. The value of k_x comes from the fact that encoder 2 generates 2000 counts per revolution which translates to approximately 50,000 counts per one meter travel of the sliding rod. Similarly, k_a is reflects the 16,000 counts per rev. resolution of encoder 1. The value of k_s is fixed by the firmware within the real-time control card.

Construct the properly scaled plant models (with variables θ_c and x_c and control force F_c) in both transfer function and state space form (i.e. Eq's 11.1-16, -18, -19, and -20) by using the following substitutions.[32]

$$\theta = \theta_c / (k_s\, k_a) \qquad (11.1\text{-}5)$$

$$x = x_c / (k_s\, k_x) \qquad (11.1\text{-}6)$$

$$F = F_c\, k_f \qquad (11.1\text{-}7)$$

[30]In Plant #1, the system c.g. – given by $(m_1 l_o + m_2 l_c)/(m_1+m_2)$ – is below the pivot and according to Eq. 5.1-17 results in imaginary poles. Hence in the presence of friction it is stable (for $\theta(s)/x(s)$, not for $\theta(s)/F(s)$). For plant #2 the system c.g. is above the pivot and hence $\theta(s)/x(s)$ is stable.

[31]The control effort gain, k_f, is the product of the DAC gain (V/increment), the amplifier gain (amps/V), the motor torque constant (N-m/amp) and the inverse of the effective belt pulley radius (m^{-1}). The "x" variable feedback gain, k_x, is actually the encoder resolution in counts/radian times the effective belt drive pulley radius in meters. The "θ" variable feedback gain is the corresponding encoder resolution.

The controller gain, k_s, is part of the controller firmware and multiplies all feedback and command input signals prior to their entering the control block (i.e. all inputs to the control law are scaled by 32x). This is for improved numerical resolution in pulse period measurement (to obtain derived rate) which occurs within the controller and is transparent to the user.. Note that for display and plotting purposes, the Executive program divides the corresponding controller-internal values by 32 so that they appear in the original scaling.

[32] For the transfer function models this scaling process is straightforward. For the state space plant model however the scaling affects both the A and B matrices. It may best to first express Eq's 5.1-14 and -15 in terms of the controller scaled variables and then rewrite the expression Eq. 5.1-16.

Table 11.1-2. Scaling Constants

Scale Factor	Value	Description
K_f	0.0013 (N / DAC increment)	Combined DAC-amplifier-motor-pulley gain (see Sect. 4.3)
K_x	50200 (increment / m)	Sliding rod encoder (encoder 2) scale factor
K_a	2546 (increment / rad)	Pendulum angle encoder (encoder 1) scale factor
K_s	32 (controller or DAC incr../ encoder or demand incr..)	DSP controller firmware scale factor

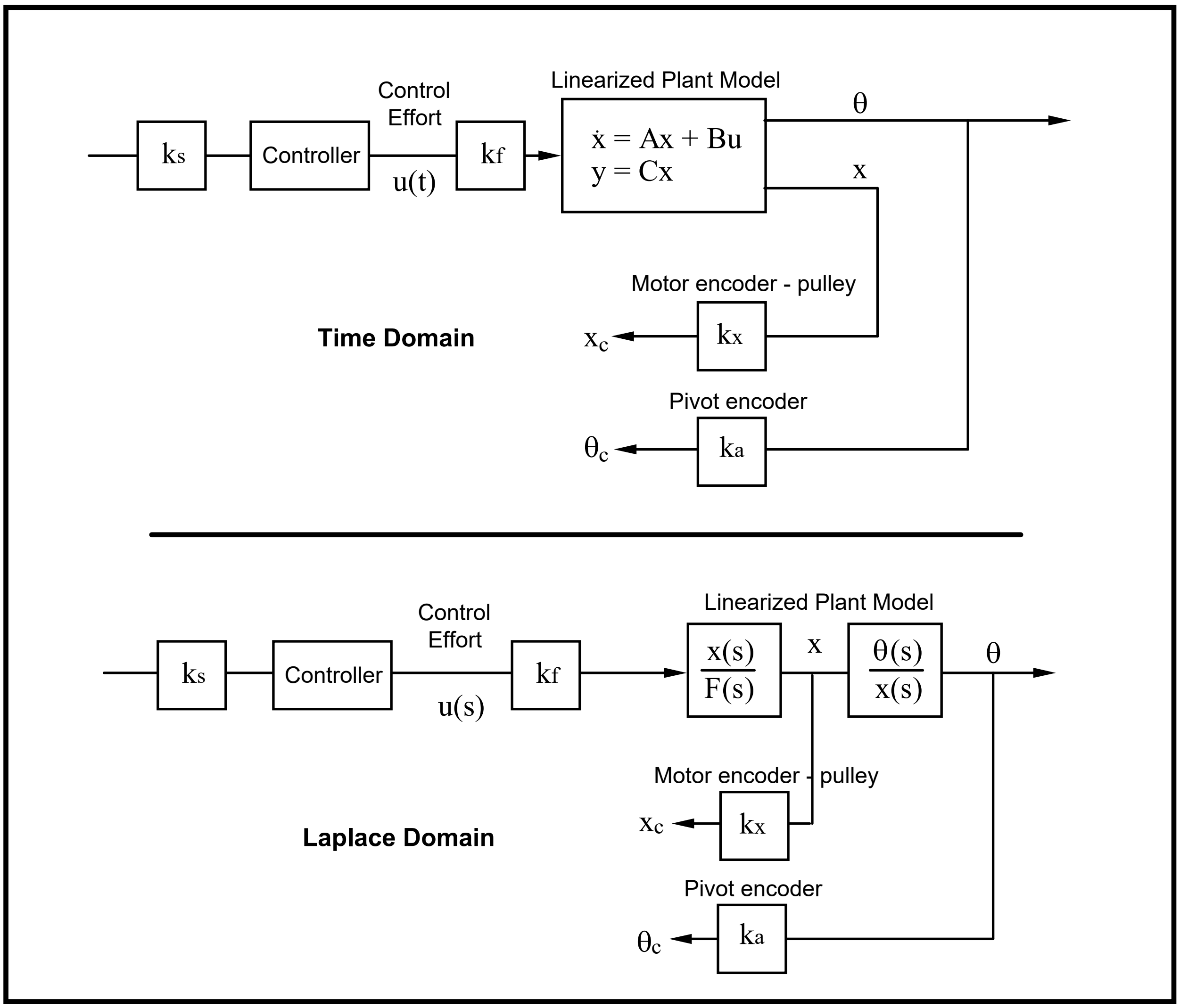


Figure 11.3-2 Plant Block Diagram With Scale Factors

11.3.1 Confirmation of k_a and k_x

These two scale factors can be easily confirmed via a simple procedure using the Executive program

Procedure

1. Remove the power cord from the control box but have all other cables connected. Position the pendulum rod to the right and the sliding rod to the far right at the limit of travel. Enter the Executive program.

2. Select Zero Position in the Utility menu. You should see the position of Encoder 1 and Encoder 2 to be approximately zero.

3 Hold on to the pendulum rod and push the sliding rod from one end until it hits the opposite end of its travel (hits the right limit switch). Now record the number of counts moved by encoder 2 on the Background Screen. With a ruler, measure the distance in meters that the sliding rod has traveled from one limit switch to the other. The ratio of the two is the value of k_x.

4. Now move the pendulum rod in the anti-clockwise direction all the way to the left. Record the number of counts moved by encoder 1 on the Background Screen. Measure the angle of rotation in radians, the ratio of the two is the value of k_a.

11.3.2 Confirmation of J_o^*

The following procedure may be used to confirm the value of J_o^* from which J^* and J_{oe} are obtained via Eq's 11.1-4 and 11.1-17

Procedure:

1. Remove the brass balance weights from the apparatus and install both "donut" weights on the sliding rod. Use a rubber band to restrain the sliding rod in its center of travel as shown in Figure 11.3-2.

2. Disconnect the power cord from the control box but leave the other cables connected. (This allows the encoder signals to pass to the control card but precludes accidental driving of the motor.) Very carefully position the entire pendulum mechanism up-side down on two coplanar flat surfaces such that the pendulum rod is free to rotate as a non-inverted (regular) pendulum. (E.g. two tables side-by-side with approx. 8 in. gap between. It is recommended that a piece of cloth or other soft material be placed between the mechanism and the table on each side to avoid scratching.)

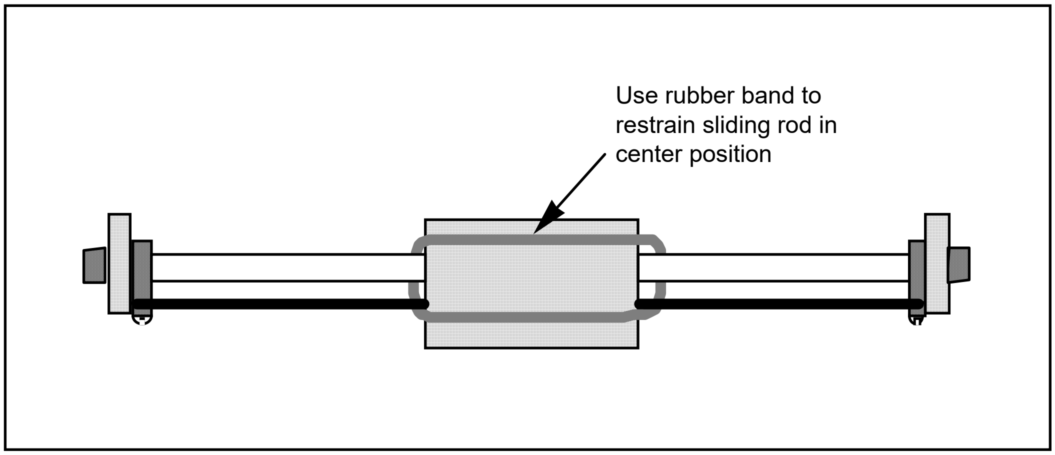


Figure 11.3-2. Securing Sliding rod For J_o^* Measurement via Pendulum Frequency

3. With the controller powered up , enter the Control Algorithm box via the Set-up menu and set $\mathbf{T_s}$ = **0.00442**. Go to Set up Data Acquisition in the Data menu and select encoder #1 as data to acquire and specify data sampling every **5** servo cycles (i.e. every 5 T_s's). Select OK to exit. With the pendulum hanging freely under gravity, select Zero Position from the Utility menu to zero the encoder positions.
4. Check again that the pendulum rod can freely rotate in this position. Now select Trajectory in the Command menu. Enter the Step dialog box and click on step-up. Choosing Open loop Step, input a step size of **0 (zero)**, a duration of **10000 ms** and **repetition** of **1**. Exit to the Background Screen by consecutively selecting **OK**. This puts the controller in a mode for acquiring 20 sec of data on command but without driving the actuator. This procedure may be repeated and the duration adjusted to vary the data acquisition period.
5. Select Execute from the Command menu. Manually displace the pendulum rod approximately 20 deg. from vertical and let go. You should notice that the "non-inverted" pendulum rod will oscillate and slowly attenuate while the encoder data is collected to record this response. Select OK after data is uploaded.
6. Select Set-up Plot from the Plotting menu and choose encoder #1 position; then select Plot Data from the Plotting menu. You should see the pendulum rod angle encoder response similar to one shown in Figure 7.1-2.

Figure 11.3-2 Determination of J_o^* via Pendulum Period Measurement

7. Measure the period of oscillation, T, in seconds by taking the time for completion of several cycles divided by the number of cycles. Confirm that its value for the factory default setting of the balance weight (all the way down the lead screw but just clearing the pocket hole) is approximately 1.25 seconds[33]. Use the following classic linearized "non-inverted" equation of motion to derive an expression for the inertia J in terms of the measured T.

$$J\ddot{\theta} + ml_{cg}g\theta = 0 \qquad (11.1\text{-}8)$$

   where m, J and l_{cg} are the respective mass, inertia, and length to center of gravity (from the pivot) of the combined pendulum elements during the test.

8. Calculate the values of m and l_{cg} for the test case and hence obtain J from your derived expression. (Note that J here is the same as J_{oe} of this manual's notation) Now determine J_o^* according to $J_o^* = J - m_1 l_o^2$ and hence verify the value shown in Table 11.1-1.

9. Carefully re-orient the mechanism to the regular "inverted pendulum" position (up-side up) and reconnect the power cord to the control box. (Make certain that the control box is turned off however.) This concludes the confirmation procedure for J^*.

From the above exercises all plant parameters and scaling constants have been found or verified.

[33] For more accurate measurement, you may use tabular data via Export Raw Data in the Data menu to export data to a text editor where precise numerical values may be read.

Experiment # 12

PD Control Of The Sliding Rod

In this experiment we first close the loop about the "x" position with a relatively high bandwidth (close tracking) PD control. We then make the assumption that the sliding rod closely follows its internal demand u_x so that for designing a controller for θ, the "plant" is approximated by the transfer function $\theta_c(s)/x_c(s)$. The block diagram for this approach is given in Figure 12.2-3.

12.2.1 PD Control Of The Sliding Rod

A simplified dynamical model of the relationship between the sliding rod and the applied force, F when the plant is at the equilibrium position is shown in Figure 6.2-1. Here the inertia of the pendulum assembly is shown as an equivalent mass m_2^* at the sliding rod interface according to:

$$m_2^* = J^* / l_o^2 \qquad (12.2\text{-}1)$$

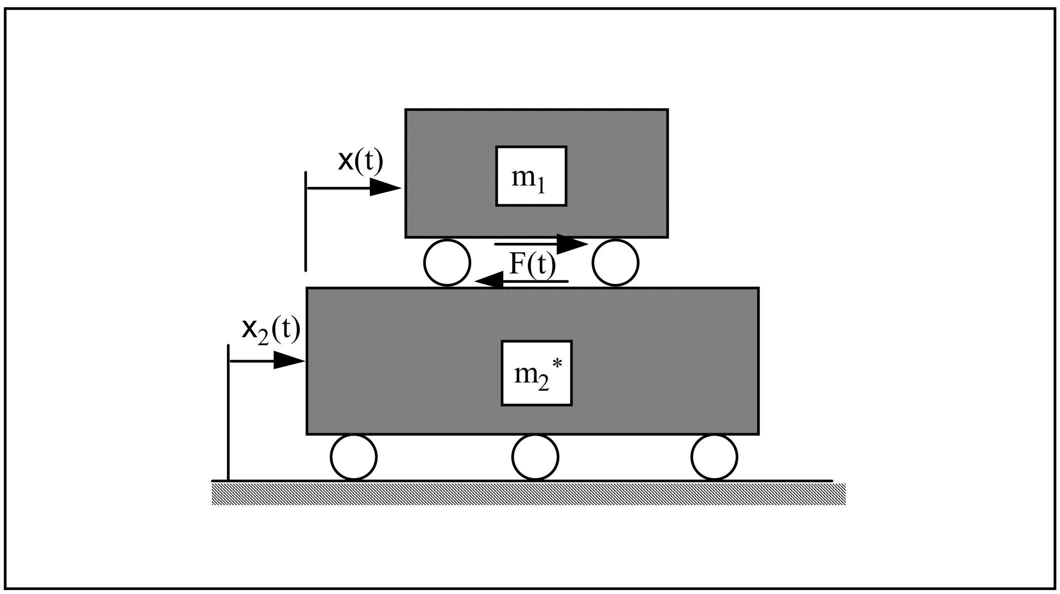


Figure 12.2-1 Simplified Model Of Sliding Rod Dynamics When Plant Is At Equilibrium

The reader should verify that the following relationship holds for the motion x(t):

$$F(t) = m^* \ddot{x} \qquad (12.2\text{-}2)$$

where

$$m^* = \frac{m_1 m_2^*}{m_1 + m_2^*} \qquad (12.2\text{-}3)$$

Thus the reduced dynamics are that of a simple rigid body. For our particular plant, the corresponding transfer function in controller units is:

$$\frac{x_c(s)}{F_c(s)} = \frac{k_{hw}}{m_1^* s^2} \tag{12.2-4}$$

where

$$k_{hw} \triangleq k_x k_s k_f \tag{12.2-5}$$

The block diagram of the proportional plus derivative (PD) control of a rigid body is shown in Figure 12.2-2 and has the closed loop transfer function:[34]

$$c(s) = \frac{x(s)}{r(s)} = \frac{\left(k_{hw}/m_1^*\right)(k_d s + k_p)}{s^2 + \left(k_{hw}/m_1^*\right)(k_d s + k_p)} \tag{12.2-6}$$

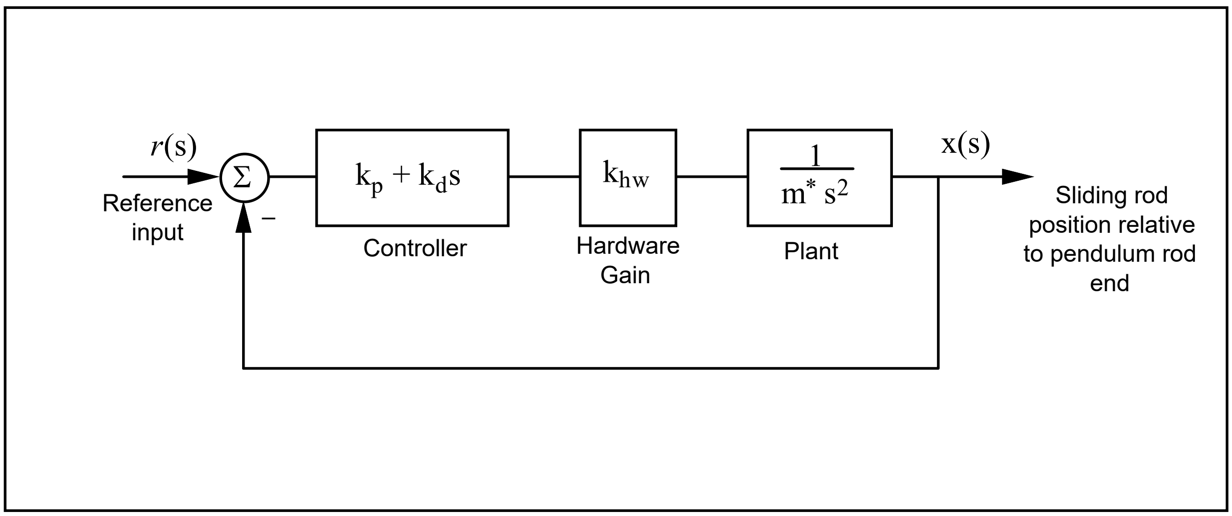


Figure 12.2-2. Rigid Body PID Control – Control Block Diagram

By defining:

$$\omega_n \triangleq \sqrt{\frac{k_p k_{hw}}{m^*}} \tag{12.2-7}$$

$$\zeta \triangleq \frac{k_d k_{hw}}{2m^* \omega_n} = \frac{k_d k_{hw}}{2\sqrt{m^* k_p k_{hw}}} \tag{12.2-8}$$

we may express:

---

[34]Here we omit the subscript "c" since $x_c(s)/r_c(s) = x(s)/r(s)$.

$$c(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad (12.2\text{-}9)$$

Here ω_n is the *natural frequency* and ζ is the *damping ratio* of the closed loop system x(s)/F(s).

Procedure :

PD Control

1. Using the results of Section 11.1 for Plant #1 determine the values of the parameters k_{hw} and m_1^*.
2. Using Equations 12.2-7 and -8, determine values of the control parameters k_p and k_d to provide a closed loop natural frequency of 10 Hz (63 rad/s) and damping ratio of 1 (*critically damped*). Note: if you obtain $0.15 < k_p < 0.35$ or $0.004 < k_d < 0.012$ you should check and correct your calculations.
3. Adjust the position of the balance masses to l_t = 10.0 cm(see Figure 11.3-1) being sure to secure them on the threaded rod by counter rotating them. Verify that the donut weights are in place and secure on the sliding rod. Hence place the apparatus in the Plant #1 configuration.
4. In the Executive program, set-up to collect Encoder #2 and Commanded Position information via the Set-up Data Acquisition box in the Data menu with data sampling every two sample periods. Enter the Command menu, go to Trajectory and select Step, Set-up. Select **Closed Loop Step** and input a step size of **1000** counts, a duration of **1000 ms** and **1 repetition**. Exit to the Background Screen by consecutively selecting OK. This puts the controller board firmware in a mode to command a 1000 count step move (about 2 cm.) forward and back with a one second dwell.
5. Enter the Control Algorithm box under Set-up and set **Ts=0.00442 s** and select Continuous Time Control. Select PID and Set-up Algorithm. Enter the k_p and k_d values (k_i = 0) determined in step 2 (again, do not enter values outside of the ranges $0.15 < k_p < 0.35$ or $0.004 < k_d < 0.012$) and select **Encoder #2** for feedback, then OK.

   Orient the mechanism with the sliding rod in its approximate middle of travel so that the pendulum rod is approximately vertical. In this and all future work, be sure to stay clear of the mechanism before doing the next step. Selecting Implement Algorithm immediately implements the specified controller; if there is an instability or large control signal[35], the plant may react violently. If the system appears stable after implementing the controller, first displace it with a light, non sharp object (e.g. a plastic ruler) to verify stability prior to touching

[35]E.g. a large error at the time of implementation.

plant. AT ALL TIMES KEEP HEAD AND ESPECIALLY EYES WELL CLEAR OF THE APPARATUS

Select Implement Algorithm, then OK.

6. Enter Execute under Command and – again staying clear of the apparatus –select Run. You should see the sliding rod move rapidly about 2 cm back and forth while the pendulum rod swings due to the reactive force.
7. Plot encoder #2 output and commanded position, both on the left axis (see step 6, Section 11.1.2). You should see a critically damped response with 90% rise time of approximately 60 ms. Print your plot via Print Data under the Plotting menu.

12.2.2 Pole Placement Control of θ(s)/x(s)

Having closed a relatively high bandwidth (≈10 Hz) loop about the sliding rod position we utilize the fact that the transfer function of Eq. 12.2-9 has near unity input/output gain (and near zero phase) through the bandwidth (≈ 0.75 Hz) that we will attempt to attain in the overall control of θ. Thus for the control of θ we consider the block diagram of Figure 12.2-3.[36]

[36]Note again that $\theta/r = q_c/r_c$.

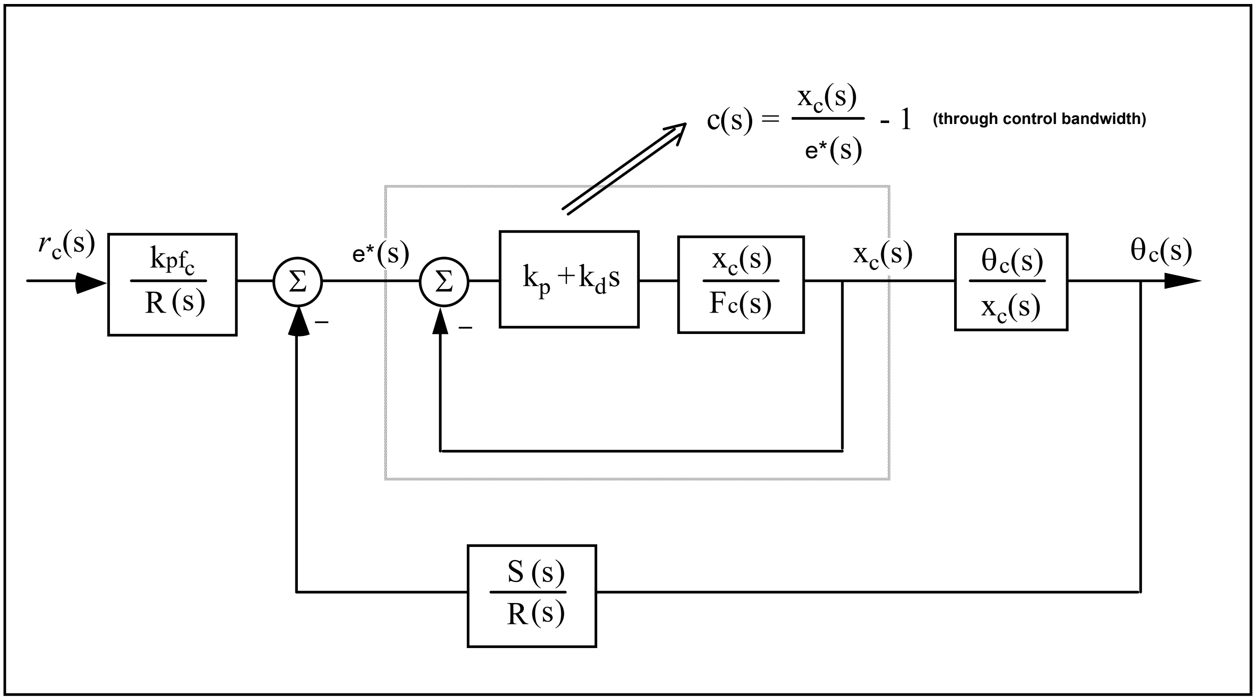


Figure 12.2-3. Control Structure For Outer Loop Closure

Now the plant to be controlled is:

$$\frac{\theta_c(s)}{x_c(s)} = \frac{k_a m_1 l_o}{k_x J_{o_e}} \frac{-s^2 + g/l_o}{s^2 - (m_1 l_o + m_2 l_c) g / J_{o_e}} \triangleq k^* \frac{N_{ax}(s)}{D_{ax}(s)} \qquad (12.2\text{-}10)$$

where the obvious substitutions are made in obtaining the right hand expression. The numerical values of the parameters in this expression were determined in Experiment #1.

We now seek to find a controller S(s)/R(s) which will result in a prescribed set of closed loop poles. The closed loop denominator will have the form:

$$D_{cl}(s) = D_{ax}(s)R(s) + k^* N_{ax}(s)S(s) \qquad (12.2\text{-}11)$$

which may be expressed as[37]

[37] The notation here is the obvious one. The constant k^* has been incorporated in the n_i's.

$$D_{cl}(s) = (d_2 s^2 + d_1 s + d_0)(r_1 s + r_0) + (n_2 s^2 + n_1 s + n_0)(s_1 s + s_0) \qquad (12.2\text{-}12)$$

where the d_i's and n_i's are known from the plant model Eq. 12.2-10

By linear system theory, for coprime $N_{ax}(s)$, $D_{ax}(s)$ with $N_{ax}(s)/D_{ax}(s)$ proper, there exists an $(n-1)^{th}$ order S(s), R(s) which when convolved as per Eq. (6.2-11) form an arbitrary $(2n-1)^{th}$ order $D_{cl}(s)$ where n is the order of $D_{ax}(s)$.

Here we shall solve for the desired denominator:

$$D_{cl}(s) = (s+\pi+j\pi)(s+\pi-j\pi)(s+3\pi) \qquad (12.2\text{-}13)$$

I.e. closed loop poles at $-0.5 \pm j0.5$ and -1.5 Hz.

Procedure :

Pole Placement Control

1. Determine the coefficients of the controller polynomials S(s) and R(s) by equating coefficients in the expanded forms of Eq's 12.2-12 and 12.2-13.
2. Calculate the scalar prefilter gain k_{pf_c} by referring to Figure 12.2-3 and using the scaled variable set θ_c, and x_c,. The goal is to have the output $\theta_c(s)$ (or $\theta(s)$) scaled equal to the input $r_c(s)$ (or $r(s)$) Hint: Consider the system in static equilibrium. Set $\theta_c = 1$ and $r_c = 1$; determine the required input $x_c(s)$ to the plant $\theta_c(s) / x_c(s)$ and hence the necessary signal at the left hand input to the left most summing junction in Figure 12.2-3.
4. In the Executive program, set-up to collect Encoder #1, Encoder #2 and Commanded Position information[38] via Set-up Data Acquisition with data sampling every two sample periods. Enter the Command menu, go to Trajectory and select Step, Set-up. Select **Closed Loop Step** and input a step size of **500** counts, a duration of **2500 ms** and **1 repetition**. Exit to the Background Screen by consecutively selecting OK. This puts the controller board firmware in a mode to command a 500 count step move (about 11 deg.) forward and back with a 2.5 second dwell.
5. Enter the Control Algorithm box under Set-up and set[39] **T_s=0.00884 s** and select Continuous Time Control. Select General Form and Set-up Algorithm and enter the k_p and k_d values (k_i = 0) from the previous section (Section 12.2.1) as the coefficients e_o and e_1 respectively. Enter the coefficients for S(s) and R(s)

[38]You may also select Control Effort if you wish to later observe this value when plotting system responses.

[39]The sample period is set higher here than for the PD inner loop to reduce noise propagation in the closed loop system. A filter shall be used for this purpose in later experiments.

determined in step1. Enter the value k_{pf} calculated in step 2 as t_o in the General Form. Make sure that Encoder #1 is selected for Loop #1 and Encoder #2 for Loop #2. Select OK to exit to the controller selection dialog box.

6. Always do the following to avoid transients when implementing your controller. Exit to the background screen. Select Abort Control to make certain that the pendulum is safe to touch. Set the sliding rod at its approximate center of travel (pendulum rod approximately vertical). Check that the controller box is powered on and turn on if it is not. Select Reset Controller from the Utility menu. Do not disturb the pendulum.

7. Reenter the Control Algorithm box and while staying clear of the mechanism select Implement Algorithm. If the pendulum reacts violently you have implemented an unstable controller or otherwise improperly entered the control coefficients and you will need to repeat the above steps as appropriate. If the pendulum is well behaved you may (again, while staying clear of the pendulum) perturb the pendulum rod gently with a long blunt object such as a ruler. You should see the sliding rod move in an attempt to regulate against the disturbance. You may notice some "twitching" or buzzing due to noise in the system. This is to be expected. The time under closed loop control should be minimized however to reduce the stress on the drive components (In the next experiment, a filter is designed to reduce the noise.)

8. Enter Jog Position under Utility and input an absolute jog of -250 counts. You should see the pendulum move clockwise about 5.5 degrees.

9. Enter Execute under Command and – again staying clear of the apparatus –select Run. You should see the pendulum – actuated by the sliding rod – move rapidly about 11 deg back and forth. Open the loop by selecting "Abort Control".

10. Plot encoder #1 output and commanded position, both on the left axis. Note the initial negative going motion of the pendulum. This is a time response characteristic of nonminimum phase systems. Print your plot via Print Data under the Plotting menu. Save your control configuration via Save Settings in the File menu.

Experiment # 13

Low Pass Filter Augmentation Of Controller

It may have been noticed in the previous experiment that the pendulum appeared to "jitter" or "twitch" as a result of quantization noise which is always amplified by numerical differentiation. To overcome this problem a low pass filter may be used which should have a break frequency much higher than the closed loop bandwidth (so that system stability is not significantly affected) and yet low enough to effectively attenuate the noise. In this section we shall investigate the effect of a simple first order lag with unity dc gain.

Procedure

1. Verify that a first order filter represented by the transfer function G(s) = 1/(τs+1) has a low-pass frequency response with bandwidth 1/τ (rad/s).

2. Consider the filter given by:

$$G(s) = 1/(0.008s+1)$$

   What is the cut-off frequency of this filter and how does its value compare to the magnitude of the closed loop poles of the design in the previous section?

3. To implement this filter as a cascade to the output of the control blocks used in experiment 7.2, we must use the General Form structure within the Setup Control Algorithm Dialog Box. Enter the ECP Executive program and in Setup Control Algorithm, enter the sampling period T_s=0.001768[40]. Input control coefficients in the General Form box from the design of experiment 7.2 (you may do this quickly by uploading your saved configuration from the last experiment via Load Settings in the File menu – make sure however that you choose T_s=0.001768) Now input g_1= 0.008, g_0=1. This corresponds to the implementation of the above first order low pass filter to the output of the control action from the PD inner loop (i.e. immediately prior to the control signal output from the controller and being put on the DAC). Now exit the General form box and after following the instructions of step 6 of Section 11.2.2 and staying clear of the mechanism, Implement the algorithm. You should see a considerable reduction in jittering (i.e. numerical noise) from your previous controller's operation. Why should this filter be implemented to the output as opposed to the input of the controller? Remember it is the numerical noise that we are trying to attenuate not sensor noise. Is sensor noise a source of concern with optical shaft encoders as used in the pendulum mechanism? Execute and plot a step response as per steps 4, 6,7,8, 9, and 10 of experiment 7.2.2.

[40]This relatively fast sampling rate is used to minimize sampled data effects.

4. Repeat the procedures in steps 2 & 3 reducing the time constant of the filter progressively (e.g. to 75% of its existing value each time) from 0.008 seconds keeping sampling time the same. You should note that as the time constant gets smaller the effect of filter fades away. Why?

5. Repeat the procedure in steps 2 & 3 increasing the time constant of the filter (e.g. by 150%) progressively from 0.008 seconds keeping the sampling time the same. You should notice that as the time constant of the filter gets larger it starts to affect the underlying dynamic response by increasing the response time to a step input, overshoots and finally instability. Why?

Experiment # 14

Pole Placement Controller For Unstable $\theta(s)/x(s)$

For Plant #1, the denominator of $\theta(s)/x(s)$ has left half plane roots and is stable[41]. The physical interpretation is that the system c.g. is below the pivot so that if the sliding rod were fixed, the mechanism would behave as an ordinary (noninverted) pendulum. In Plant #2, however, the balance masses are raised bringing the c.g. above the pivot and $\theta(s)/x(s)$ is unstable. Stabilizing and controlling such a system graphically demonstrates the effectiveness of closed loop control.

Procedure:

1. Adjust the position of the balance masses to l_t = 7.0 cm being sure to secure them on the threaded rod by counter rotating them. Verify that the donut weights are in place and secure on the sliding rod. Hence place the apparatus in the Plant #2 configuration.
2. Repeat the design and implementation process of experiment 6.2, noting the following:
   a) You may assume that the PD controller designed previously experiment has similar natural frequency and damping (you may verify this by recalculating m^* and then ω_n and ζ for the new mass properties) when applied here and hence you may use the same k_p & k_d.
   b) Use the same desired closed loop denominator given by Eq. (11.2-13).
   c) In initializing the system per step 6 in Section 11.2.2 it will be more difficult to manually balance the pendulum such that you Reset Controller then Implement Algorithm without the pendulum moving. With a little practice however you should be able to do it.
3 You should add a noise filter to your design per experiment 11.3 before implementing.
4. Compare your step response with that obtained in experiment 11.2. How do the closed loop transfer functions compare? (Assume that c(s)=1 throughout the control bandwidth.)
5. Save your controller configuration via Save Settings in the File menu.

---

[41]Recall however that $\theta(s)/Fs)$ is unstable.

Model 750

Gyroscope system

Experiment#15

Gyroscope Dynamics: Nutation and Precession

Background Theory

The gyroscope depicted in Figure 1 consist of 4 (rigid) rotating masses. The 4 rigid bodies each have a angular position θ relative to their rotating gimbal axis and an overview of the definition of the rigid bodies has also been given in Table 1.
For the derivation of a dynamic model we assume the gyroscope to be symmetric and the center of the rigid bodies to all lie at the center of body D (the rotor). As a result, only the rotational dynamics needs to be taken into account. For the rotational or angular position θi, $i = 1, 2, 3, 4$ of each rigid body, we adopt the following convention..

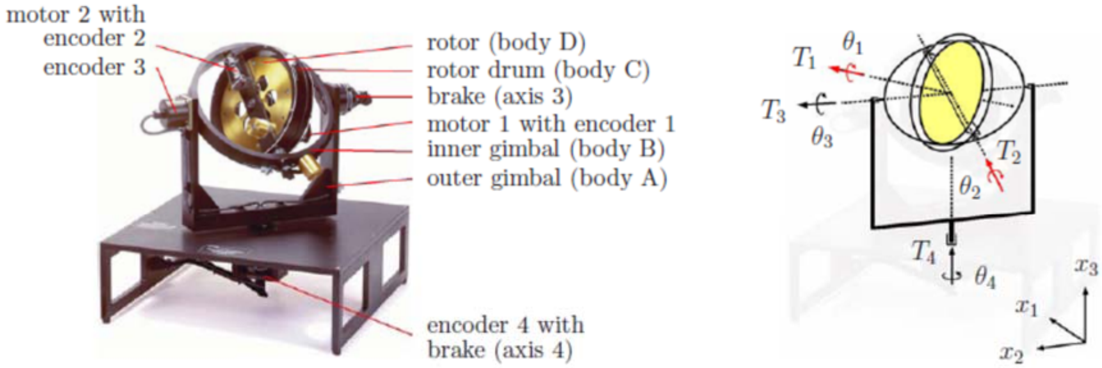


Figure 1: Picture of moment gyroscope (left) and schematics with variables (right).

body	definition	angular position	inertia
A	outer gimbal	θ_4	$\mathbf{I}^A$
B	inner gimbal	θ_3	$\mathbf{I}^B$
C	rotor drum	θ_2	$\mathbf{I}^C$
D	rotor	θ_1	$\mathbf{I}^D$

Table 1: Overview of rigid body elements in gyroscope

15.1.2 Conventions for angular position

• The angular position $\theta 1$ of the rotor (body D) is not of importance. We will only be considering the angular velocity $\omega 1 = \dot{\theta} 1$.
• The angular position $\theta 2$ of the rotor drum (body C) is set to $\theta 2 = 0$ if *the rotor drum (body C) is perpendicular to the inner gimbal (body B)*.
• The angular position $\theta 3$ of the inner gimbal (body B) is set to $\theta 3 = 0$ if *the inner gimbal (body B) is perpendicular to the outer gimbal (body A)*.
• Since the outer gimbal (body A) is able to rotate freely and the gyroscope is assumed to be symmetric, $\theta 4$ can be reset to $\theta 4 = 0$ at *any angular position of the outer gimbal (body A)*.
Since each rigid body might be able to rotate along a 3 dimensional axis, we must consider the inertia I, J and K of each rigid body respectively along $x1$, $x2$ or $x3$ axis. This defines the inertia $\mathbf{I}b$, with $b = A,B,C$ or D in Table 1 as

$$\mathbf{I}^b = \begin{bmatrix} I_b & 0 & 0 \\ 0 & J_b & 0 \\ 0 & 0 & K_b \end{bmatrix}, \ b = \text{body } A, B, C \text{ or } D$$

where Ib denotes the inertia along the $x1$-axis, Jb denotes the inertia along the $x2$ axis and Kb denotes the inertia along the $x3$ axis for $b = A,B,C,D$. Note that only moments of inertia are considered, while products on inertia are considered to be zero due to the symmetric nature of the gyroscope.

The angular position of the 4 rigid bodies in the gyroscope can be changed by 2 internal torques and labeled $T1$ and $T2$ in Figure 1. The 2 internal torques $T1$ and $T2$ are generated by small DC motors that apply a torque to respectively to the rotor (body D) and the rotor drum (body C). Torque $T1$ will make the rotor spin like a wheel around its (perpendicular) axis 1, whereas torque $T2$ will make the rotor drum spin around the (longitudinal) axis 2.

15.1.3 Non-linear dynamics of gyroscope

Since both $T1$ and $T2$ are applied *internally* on a rigid body of the gyroscope, each torque will have a counteracting torque on another rigid body. By inspecting the mechanical connections in the schematics of the gyroscope in Figure 1, the following simple observations can be made:

- Depending on the angular position $\theta 2$ and $\theta 3$, application of the torque $T1$ on the rotor (body D) will for example result in a direct counteracting torque $T3$ causing rotation of the inner gimbal (body B) and/or a direct counteracting torque $T4$ causing rotation of the outer gimbal (body A).
- The angular position of $\theta 2$ can be changed by application of $T2$ on the rotor drum (body C). Depending on the angular position $\theta 3$, application of the torque $T2$ on the rotor drum (body C) will for example result in a direct counteracting torque $T4$ causing rotation of the outer gimbal (body A). Interestingly, when $\omega 1 = \dot{\theta} 1$ _= 0, rotation of the outer gimbal (body A) is possible even if $\theta 3 = 0$ due to an (indirect) moment caused by a Coriolis force (change in angular momentum).

On the basis of these simple observations, it is clear that the relationship between the internal torques $T1$, $T2$ and the angular positions θi and velocities $\omega i = \dot{\theta} i,\ i = 1, 2, 3, 4$ will formulate the equations of motion of the gyroscope. The equations of motions can be derived using Lagrange's equations or Kane's method and will result in a set of coupled (non-linear) differential equations of the form

$$\begin{aligned} T_1 &= f_1(\theta_2, \theta_3, \omega_2, \omega_3, \omega_4, \dot{\omega}_1, \dot{\omega}_3, \dot{\omega}_4) \\ T_2 &= f_2(\theta_2, \theta_3, \omega_1, \omega_3, \omega_4, \dot{\omega}_1, \dot{\omega}_2) \\ 0 &= f_3(\theta_2, \theta_3, \omega_1, \omega_2, \omega_3, \omega_4, \dot{\omega}_1, \dot{\omega}_3, \dot{\omega}_4) \\ 0 &= f_4(\theta_2, \theta_3, \omega_1, \omega_2, \omega_3, \omega_4, \dot{\omega}_1, \dot{\omega}_2, \dot{\omega}_3, \dot{\omega}_4) \end{aligned} \qquad (1)$$

in which the torques $T1$ and $T2$ are considered as input signals. The full derivation of the (non-linear) equations of motion can be found in the Model 750 Control Moment Gyroscope

Manual. It can be noted here that the equations of motion do not depend on $\theta 1$ and $\theta 4$, as the angular position $\theta 1$ of the rotor (body D) and the angular position $\theta 4$ of the outer gimbal (body A) is irrelevant for the dynamic behavior of the gyroscope.

15.1.4 Linearized dynamics of gyroscope

Considering only (small) perturbations around the angular velocity $\omega 1 = \dot{\theta} 1$ of the rotor (body D), the angular position $\theta 2$ of the rotor drum (body C) and the angular position $\theta 3$ of the inner gimbal (body B) allows for a significant simplification of the (non-linear) equations of motion. In case we assume an operating point of the gyroscope with

$$\begin{aligned} \omega_1 &= \Omega \\ \theta_2 &= \bar{\theta}_2 \\ \theta_3 &= \bar{\theta}_3 \end{aligned}$$

the equations of motion in (1) reduce to

$$\begin{aligned} J_D\dot{\omega}_1 &= T_1 - J_D\cos\bar{\theta}_2\dot{\omega}_3 - J_D\sin\bar{\theta}_2\cos\bar{\theta}_3\dot{\omega}_4 \\ (I_C+I_D)\dot{\omega}_2 &= T_2 - J_D\Omega\sin\bar{\theta}_2\omega_3 + J_D\Omega\cos\bar{\theta}_2\cos\bar{\theta}_3\omega_4 + (I_C+I_D)\sin\bar{\theta}_3\dot{\omega}_4 \end{aligned}$$

$$\begin{gathered} (J_B+J_C+J_D-(J_C+J_D-I_D-K_C)\sin^2\bar{\theta}_2)\dot{\omega}_3 = \\ -J_D\cos\bar{\theta}_2\dot{\omega}_1 + J_D\Omega\sin\bar{\theta}_2\omega_2 - J_D\Omega\sin\bar{\theta}_2\sin\bar{\theta}_3\omega_4 - \sin\bar{\theta}_2\cos\bar{\theta}_2\cos\bar{\theta}_3 \end{gathered} \tag{2}$$

$$\begin{gathered} (I_D+K_A+K_B+K_C+(J_C+J_D-I_D-K_C)\sin^2\bar{\theta}_2+ \\ (I_B+I_C-K_B-K_C-(J_C+J_D-I_D-K_C)\sin^2\bar{\theta}_2)\sin^2\bar{\theta}_3)\dot{\omega}_4 = \\ -J_D\sin\bar{\theta}_2\cos\bar{\theta}_3\dot{\omega}_1 - J_D\Omega\cos\bar{\theta}_2\cos\bar{\theta}_3\omega_2 + (I_C+I_D)\sin\bar{\theta}_3\dot{\omega}_2+ \\ J_D\Omega\sin\bar{\theta}_2\sin\bar{\theta}_3\omega_3 - (J_C+J_D-I_D-K_C)\sin\bar{\theta}_2\cos\bar{\theta}_2\cos\bar{\theta}_3\dot{\omega}_3 \end{gathered}$$

Although the equations in (2) look complicated, they have all been written in the form where the inertia times angular acceleration equals the sum of torques:

$$I\ddot{\theta}_i = \sum T$$

reflecting 2nd Newton's law for rotational motion. The coupled set of (linear) differential equations are useful in determining the linear dynamic model of the gyroscope for special cases.

15.1.5 Linear dynamics for special case in laboratory experiment

To further simplify the dynamical model of the gyroscope, we consider several special cases on the basis of an operating point of the gyroscope given by

$$\begin{aligned} \omega_1 &= \Omega \\ \theta_2 &= \bar{\theta}_2 = 0 \\ \theta_3 &= \bar{\theta}_3 = 0 \end{aligned} \tag{3}$$

where zero angles are defined according to the convention defined on page 5 of this laboratory handout. With the operating point defined in (3), special cases of the dynamics of the gyroscope are found by applying some of the (electromechanical) brakes for either axis 3 (rotation of outer gimbal) or axis 4 (rotation of inner gimbal) of the gyroscope.
In case none of the brakes are used, both the inner and outer gimbals are able to rotate freely. With the operating point defined in (3), the linearized equations of motion in (2) reduce to

$$\begin{aligned} J_D\dot{\omega}_1 &= T_1 - J_D\dot{\omega}_3 \\ (I_C + I_D)\dot{\omega}_2 &= T_2 + J_D\Omega\omega_4 \\ (J_B + J_C + J_D)\dot{\omega}_3 &= -J_D\dot{\omega}_1 \\ (I_D + K_A + K_B + K_C)\dot{\omega}_4 &= -J_D\Omega\omega_2 \end{aligned} \tag{4}$$

and formulate a set of couple linear (second order) differential equations. This can also be written as a *set of coupled first order differential equations* by definition of the state vector:

$$x(t) = \begin{bmatrix} \theta_2(t) & \theta_3(t) & \theta_4(t) & \omega_1(t) & \omega_2(t) & \omega_3(t) & \omega_4(t) \end{bmatrix}^T$$
$$\text{and the input : } u(t) = \begin{bmatrix} T_1(t) & T_2(t) \end{bmatrix}^T$$

formulating the state space model

$$\dot{x}(t) = Ax(t) + Bu(t), \text{ with}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{J_D\Omega}{I_C+I_D} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-J_D\Omega}{I_D+K_A+K_B+K_C} & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{J_B+J_C+J_D}{J_D(J_B+J_C)} & 0 \\ 0 & \frac{1}{I_C+I_D} \\ \frac{-1}{J_B+J_C} & 0 \\ 0 & 0 \end{bmatrix} \tag{5}$$

The result is a 7th order state space model. Computing the eigenvalues of the state matrix *A* results in the following pole locations:

- 2 poles at 0 due to the rigid body mode. The rigid body mode is due to free rotation $\theta 3$ of the inner gimbal (body B) as a result of a direct counteracting torque $T3$ caused by torque $T1$ to rotate the rotor (body D).
- 3 additional poles at 0 due to the kinematic differential equations.
- 2 complex poles that models the oscillatory behaviour that couples the rotor drum (body C) rotation with the rotation of the outer gimbal (body A). The resonance

frequency ωn found from the complex pole pair located at $\}j\omega n$ is called the *nutation frequency of the gyroscope* in rad/s.
Instead of writing a state space model (5), the linearized equations (4) can also be used to write a transfer function representation between the angular positions and applied torques via Laplace transform. Some of the resulting transfer functions are

$$\begin{aligned}
\theta_2(s) &= G_{22}(s)T_2(s),\ G_{22}(s) = \frac{I_D + K_A + K_B + K_C}{(I_C + I_D)(I_D + K_A + K_B + K_C)s^2 + \Omega^2 J_D^2} \\
\theta_3(s) &= G_{31}(s)T_1(s),\ G_{31}(s) = -\frac{1}{(J_B + J_C)s^2} = -\frac{K}{s^2} \\
\theta_4(s) &= G_{42}(s)T_2(s),\ G_{42}(s) = \frac{-\Omega J_D}{(I_C + I_D)(I_D + K_A + K_B + K_C)s^3 + \Omega^2 J_D^2 s}
\end{aligned}$$

The transfer function $G31(s)$ is the relation between $\theta 3(s)$ of the inner gymbal (body B) and the applied torque $T1(s)$ on the rotor (body D) and indicates a simple rigid body motion (2 poles at origin) due to free rotation $\theta 3$ of the inner gimbal (body B). Transfer function $G22(s)$ is the relationship between torque $T2$ and the resulting angular position $\theta 2$ of the rotor drum (body C)

This section measures the nutation and precession of the gyroscope as a function of wheel speed. It also demonstrates the damping of nutation through rate feedback at Gimbal #2, and finally, gives another way of interpreting precession in terms of conservation of angular momentum. All tests in this section are performed with the apparatus in the configuration of Figure 8.2-1.
Important Notice: From this point on in the instructions, no specific reminders to perform the required safety procedures shall be given.

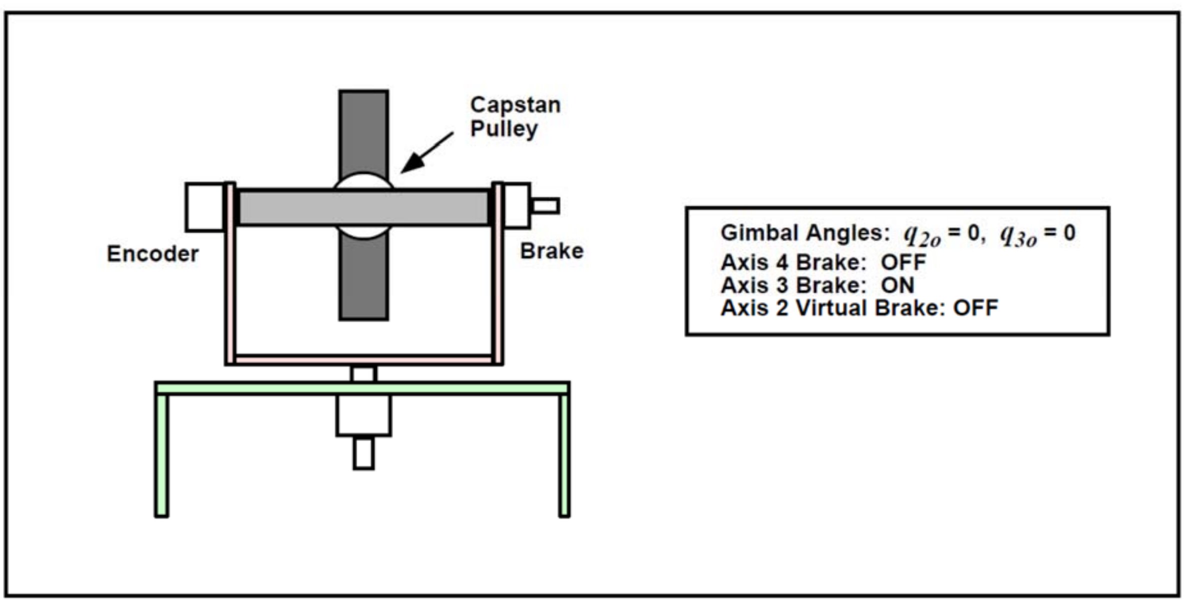


Figure 15.2-1. Configuration For All Tests In This Section.

15.2.1 Nutation: Frequency & Mode Shapes

Procedure

1. Setup the mechanism as shown in Figure 15.2-1.
2. Write a simple real-time algorithm to activate Motor #2 (i.e. put control effort values on the DAC) with a *Control Effort* equal to the (*Commanded Position)*/32.1 Use the global real-time variables "control_effort2" and "cmd1_pos" for this purpose.
3. Go to Trajectory 1 Configuration. Enter Impulse and specify an *Amplitude* of 16000 counts, a *Pulse Width* of 50 ms, a *Dwell Time* of 4000 ms, and 1 *repetition* (this prepares the controller board to input a 16000 count positive-going impulse followed immediately by 4 seconds of zero input during which data is collected.
4. Setup Data Acquisition (Setup menu). Specify *Commanded Position 1, Sensor 2 Position, Sensor 4 Position,* and *Control Effort 2* as data to be acquired with a *Sample Period* of 4 servo cycles
5. Enter the ECP Multivariable Executive program. Implement your algorithm from Step 2 above with sampling period set to Ts = 0.00442 seconds.
6. Initialize Rotor Speed to 200 RPM and zero the encoder positions (Utility menu). Execute the maneuver selecting *Normal Data Sampling* and *Execute Trajectory 1 Only* .
7. Plot the Encoder 2 and Encoder 4 Position data and subsequently the velocity data. Note the frequency of the oscillations and the relative amplitude and phase of the Encoder 4 response verses the Encoder 2 response. Save your plots.
8. Disable the rotor speed loop (Command menu). You may also want to turn off the Control Box to more rapidly decelerate the rotor. Wait for the rotor to stop (if you turned off the Control Box, turn it back on at this point). Repeat Steps 6 and 7 for a rotor speed of 400 RPM.
9. Repeat Step 8 for a rotor speed of 800 RPM.

8.2.2 Precession

Procedure

10. Repeat Steps 1 through 6 of the Section 15.2.1 except in Step 3 setup the *Impulse* trajectory for an *Amplitude* of 6000 counts, a *Pulse Width* of 8000 ms, a *Dwell Time* of 0 ms, and 1 *repetition* (this prepares the controller board to input a 6000 count constant input for 8 seconds). The first maneuver should be at 200 RPM and should result in a initial transient series of attenuating nutation oscillations followed by a steady state response. Plot the position data for Encoders 2 and 4 and also their velocity data. Save your plots.
11. Repeat Step 10 for the 400 and 800 RPM cases. Note the change in steady state velocity for Encoder 4 with rotor speed.

15.2.3 Nutation Damping

Procedure

12. Augment your algorithm from Step 1 of Section 8.2.1 to add rate feedback damping at Axis 2. I.e. add a term u_{2damp} of the form

$$u_{2damp} = -k_v q_2 s$$

where *kv* is the rate feedback gain. You may use the backwards difference transformation to implement discrete time differentiation according to

$$s - \frac{1-z^{-1}}{T_s} \qquad 15.2\text{-}8$$

where *Ts* is the sampling period. Use *Ts* =0.00884 s. in this experiment.

Have your laboratory supervisor review and approve your algorithms before proceeding..

13. Set the rotor speed to 400 RPM. Beginning with a value of *kv* = 0.005 Implement your algorithm with *Ts* =0.00884 seconds.

14. Repeat Step 10 except maintain the rotor speed at 400 RPM rotor speed. Do you see a reduction in the nutation mode amplitude?

15. Repeat Steps 13 and 14 for various increasing values of *kv*. Do not exceed kv=0.10, higher values could lead to excessive numerical noise and damage to the system! How are the nutation oscillations affected by increased rate feedback gain? Save your plot of a case where the nutation is well damped.

A. From the plotted data of Section 15.2.1, measure the frequency of the nutation mode for the three rotor speeds. (Hint: divide the number of cycles considered [typically between 2 and 5] by the time taken to complete them. Zoom the plot if necessary or export the raw numerical data to get precise readings and make sure that you start and end the evaluation period at the same phase in the respective cycles.) Compare your result with that predicted by the theory (i.e. the eigenvalues) or equivalently the characteristic roots . Assuming the measured and provided moments of inertia are within 10% of their actual values, are your results in agreement with the theory? What is the relationship between rotor speed and nutation frequency?

B. Consider the position plots versus those of velocity for the data of Section 15.2.1. For a given test, are the oscillation frequencies the same? Are the relative amplitudes of the outputs at encoders 2 and 4 the same? Are the steady state values the same? Explain your answers (you may neglect the effects of friction).

C. Solve for the eigenvectors in the system matrix . Solve for the homogeneous motion solution (Hint: isolate the real and imaginary parts of the homogeneous solution and consider only the real part.) Measure and report the relative amplitudes and phasing of the axis 2 and 4 outputs from the tests of Section 15.2.1. How well do your results compare with theory? What is the relationship between rotor speed and nutation mode shape?

D. Determine the precession rate (steady sate value of ω_4) for each of the rotor speeds tested. (You may wish to consider the change in position, *q4* divided by the time taken. This will generally result in greater precision than reading the velocity data directly.) How does this compare with that predicted by the gyroscopic relationship $T = \omega \times H$? (Recall that in this case, $T=u2ku2$). What is the relationship between rotor speed and precession rate?

E. What is the effect of rate feedback at axis 2 on the nutation mode? Plot a root locus of this system where the closed loop roots are plotted as $k_v k_{u2} k_{e2}$ is varied of *kv*?

Experiment 16

Matlab, Simulink, state feedback control

To introduce the use of Matlab in control system design

Introduction:
Matlab is a software system for doing numerical mathematics. The basic Matlab software is structured for convenient and rapid solution of problems in numerical linear algebra. In addition there are a number of toolboxes, which adapt Matlab to special tasks. We will study the basic Matlab package and the Control System Toolbox.
The easiest way to get started on Matlab is to run demos 1 and 2 of the Matlab DEMO program, run the Matlab HELP programs and then start playing around with simple examples.

Procedure:

1. The basic Matlab system
    a. Throughout this and the other experiments the symbol <cr> means the ENTER or RETURN key on the keyboard.
    b. At the DOS prompt type **matlab<cr>** and then wait for the Matlab prompt, which is >>. At the Matlab prompt type **demo<cr>**. Follow the instructions to observe Demos 1) and 2). (Observe others if you want)
    c. At the Matlab prompt type **help<cr>** and note the available operators and functions. Next type **help sin<cr>** to see the type of help available for individual functions. Next try others.
    d. Enter a vector by typing x = [1 2 3]. Next type x'. Now create a matrix by typing A = [x' x' x']. Find determinant of this matrix. Enter another matrix A that is not singular. Calculate A * x. Solve A * y = x for y.
    e. Generate the vector t = 0:10. Repeat for t = 0:0.1:1.
    f. Generate the logspaced vector w = **logspace**(-1,2) and note the entries. Read help logspace.

2. Matlab Graphics
    a. Plot several cycles of a sinusoid. Put axis labels and a title on your plot.
    b. Define a random vector and plot it.
    c. Define two random vectors x and y and make a point plot of x vs y.
    d. Print out one of your plots on printer.

3. The control system toolbox
    a. Type **ctrldemo<cr>** and observe the demo.
    b. Type help **rlocus<cr>**, help bode<cr> and help step<cr>
    c. Plot the locus of the roots of $(s^3 + 6s^2 + 5s) + K (s^2 + 6s + 18) = 0$ for 0 δ Kδ 200. On your plot indicate roots locations for K = 10. Re-scale your root locus plot showing only the second quadrant.(Hint: see help axis. Let $(s^2 + 6s + 18)$ as **num** and $(s^3 + 6s^2 + 5s)$ as **den**.)

4. Control system Example: A unity feedback control system has loop gain function
   L(s)= 200 (s + 5) /[s (s+1) (s+20)(s + 30)].
    a. Plot a bode plot for L (jw).
    b. Plot a Nyquist plot for L (jw).
    c. Plot the closed loop poles and zeros on the sgrid (Type **help sgrid<cr>**.)
    d. Plot the closed loop step response. Put on axis labels and a title and make a hardcopy. Put your name and date on your plot using the Matlab function text (see **help text <cr>**.)
    e. Generate a discrete time transfer function L (z) using the Toolbox function c2d

5. (do help c2d <cr>). Use a sampling period $T_s = 0.01$ sec.
    a. Plot the poles and zeros of the closed loop discrete time system on zgrid.
    b.

6. Simulink Example 1:
    a. Wire simulation for two-integrator feedback system shown in fig 0.1.
    b. Run simulation.
    c. Plot unit step response.
    d. Replace first integrator by the transfer function 2 / (s + 3) as shown in fig 0.2 and select the two feedback gains a and b to place closed loop poles at -1 + j2 and
7. -1 - j2.
    a. Plot unit step response of your design.

8. Simulink example 2:
    a. Wire the simulink diagram of the Lorentz attractor shown in the simulink help
9. write up
    a. Plot 3D response of the system.

Postlab Report:
Turn in hardcopy of step response generated in part 4.d and 5.e. and calculations involved in 5.d. Also turn in the 3D plot requested in 6.b. above.

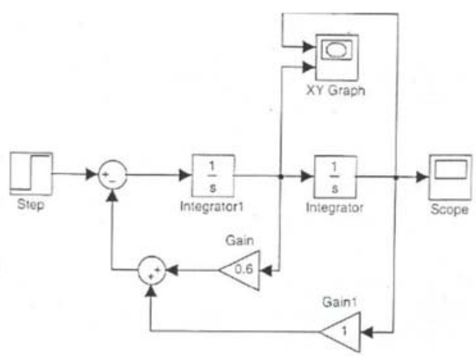


Fig 0.1: Two integrator example.

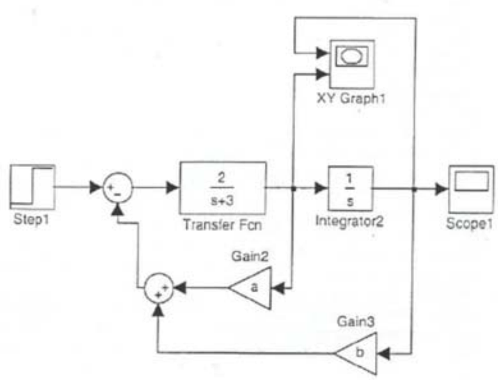


Fig 0.2: Integrator replaced by transfer function.

To get started, type one of these: helpwin, helpdesk, or demo. For product information, visit www.mathworks.com.
» help simulink
Simulink
Version 7.0 (R11)
Model analysis and construction functions. Simulation.

sim - Simulate a Simulink model.
sldebug - Debug a Simulink model.
simset - Define options to SIM Options structure.
simget - Get SIM Options structure
Linearization and trimming.
linmod - Extract linear model from continuous-time system.
linmod2 - Extract linear model, advanced method.
dlinmod - Extract linear model from discrete-time system.
trim - Find steady-state operating point.

Model Construction.

close_system - Close open model or block.
new_system - Create new empty model window.
open_system - Open existing model or block.
load_system - Load existing model without making model visible.
save_system - Save an open model.
add_block - Add new block.
add_line - Add new line.
delete_block - Remove block.
delete_line - Remove line.
find_system - Search a model.
replace_block - Replace existing blocks with a new block.
set_param - Set parameter values for model or block.
get_param - Get simulation parameter values from model.
bdclose - Close a Simulink window.
bdroot - Root level model name.
gcb - Get the name of the current block.
gcbh - Get the handle of the current block.
gcs - Get the name of the current system.
getfullname - get the full path name of a block
slupdate - Update older 1.x models to 3.x.
addterms - Add terminators to unconnected ports.
bool - Convert numeric array to boolean. Masking.
hasmask - Check for mask.
hasmaskdlg - Check for mask dialog.
hasmaskicon - Check for mask icon.
iconedit - Design block icons using ginput function.
Maskpopups - Return and change masked block's popup menu items.
movemask - Restructure masked built-in blocks as masked subsystems.

Library.

libinfo - Get library information for a system.

Hardcopy and printing.
frameedit - Edit print frames for annotated model printouts.
print - Print graph or Simulink system; or save graph to M-file.
printopt - Printer defaults.
orient - Set paper orientation.

See also BLOCKS and SIMDEMOS.
.
» help step
STEP Step response of LTI models.
STEP(SYS) plots the step response of the LTI model SYS (created with either TF, ZPK, or SS). For multi-input models, independent step commands are applied to each input channel. The time range and number of points are chosen automatically. STEP(SYS,TFINAL) simulates the step response from t=0 to the final time t=TFINAL. For discrete-time models with unspecified sampling time, TFINAL is interpreted as the number of samples. STEP(SYS,T) uses the user-supplied time vector T for simulation. For discrete-time models, T should be of the form Ti:Ts:Tf where Ts is the sample time. For continuous-time models, T should be of the form Ti:dt:Tf where dt will become the sample time for the discrete approximation to the continuous system. The step input is always assumed to start at t=0 (regardless of Ti). STEP(SYS1,SYS2,...,T) plots the step response of multiple LTI models SYS1,SYS2,... on a single plot. The time vector T is optional. You can also specify a color, line style, and marker for each system, as in
step(sys1,'r',sys2,'y--',sys3,'gx').
When invoked with left-hand arguments,
[Y,T] = STEP(SYS)
returns the output response Y and the time vector T used for simulation. No plot is drawn on the screen. If SYS has NY outputs and NU inputs, and LT=length(T), Y is an array of size [LT NY NU] where Y(:,:,j) gives the step response of the j-th input channel.

For state-space models,
[Y,T,X] = STEP(SYS)
also returns the state trajectory X which is an LT-by-NX-by-NU array if SYS has NX states.
See also IMPULSE, INITIAL, LSIM, LTIVIEW, LTIMODELS. Overloaded methods
help lti/step.m

help frd/step.m

DISPLAY FORMATS

Formats	format short (default) format short *e* format long, and format long *e*

CONSTANTS

Identity matrix, n x n	eye(n)
Zero matrix, n x n	zeros(n)
Matrix of all ones, n x n	ones(n) eye(A), zeros(A), ones(A) generate the corresponding matrix the same size as A
Value of π	in variable named pi

OPERATORS

Arithmetic	+ - * / \ ^ for scalars and matrices where applicable; for element-by-element operation, precede symbol with a period (.*, ./, .\, and .^ only)
Conjugate transpose	Denoted by prime as in A'; if A is real then A' is A transpose
For matrices	diag(A) extracts the diagonal and stores vector. triu(A) extracts the upper triangular part; tril (A) extracts the lower triangular part
Norms	norm(x, l), norm (x), norm (x, 'inf'), x a vector; norm(A,1), norm(A), norm(A, 'inf'), A a matrix

SOLVING SYSTEMS

Solving Systems	To solve Ax = b, type x=A\b after A and b are entered

TIME RESPONSE	
step	step response
dstep	discrete step response
initial	continuous initial condition response
dinitial	discrete initial condition response
impulse	impulse response
dimpulse	discrete unit sample response
lsim	continuous simulation to arbitrary inputs
dlsim	discrete simulation to arbitrary inputs
filter	SISO z-transform simulation (see main guide)
ltitr	low level time response function

FREQUENCY RESPONSE	
bode	Bode plots
dbode	discrete Bode plots
fbode	fast Bode response of continuous systems
margin	gain and phase margins
nichols	Nichols plots
dnichols	discrete Nichols plots
ngrid	grid lines for Nichols plot
nyquist	Nyquist plots
dnyquist	discrete Nyquist plots
sigma	continuous singular value frequency plots
dsigma	discrete singular value frequency plots
freqz	z-transform frequency response (see main guide)
freqs	Laplace-transform frequency response (see main guide)
ltifr	low level frequency response function

nargchk	check number of M-file arguments
perpxy	find nearest perpendicular point
poly2str	convert polynomial to string
printmat	print matrix with row and column labels
ric	Riccati equation residual calculation
schord	ordered schur decomposition
tfchk	check consistency of transfer functions
timvec	auto-ranging algorithm for continuous time response
tzreduce	reduce system when computing transmission zeros
unwrap	unwrap phase for Bode plots (see main guide)
vsort	sort complex eigenvalues for root locus

ROOT LOCUS	
rlocus	Evans root-locus
rlocfind	interactive root locus gain determination
sgrid	continuous root locus ω_n, ζ grid
zgrid	discrete root locus ω_n, ζ grid
pzmap	pole-zero map

GAIN SELECTION	
lqr	linear-quadratic regulator design
lqr2	linear quadratic regulator design using Schur method
lqry	regulator design with weighting on the outputs
dlqr	discrete linear-quadratic regulator design
dlqry	discrete regulator design with weighting on the outputs
lqe	linear-quadratic estimator design
lqe2	linear quadratic estimator design using Schur method
lqew	general linear-quadratic estimator design
dlqe	discrete linear-quadratic estimator design
dlqew	general discrete linear quadratic estimator design
lqrd	discrete regulator design from continuous cost function
lqed	discrete estimator design from continuous cost function
acker	SISO pole placement
place	pole placement

UTILITY	
abcdchk	check the consistency of an (A,B,C,D) set
chop	round to n significant figures
dexresp	discrete example response function
dfrqint	auto-ranging algorithm for discrete Bode response
dfrqint2	auto-ranging algorithm for discrete Nyquist response
dmulresp	discrete multivariable response function
dtimvec	auto-ranging algorithm for discrete time response
distsl	distance to straight line
dric	discrete Riccati equation residual calculation
exresp	example response function
freqint	auto-ranging algorithm for Bode frequency response
freqint2	auto-ranging algorithm for Nyquist frequency response
freqresp	low level frequency response function
givens	Givens rotation matrix
housh	construct Householder transformation
mulresp	multivariable response function