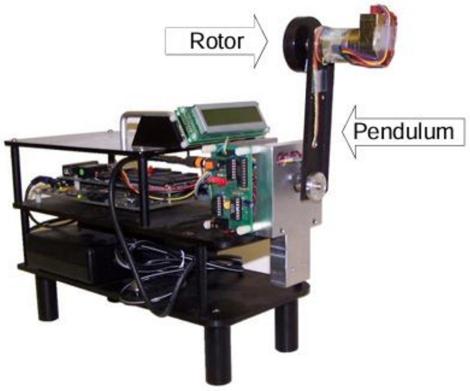
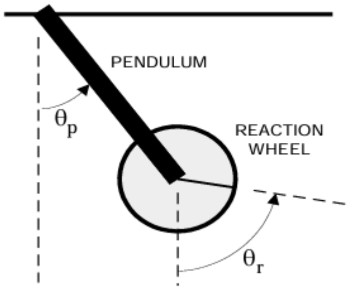
Paul Osuma, Krish Sahni

ECE 486: Final Project Report
Spring 2024
TA: Srushti Manjunath
Lab Section: Thursday 3pm

Introduction

In this lab, we aim to control a Reaction Wheel pendulum system by leveraging the effects of Newton’s 3rd law of action reaction and relying on state space models for system representation. The system mainly consists of a pendulum arm and a motor with a rotor attached at the end of the pendulum. Both the pendulum and the motor have their individual optical encoders. The encoders measure the relative angle of rotation between the fixed and the moving part of the subsystems. At equilibrium, the pendulum arm seen below lies pointing down due to gravity. In this configuration the RWP exhibits stable equilibrium as either small torque applied to it will result in the pendulum ending at the start position. We seek to generate a torque on the pendulum using the motor reaction and simultaneously regulate the position of equilibrium for the RWP. We use Matlab for our computations.





Mathematical Model

To derive the equations of motions for our system, we utilize the Lagrangian approach. With this approach we deal with the scalar energies of the system as opposed to the vector forces which makes our equation derivations much simpler. Finally, we linearize the obtained equations.

The system has 2 degrees of freedom (DOF). We use generalized coordinates for each DOF, q_1 for the pendulum (p) and q_2 for the motor r. The system variables are provided in the lab sheet.

q_1: $L_{q1} = KE_{q1} - PE_{q1} = = \frac{1}{2}\mathrm{J}\dot{\theta}_\mathrm{p}^2 - \mathrm{m_p}\,\mathrm{g}\,\mathrm{l}\cos\theta_\mathrm{p}$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q_1}}\right) - \frac{\partial L}{\partial \dot{q_1}} = \tau_1 = -ki$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_p}}\right) - \frac{\partial L}{\partial \theta_p} = \tau_1 = -ki$$

$$\frac{\mathrm{J}}{\mathrm{J}}\ddot{\theta}_\mathrm{p} + \mathrm{m_p}\,\mathrm{g}\frac{\mathrm{l}}{\mathrm{J}}\sin\theta_\mathrm{p} = -\frac{ki}{J}$$

$$but\ \mathrm{m_p}\,\mathrm{g}\frac{\mathrm{l}}{\mathrm{J}} = w_{np}^2$$

$$\ddot{\theta}_p + w_{np}^2\sin\theta_p = -\frac{ki}{J} \quad \dots\dots\dots\dots\dots(i)$$

q_2: $L_{q2} \;=\; KE_{q2} - PE_{q2} == \frac{1}{2}\,\mathrm{J_r}\dot{\theta}_{\mathrm{r}}^2 - 0$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) - \frac{\partial L}{\partial q_2} = \tau_2 = ki$$

$$J_r\ddot{\theta}_r = ki$$

$$\ddot{\theta}_r = \frac{ki}{J_r} \qquad \dots\dots\dots\dots\dots\dots(ii)$$

Linearization

(i) State space form: $\theta_1 = \theta_p, \quad \theta_2 = \dot{\theta}_p$

$$\dot{\theta}_2 = f_2(\theta_1, \theta_2, i) = \ddot{\theta}_p = -w_{np}^2 \sin\theta_1 - \frac{ki}{J}$$

$$linearize\ f_2\ around\ equilibrium\ (\theta_1, \theta_2, i) = (\pi, 0, 0)$$

$$\frac{\partial f_2}{\partial \theta_1} = -w_{np}^2 \cos\theta_1 \qquad \frac{\partial f_2}{\partial \theta_2} = 0 \qquad \frac{\partial f_2}{\partial i} = -\frac{k}{j}$$

Evaluate at equilibrium:

$$\frac{\partial f_2}{\partial \theta_1} = w_{np}^2 \qquad \frac{\partial f_2}{\partial \theta_2} = 0 \qquad \frac{\partial f_2}{\partial i} = -\frac{k}{j}$$

$$\dot{\theta}_2 = w_{np}^2\theta_1 - \frac{ki}{J}$$

(i) State space form: $\theta_3 = \theta_r, \quad \theta_4 = \dot{\theta}_r$

$\dot{\theta}_4 = \ddot{\theta}_r = \frac{ki}{J_r}$ is already linearized

State Space

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} = \begin{bmatrix} 0\ 1\ 0\ 0 \\ w_{np}^2\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{k}{J} \\ 0 \\ \frac{k}{J_r} \end{bmatrix} i$$

$$\begin{bmatrix} \dot{\theta}_p \\ \ddot{\theta}_p \\ \dot{\theta}_r \\ \ddot{\theta}_r \end{bmatrix} = \begin{bmatrix} 0\ 1\ 0\ 0 \\ w_{np}^2\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0 \end{bmatrix} \begin{bmatrix} \theta_p \\ \dot{\theta}_p \\ \theta_r \\ \dot{\theta}_r \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{k}{J} \\ 0 \\ \frac{k}{J_r} \end{bmatrix} i$$

Introducing the variables $a = w_{np}^2, b_p = \frac{k_u}{J}, b_r = \frac{k_u}{J_r}$, we end with the following state space.

$$\begin{bmatrix} \dot{\theta}_p \\ \ddot{\theta}_p \\ \dot{\theta}_r \\ \ddot{\theta}_r \end{bmatrix} = \begin{bmatrix} 0\ 1\ 0\ 0 \\ a\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0 \end{bmatrix} \begin{bmatrix} \theta_p \\ \dot{\theta}_p \\ \theta_r \\ \dot{\theta}_r \end{bmatrix} + \begin{bmatrix} 0 \\ -b_p \\ 0 \\ b_r \end{bmatrix} u$$

Full State Feedback Control with Friction Compensation

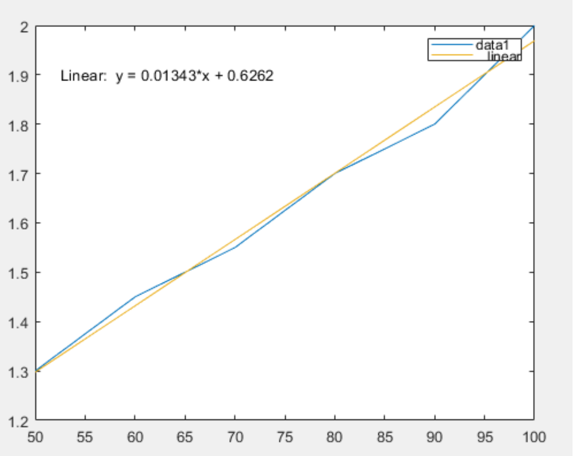
1. Development of PD Control with Friction Compensation

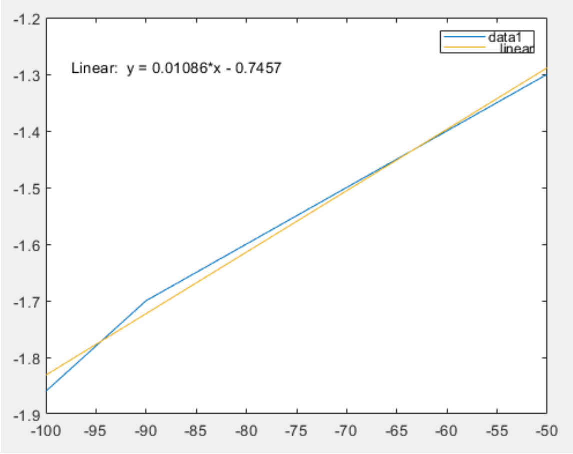
In the development of our PD (Proportional-Derivative) control strategy for the Reaction Wheel Pendulum (RWP), we focused on integrating friction compensation to enhance system response and stability. This approach was based on the necessity to account for the frictional forces that inherently oppose the motion of the pendulum, potentially destabilizing the system and introducing steady-state errors.

The derivation of our PD control, including both two-state and three-state feedback controllers, was guided by the principles outlined in Chapter 4 of our course material. Friction compensation was specifically addressed during the system identification phase described in Chapter 2. By conducting open-loop tests at various speeds and directions, we systematically recorded the control efforts required to maintain predetermined motor speeds. This experimental setup allowed us to model friction as a linear function of the angular velocity.

The process involved maintaining the pendulum at a fixed angle while varying the motor speed. The linear relationship between the required control effort and the angular velocity was then used to derive the friction compensator values. We implemented the friction compensation by adjusting the controller's feedback gains to counteract the measured frictional forces, thereby improving the controller's performance.

The following are the linearized graphs we generated using the data we found.





Graph 1: plotting the data collected for w > 0
Graph 2: plotting the data collected for w < 0

Friction Compensation Equations for Positive and Negative Angular Velocities:

$$y = 0.01343\omega + 0.6262 \text{ for } \omega > 0$$
$$y = 0.01086\omega - 0.7457 \text{ for } \omega < 0$$

From these graphs, we can clearly tell the frictional coefficients are the following:

$$b^{+} = 0.1343 \quad c^{+} = 0.6262 \quad b^{-} = 0.01086 \quad c^{-} = -0.7457$$

To design a proportional controller, we started with the following:

$$t_r = 0.2s \quad u(t) = 100 * \frac{1(t)rad}{s} \quad b_r = \frac{198rad}{s} \quad \Omega(s) = \frac{br}{s}$$

Laplace domain:

$$u(t) = -k\big(w_r - \omega(t)\big) \rightarrow \ \Omega(s) = -k\left(\frac{w_r}{\mathrm{s}} - \omega(t)\right)$$

$$Simplify\text{: } \Omega(s) = -\frac{kbr\omega r}{s(s + kbr)}$$

This gives the following:

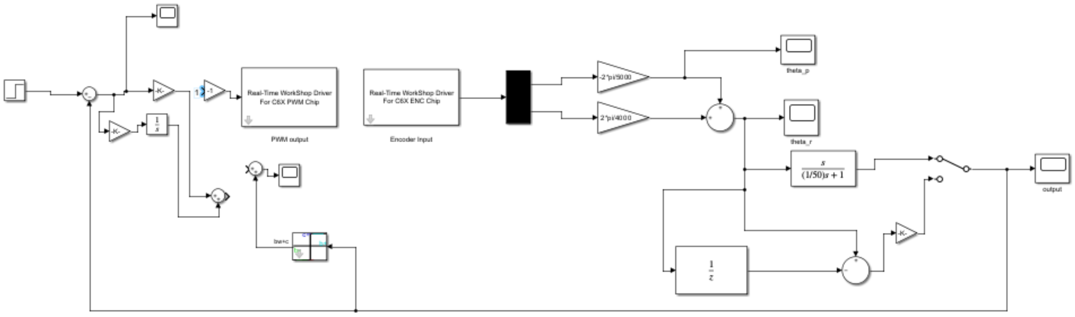
$$r\big(1 - e^{-k_{br}t_r}\big) = 0.9r \quad \text{and} \quad e^{-k_{br}0.2} = 0.1$$

$$-k_{br}0.2 = \ln(0.1)$$

Simplifying gives:

$$k = -\frac{\ln(0.1)}{0.2b_r} = -\frac{\ln(0.1)}{0.2 \times 198} \approx 0.0581$$

In the Windows Target implementation, friction compensation reduced the pendulum's swing around the equilibrium position by allowing the controller to apply less torque. Below is the block diagram of our setup.



Impact of Friction Compensation

Friction compensation proved to be significantly beneficial for the stability and performance of our state feedback controller. By accurately compensating for the frictional forces, the controller could maintain the pendulum in the inverted position with reduced oscillations around the equilibrium point. The compensation allowed for a smoother transition between control states and reduced the overall energy consumption by minimizing unnecessary control efforts.

This enhancement was particularly evident during the testing phase, where the compensated system demonstrated a superior ability to reject disturbances and maintain stability compared to the uncompensated setup. The friction compensator values, calculated based on empirical data, enabled precise tuning of the controller, which was critical in achieving an optimal balance between responsiveness and stability.

The effect of PI and friction compensation can be seen in the following images.

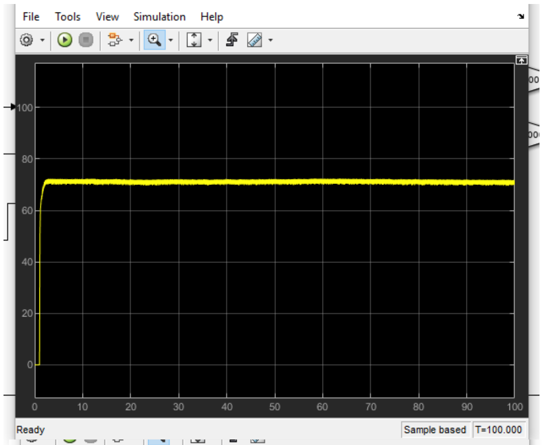


Figure: No compensation. We can see that the plot does not even reach the expected output of a 100 due to no control.

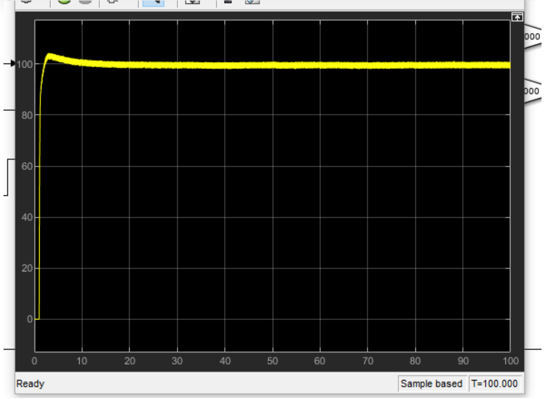


Figure: We can see that with PI and frictional compensation, the graph is much better.

In the development of our three-state feedback controller, the primary objective was to regulate the velocity while disregarding positional control. This approach stabilizes the rotor's velocity without impacting its positional dynamics, enabling us to ignore θr′ by setting its control gain to zero. The entire system is maintained as a four-dimensional matrix for simplicity.

We configured the system to stabilize around the equilibrium point θp = π, defining δθp as the deviation from this point. Below is the mathematical formulation used:

$$k = -\frac{\ln(0.1)}{0.2b_r} = -\frac{\ln(0.1)}{0.2 \times 198} = 0.0581$$

$$\delta\theta_p = \theta_p - \pi \text{ and } \delta\theta_r = \theta_r$$

Experimental procedures helped us determine the necessary system parameters:

$$b_p = 1.0951, \quad b_r = 206.5608, \quad a = 70.1741$$

For stability analysis, we applied $-Kx$ as the system input and computed the eigenvalues of the matrix $A - BK$. These eigenvalues derived using MATLAB, all lie in the left half-plane, confirming system stability.

The system's stability and the definition of $\delta\theta p$ ensure that at steady state as time approaches infinity, θp converges to π and θr to zero, thus validating the stability of the system's upward equilibrium.

The state-space representation and feedback mechanism are outlined as follows:

$$\begin{bmatrix} \dot{\theta}_p \\ \ddot{\theta}_p \\ \dot{\theta}_r \\ \ddot{\theta}_r \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_p \\ \dot{\theta}_p \\ \theta_r \\ \dot{\theta}_r \end{bmatrix} + \begin{bmatrix} 0 \\ -b_p \\ 0 \\ b_r \end{bmatrix} u \qquad x = Ax + Bu = (A - BK)x$$

$$\text{A-BK} = \begin{bmatrix} 0\ 1\ 0\ 0 \\ 68.32\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0 \end{bmatrix} - 10^4 \begin{bmatrix} 0\ 0\ 0\ 0 \\ 0.0228\ 0.0026\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ -4.3121 - 0.4953 - 0.0005 - 0.0008 \end{bmatrix} =$$

$$10^4 \begin{bmatrix} 0\ 0.0001\ 0\ 0 \\ -0.016 - 0.0026\ 0\ 0 \\ 0\ 0\ 0\ 0.0001 \\ 4.3121\ 0.4953\ 0.007\ 0.0008 \end{bmatrix}$$

We know know that since $\delta\theta_p$ is defined as $\theta_p - \pi$, the equilibrium point occurs at $\theta_p = \pi$ as described below. As time goes infinity, at steady state, x'=0 and this implies that θ_p goes to pi and θ_r goes to zero as shown below. Hence the position at the upward equilibrium is stable.

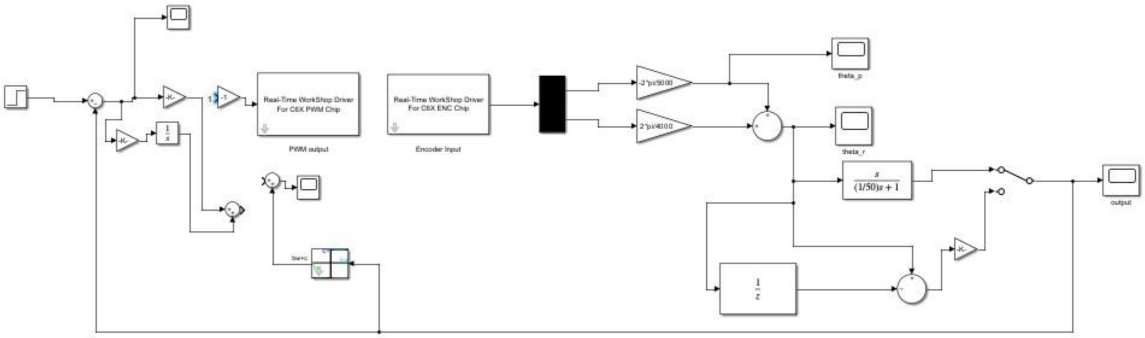
$$\delta\theta_p = \theta_p - \pi \rightarrow \theta_p = \pi$$

$$\delta\theta_r = \theta_r = 0 \rightarrow \theta_r = 0$$

$$\begin{bmatrix} \delta\theta_p \\ \delta\dot{\theta}_p \\ \delta\theta_r \\ \delta\dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \pi \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The adjustments to A−BK reflect the implementation of the feedback parameters, ensuring the system's response aligns with the theoretical model. The final setup is visualized in the provided block diagram.

Table of Values



From our Simulink simulations of the RWP, the maximum IC deviation, pulse disturbance, and constant perturbation that the controller can stabilize are given in the table below.

	Two state feedback		Three state feedback		Observer	
Max IC deviations	$\delta\theta_p$	$\delta\dot{\theta}_p$	$\delta\theta_p$	$\delta\dot{\theta}_p$	$\delta\theta_p$	$\delta\dot{\theta}_p$
	0.13	1.2	0.12	1.1		
Max pulse	8		8.8			
Max disturbance	5.1		7.7			

Behavior of System

Using the same controller K values as the one for the animation did not immediately result into stabilization of our RWP. We had to slightly adjust the K4 value which controlled the angular velocity of the rotor. Subtracting 0.5 from the original K4 value helped us realize a stable RWP. The behavior we saw in the system was that at the inverted position, the rotor stopped spinning and the pendulum held up momentarily before gravity made it fall back again. When that happened, the rotor would begin to rapidly spin to generate a torque on the pendulum that would cause it to go up to its equilibrium position. At that point it was difficult for the RWP to stay at an upright position because of the generated angular momentum.

Full State Feedback Control with Decoupled Observer

Why are observers used

In full state feedback u = -Kx is not implementable therefore, in order to provide an accurate estimate of the position and velocity states $\hat{x}$, an observer is used. Given our state space:

$$\begin{bmatrix}\dot{\theta}_p\\ \ddot{\theta}_p\\ \dot{\theta}_r\\ \ddot{\theta}_r\end{bmatrix} = \begin{bmatrix}0\,1\,0\,0\\ a\,0\,0\,0\\ 0\,0\,0\,1\\ 0\,0\,0\,0\end{bmatrix}\begin{bmatrix}\theta_p\\ \dot{\theta}_p\\ \theta_r\\ \dot{\theta}_r\end{bmatrix} + \begin{bmatrix}0\\ -b_p\\ 0\\ b_r\end{bmatrix} u$$

We can decouple the 4 state observer into 2-state observers because our state space matrix A above is of a block diagonal form. We can break it into:

$$\begin{bmatrix}\dot{\theta}_{12}\\ \dot{\theta}_{34}\end{bmatrix} = \begin{bmatrix}J & 0\\ 0 & K\end{bmatrix}\begin{bmatrix}\theta_{12}\\ \theta_{34}\end{bmatrix} + \begin{bmatrix}M\\ N\end{bmatrix} u$$

$$\dot{\theta}_{12} = \begin{bmatrix}0 & 1\\ a & 0\end{bmatrix}\theta_{12} + \begin{bmatrix}0\\ -b_p\end{bmatrix} u$$

$$C_{12} = [1\ 0]$$

$$\dot{\theta}_{34} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\theta_{34} + \begin{bmatrix}0\\ b_r\end{bmatrix} u$$

$$C_{34} = [1\ 0]$$

The advantage of using this decoupled 4 state observer is that it allows us to control the poles of the controller and the observer independently.

Proof that the observer states converge to the real states over time

When we differentiate the equation for the error $e = x - \hat{x}$, we end up with:

$$\dot{e} = \dot{x} - \dot{\hat{x}}$$

$$\dot{x} = Ax + Bu \ \ and\ \dot{\hat{x}} = (A - LC)\hat{x} + LY$$

$$\dot{e} = \ Ax + Bu\ - \left((A - LC)\hat{x} + LY\ \right)$$

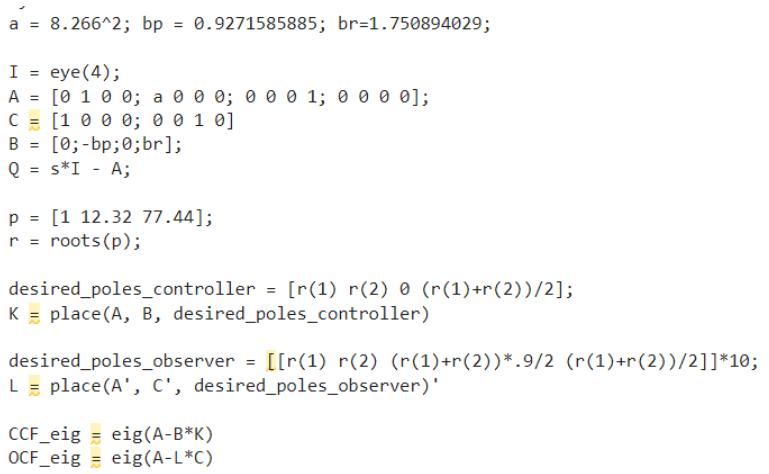
$$\dot{e} = (A - LC)x - (A - LC)\hat{x}$$

$$\dot{e} = (A - LC)e$$

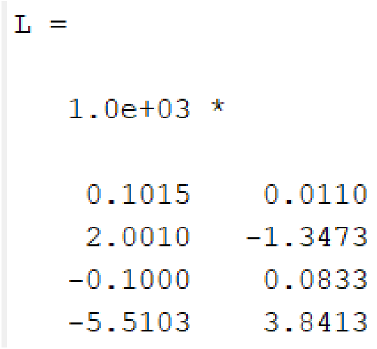
To show that e goes to zero over time, the eigen values of (A-LC) must lie in the LHP. We chose our desire poles for the observer to be lie 10 times further than the controller's. Using:

$$A = \begin{bmatrix} 0\ 1\ 0\ 0 \\ a\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0 \end{bmatrix}\ and\ C = \begin{bmatrix} 1\ 0\ 0\ 0 \\ 0\ 0\ 1\ 0 \end{bmatrix}$$

$$in\ OCF, we\ use\ A'and\ C'in\ the\ Matlab\ command\ 'place'to\ get\ our\ L$$



```
a = 8.266^2; bp = 0.9271585885; br=1.750894029;

I = eye(4);
A = [0 1 0 0; a 0 0 0; 0 0 0 1; 0 0 0 0];
C = [1 0 0 0; 0 0 1 0]
B = [0;-bp;0;br];
Q = s*I - A;

p = [1 12.32 77.44];
r = roots(p);

desired_poles_controller = [r(1) r(2) 0 (r(1)+r(2))/2];
K = place(A, B, desired_poles_controller)

desired_poles_observer = [[r(1) r(2) (r(1)+r(2))*.9/2 (r(1)+r(2))/2]]*10;
L = place(A', C', desired_poles_observer)'

CCF_eig = eig(A-B*K)
OCF_eig = eig(A-L*C)
```



```
L =

   1.0e+03 *

    0.1015    0.0110
    2.0010   -1.3473
   -0.1000    0.0833
   -5.5103    3.8413
```

When we compute our eigen values for (A-LC) we indeed see that they lie 10 times further than the CCF's. Since (A-LC) is stable, if we set $\dot{e} = 0$ in our equation

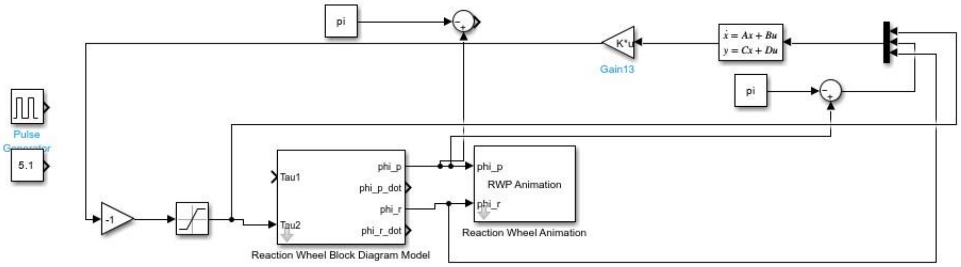
$$\dot{e} = (A - LC)e = 0$$

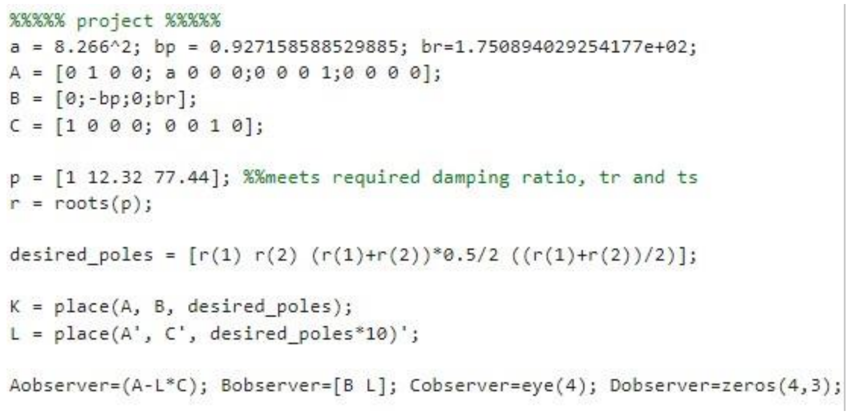
the only e that satisfies this equation is:

$$e = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Table of Values

We were not able to realize meaningful values in this part because our system exhibited unexpected behavior. We derived our observer matrices according to the following MATLAB equations and built a Simulink model to take as input the values and simulate the disturbances behavior.





```
%%%%% project %%%%%
a = 8.266^2; bp = 0.927158588529885; br=1.750894029254177e+02;
A = [0 1 0 0; a 0 0 0;0 0 0 1;0 0 0 0];
B = [0;-bp;0;br];
C = [1 0 0 0; 0 0 1 0];

p = [1 12.32 77.44]; %%meets required damping ratio, tr and ts
r = roots(p);

desired_poles = [r(1) r(2) (r(1)+r(2))*0.5/2 ((r(1)+r(2))/2)];

K = place(A, B, desired_poles);
L = place(A', C', desired_poles*10)';

Aobserver=(A-L*C); Bobserver=[B L]; Cobserver=eye(4); Dobserver=zeros(4,3);
```

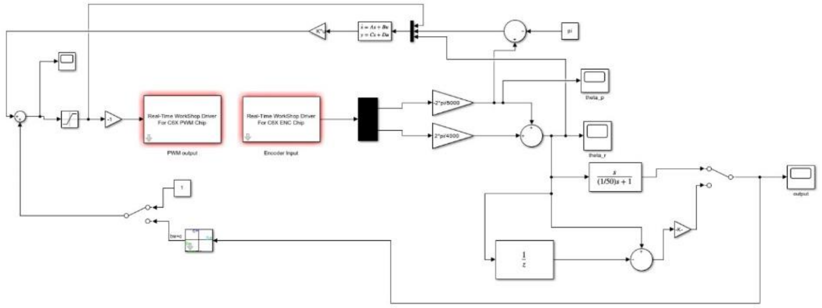
For the maximum IC deviation, pulse disturbance, and constant perturbation cases our simulated RWP kept spinning and did not settle at the equilibrium. We suspect that was the case because some matrices realized from the desired second order poles p might have not lacked independence leading to a singularity problem.

	Two state feedback		Three state feedback		Observer	
Max IC deviations	$\delta\theta_p$	$\delta\dot{\theta}_p$	$\delta\theta_p$	$\delta\dot{\theta}_p$	$\delta\theta_p$	$\delta\dot{\theta}_p$
	0.13	1.2	0.12	1.1		
Max pulse	8		8.8			
Max disturbance	5.1		7.7			

We expected to see the animated RWP being very robust to disturbances and quickly getting to equilibrium state because of the fast poles from the observer.

Behavior of System

Because of the above issue, upon running our real system, no physical occurrence took place. In this section we expected to similarly see the RWP arm stabilize the inverted equilibrium position and thwart any disturbances rapidly to continue staying at equilibrium. This is our block diagram that unfortunately did not yield some results for us after repeated attempts.



Conclusions

In this lab we were able to put into practice all the knowledge that we have acquired over the semester from both the class and the labs. We were able to model a system through governing non-linear equations of motion, linearize the equations and convert them into state space. We were also able to characterize the system by performing friction and system identifications which we equipped us with the values to use in designing in our controls for the system. Finally, we implemented different controllers, primarily relying to state space techniques to see the tradeoffs that came with each controller. Comparing the observer controller and the three state and two controllers, it is clear that the full state observer is very sensitive to disturbance quickly thwarts them better than the other controllers. In our three state controller we saw that by controlling the rotor velocity we were able to achieve a robust inverted pendulum controller.

Even though we were caught by time and were not able to leverage the lab TA's help we are glad that we dived deep into our notes and online resources to better understand CCF and OCF and how to get to a decoupled system. Learning and practicing controller design using state space is has unlocked our ability to interface with the more recent and widely adopted controls technology being used today.