ECE 486: Final Project Report

Tianhao Ji, Rawnie Singh
TA : Yuju Chang

Spring 2024

Introduction to the Reaction Wheel Pendulum

The Reaction Wheel Pendulum (RWP), as depicted in Figure 1, incorporates a simple pendulum mechanism enhanced by a rotating wheel at its extremity, driven by a 24-Volt DC motor The inputs to our controller are derived from two relative encoders that measure the position of both the pendulum and the rotor. The output from the controller is utilized to operate a motor, which is the sole actuator in the Rotary Wing Pendulum (RWP) system. This detail is crucial, as our controller was required to monitor and adjust four variables within the system using only the actions of this single available actuator.

The equilibrium position was set to be the upright and downward positions, which are 0 and π. The program used for much of the design was Matlab Simulink.

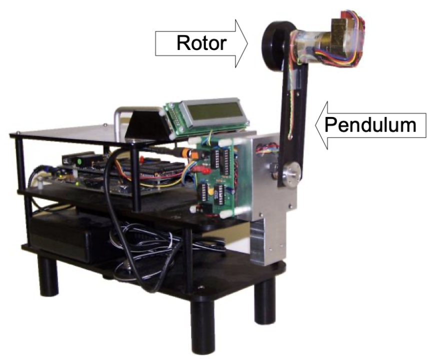


Figure 1: The Reaction Wheel Pendulum

Mathematical Model

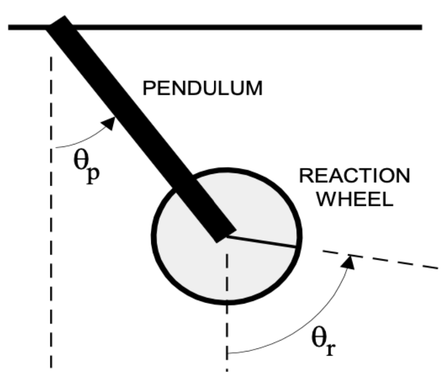


Figure 2: Schematic Diagram

The mathematical model of the Reaction Wheel Pendulum (RWP) is derived using the Lagrangian method, which considers kinetic and potential energies to formulate the equations of motion. The system is defined by the generalized coordinates θ_p and θ_r for the pendulum and the reaction wheel, respectively.

The kinetic energy K and potential energy V are expressed in terms of these coordinates and their derivatives, leading to the Lagrangian $L = K - V$. From the Lagrangian, the equations of motion are derived using:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} = \tau_k,$$

where τ_k represents the generalized forces and q_k the generalized coordinates. Key parameters include the masses of the pendulum and the rotor (m_p and m_r), their respective distances from the pivot (ℓ_p and ℓ_r), and the inertia components. These form the basis for analyzing the system's dynamic behavior under rotational motion.

$$\begin{aligned}
KE_{\text{pendulum+rotor}} &= \frac{1}{2}J\omega^2, \\
PE_{\text{pendulum+rotor}} &= mg\ell(1-\cos\theta_p), \\
KE_{\text{rotor}} &= \frac{1}{2}J_r\omega^2, \\
PE_{\text{rotor}} &= 0, \\
L_{\text{pendulum+rotor}} &= KE_{\text{pr}} - PE_{\text{pr}} = \frac{1}{2}{J_w}^2 - mgl\left(1-\cos\theta_p\right), \\
L_{\text{rotor}} &= KE_{\text{rotor}} - PE_{\text{rotor}} = \frac{1}{2}Jrw^2, \\
\text{Lagrange Equation}_{\text{pendulum+rotor}} &= \frac{d}{dt}(\dot{\theta}_p) + mg\ell\sin\theta_p = -\frac{k}{J}, \\
\text{Lagrange Equation}_{\text{rotor}} &= \dot{\theta}_r = \frac{k}{J}.
\end{aligned}$$

Linearized Mathematical Model

Considering a small deviation from the equilibrium position upright $\theta = \pi$, let $\theta_p = \pi + \delta\theta_p$ where $\delta\theta_p$ is small. The trigonometric functions are approximated as follows:

$$\cos(\theta_p) \approx -1, \quad \sin(\theta_p) \approx \delta\theta_p$$

we only focus on the first two term of taylor expanison.

Substitute these into the Lagrangian to get the linearized equations. The state-space form of the system can be represented as:

$$\dot{x} = Ax + Bu, \quad y = Cx + Du$$

where

$$x = \begin{bmatrix} \delta\theta_p \\ \dot{\theta}_p \end{bmatrix}$$

and matrices A, B are defined based on system parameters.

The state-space representation of the system is derived by:

$$\dot{x} = Ax + Bu$$

where

$$x = \begin{bmatrix} \delta\theta_p \\ \dot{\theta}_p \\ \delta\theta_r \\ \dot{\theta}_r \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ -b_p \\ 0 \\ b_r \end{bmatrix},$$

$$u = -Kx$$

$$K = k_1 \quad k_2 \quad k_3 \quad k_4$$

after calculation,

1x4 double

	1	2	3	4
1	-218.8401	-19.2190	0	-0.0308

$$a = \omega_r^2 \approx 8.3^2 = 68.89$$

In order to get $b_p and b_r$, we need to do a poly fit for the data we measure the pendulum rotor is under different situation

$$V = b_p * Mean + b_r$$

V= -100 , Mean = -2.010
V= -150, Mean = -2.492
V= -50, Mean = -1.261
V= -75 , Mean = -1.557
V= -250, Mean = -3.999

V= 100 , Mean = 1.836
V= 150, Mean = 2.45
V= 50, Mean = 1.241
V= 75 , Mean = 1.532
V= 250, Mean = 3.959

$$b_p \approx -0.87 b_r \approx 163$$

Full State Feedback Control with Friction Compensation

Development of the PD control with friction compensation: Next, PD control of the Pendelum was implemented using friction compensation and the closed loop identification method, with the goal of velocity control. We modeled the system's friction as a linear function and used a best fit model on data collected, as explained below.

Derivation of friction compensator values: Using equations 15 and 16 of the lab manual:

$$\dot{\omega}_r = \ddot{\theta}_r = b_r(u + F(\dot{\theta}_r)) = 0$$

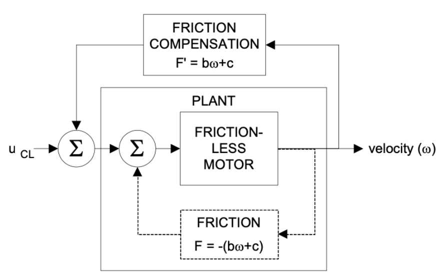
$$u = -F(\dot{\theta}_r) + u_{cl}$$

we ran the motor at multiple speeds to create datapoints while keeping the pendelum rotation locked for a constant position. The speeds represented a set velocity, and this allowed us to simplify our relationship to

$$a = \omega_r^2 \approx 8.3^2 = 68.89$$

which uses the non-zero control effort to give us a friction value for that specific initial speed.

This yielded viscous and coulomb values of 0.008 and -0.0016. Friction compensation was then implemented through a switch block, as follows:



OPEN-LOOP SYSTEM WITH FRICTION COMPENSATION

Equilibrium State and Stability Analysis

Show that the only place where $\dot{x} = 0$ is at your linearized equilibrium position. Prove that for the linearized system the equilibrium state $x = [0, \text{whatever}, 0]$ is stable.

The initial conidtion was set to $\theta = \pi$, which means that the system is stabilized around that equilibrium. Using the matrix from the lab manual, A-BK can be manipulated into the following system of equations:

$$\ddot{\theta}_p + a\sin(x) + \cos(x)(\theta - x) = -b_p u$$

$$\ddot{\theta}_p = 0 + \delta\theta_p - b_p u$$

$$\begin{array}{ccccccc} \dot{\mathbf{x}} & = & \mathbf{A} & \mathbf{x} & + & \mathbf{B}\,u & \\ \begin{bmatrix} \delta\dot{\theta}_p \\ \ddot{\theta}_p \\ \delta\dot{\theta}_r \\ \ddot{\theta}_r \end{bmatrix} & = & \begin{bmatrix} 0 & 1 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} \delta\theta_p \\ \dot{\theta}_p \\ \delta\theta_r \\ \dot{\theta}_r \end{bmatrix} & + & \begin{bmatrix} 0 \\ -b_p \\ 0 \\ b_r \end{bmatrix} u & \end{array} \tag{20}$$

The previously determined constants in the matrix are a = 68.89, bp = -0.87, and br = 163. Now that we know the A matrix, we can check the eigenvalues of A-BK to prove stability about the equilibirum - where B and K are all defined. The eigenvalues as per a Matlab script computing eig(A-BK) were all in the LHP, proving stability and full rank.

Since the output matrix is stable, the only values that would make $\delta\theta p$ and $\delta\theta r$ valid would be π for$\delta\theta p$and 0 for$\delta\theta r$, proving stability at equilibrium

Simulations and Implementation

Maximum IC deviation, pulse disturbance, and constant perturbation: Next, various simulations were conducted on the RWP simulink block diagram to test the maximum possible deviations and disturbances that the controller can take while still able to find a path to stability. This was simply done through an iterated series of modifying values and observing affects on the system through the animation block.

The Windows Target simulations provided the following values that met the goal of finding stability at equilibrium, meaning these are the thresholds for the reliable functionality of the system.

Table 1: Robustness Comparisons

	Two-State Feedback (4.2)		Three-State Feedback (4.3)		Observer (5.1)	
① Max IC deviations	0.4 $\delta\theta_p$	0.9 $\delta\dot{\theta}_p$	0.89	0.8	0.6	0.7
② Max pulse	6		7		5	
③ Max disturbance	4		6		4	

We can see that for the most part, the controller was able to stabilize slight disturbances. However, the three state controller behaved much more conservatively, as the addition of an external state took an expected toll on the system.

Full State Feedback Control with Decoupled Observer

Discuss why are observers used? Can we decouple the 4-state observer design into two 2-state observers? Observers are used to create a more accurate estimate of the system's position and velocity. We attempt to design an observer based on desired poles to see the drawbacks and benefits. Both become clear through the experiment, and the drawbacks of limited control stand out.

There are 2 types of observers - 2 state and 4 state. The 4 state observer design can be decoupled into the 2 state design using the block diagonal form. This adds the benefit of keeping the two 2 state observers independent from one another, boosting control stability.

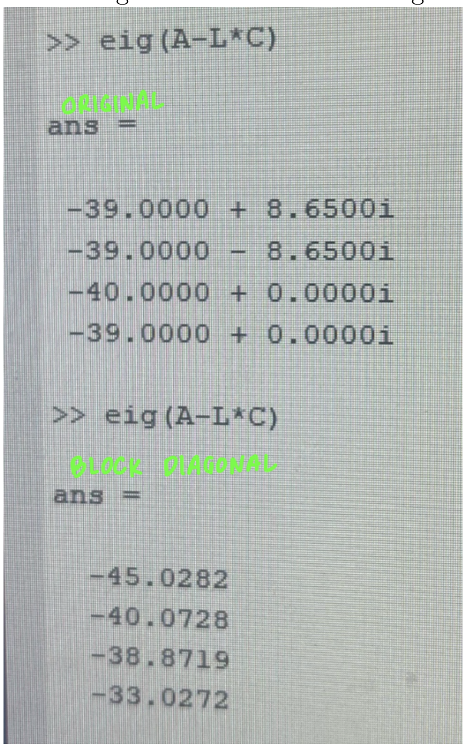
By seperating our $\dot{x}$ into $\dot{x}_{12}$ and $\dot{x}_{34}$ we can then converge our matrixes into our A-LC form, giving us the following L matrix.

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \\ l_{31} & l_{32} \\ l_{41} & l_{42} \end{bmatrix}, \quad (\mathbf{A} - \mathbf{LC}) = \begin{bmatrix} -l_{11} & 1 & -l_{12} & 0 \\ a - l_{21} & 0 & -l_{22} & 0 \\ -l_{31} & 0 & -l_{32} & 1 \\ -l_{41} & 0 & -l_{42} & 0 \end{bmatrix}$$

All of these computations were done using a Matlab script:

```
% keep K < 300

%calculate a
wnp_meas = 8.8;
a = wnp_meas*wnp_meas;

% A matrix
A = [ 0 1 0 0 ;
      a 0 0 0 ;
      0 0 0 1 ;
      0 0 0 0];

% B matrix
B = [ 0; -0.87; 0; 163];
C = [1 0 0 0 ; 0 0 1 0];
At = transpose(A);
Ct = transpose(C);

% poles
% wnp = 9, zeta = 0.5 --> use 2nd order sys denominator to solve for poles
p = [-3.9+8.65i, -3.9-8.65i, -3.9, 0] ;
p_observed = [-39+8.65i, -39-8.65i, -39, -40] ;

% compute K
rank(ctrb(A,B));
K = place (A,B, p);
L1 = place (At,Ct, p_observed);
L = transpose(L1)

% check stability
X_new = (A-B*K);
eig(X_new);
```

both being stable with the following results:



```
>> eig(A-L*C)

ORIGINAL
ans =

 -39.0000 + 8.6500i
 -39.0000 - 8.6500i
 -40.0000 + 0.0000i
 -39.0000 + 0.0000i

>> eig(A-L*C)

BLOCK DIAGONAL
ans =

  -45.0282
  -40.0728
  -38.8719
  -33.0272
```

Stability of the Observer

To prove that the observer states converge to the real states over time, consider the error between the actual states x and the estimated states $\hat{x}$, defined as $e = x - \hat{x}$. The dynamics of the error can be described by the differential equation:

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

Given the observer equation $\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$, where L is the observer gain and C is the output matrix, substituting $\dot{\hat{x}}$ into the error dynamics gives:

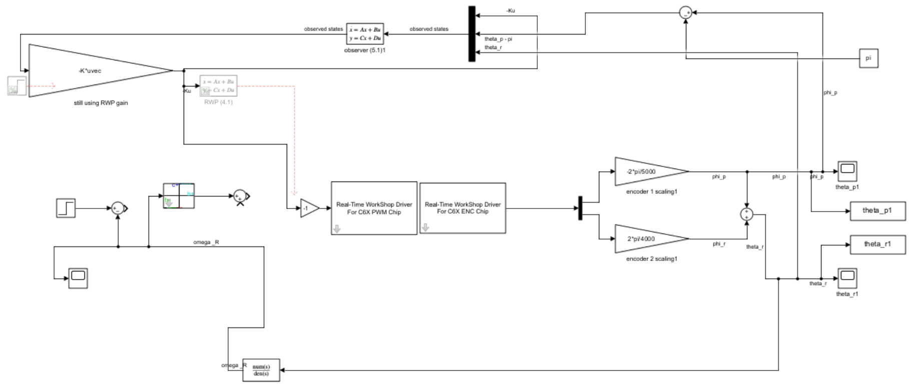
$$\dot{e} = Ax - A\hat{x} - L(Cx - C\hat{x})$$

Simplifying, we find:

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

To analyze the stability of the observer, calculate the eigenvalues of the matrix $(A - LC)$. The placement of the observer poles through the matrix L ensures that the eigenvalues are placed in the left half of the complex plane, indicating stability.

Thus, if $(A - LC)$ is stable, the error dynamics governed by $\dot{e} = (A - LC)e$ imply that $e \to 0$ as $t \to \infty$. This shows that the observer states converge to the real states over time, confirming the stability of the observer.



When tested with the reaction wheel pendulum, the observer model proved highly effective in maintaining the pendulum arm's position at the inverted equilibrium. Unlike the transfer function used in the three-state controller, the observer provided greater accuracy and did not induce any unintended motions into the system. Although our simulations indicated that the controller struggled with consistent external disturbances or deviations in initial conditions, these issues were not observed in the final implementation.

Compared to the three-state controller, the observer model demonstrated enhanced robustness to pulse disturbances—a feature that was retained in the final model. Due to its effectiveness, we opted to use this controller for our swing-up mechanism, as it effectively replicated the motion caused by pulse disturbances pushing the arm. The observer model underscored the importance of having accurate state estimates in the system and demonstrated that it is feasible to achieve reasonably accurate estimates even when a state is not directly measured.

Conclusion

simulation

For the simulation parameters, as we can see that three-state feedback has higher max pulse and max disturbance, which would result in it has best performance to counter pulse and disturbance, Max IC deviation would show the range of stable region that controller would be able to achieve

Reality

The Observer was less effective to maintain the equilibrium position of pendulum, when we apply disturbance to both controller, the three state feedback is more robust, For some reason, it takes more time for observer to find the equilibrium point, because I have to manually rotate the rotor to the upright and wait for observer response.

Over all, The three state feedback controller is working better. since the pendulum system is well-defined and simple to construct.

Extra credit part (6)

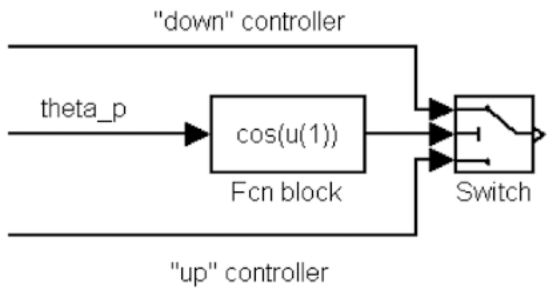


Figure 9: Implementing Switching Control

The concept of switching control is applied to the RWP to manage its dual equilibrium states: upright and downwards. The controller switches between two modes based on the pendulum's angle, θ_p, to stabilize the pendulum in both directions:

- **Down Controller:** Activates when θ_p is between 0 to $\frac{\pi}{2}$ and $\frac{3\pi}{2}$ to 2π. It aims to stabilize the pendulum swinging downwards.
- **Up Controller:** Engages when θ_p is between $\frac{\pi}{2}$ to $\frac{3\pi}{2}$. It stabilizes the pendulum in the upright position.

The switch between controllers is based on a threshold angle to prevent instability during the transition phase. This strategy ensures that the appropriate controller is active, considering the pendulum's current state and motion dynamics.

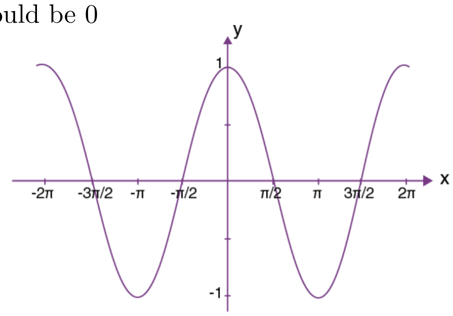
In order to construct down controller, we did same procedure as up controller, but during linearization of math model, we need to linearize around θ_p = 0, and the only variation is A matrix, thus we have new K for the down controller

1x4 double

MATLAB Online

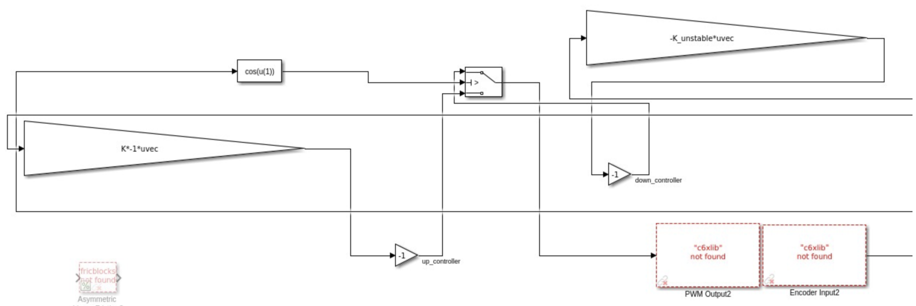
		2	3	4
1	-58.0622	-7.6776	0	0.0308

we use cos(u(1)) as function for the switch, thus the threshold for the switch would be 0



as we can tell from the cos graph, the range of $\frac{\pi}{2}$ to $\frac{3\pi}{2}$ will be -1, which is the value after linearized state space model around upright position $= \pi$

vice versa, 0 to $\frac{\pi}{2}$ and $\frac{3\pi}{2}$ to 2π is for downwards after linearized state space model



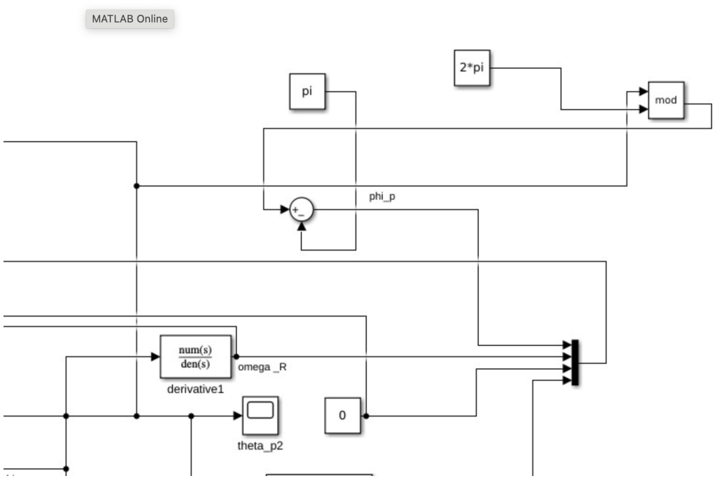
Extra credit part (7)

in order to achieve the feature of swinging up control, one of the simple approach will be make down side controller unstable, so I simply make the poles on RHP before computing place command in matlab

unstable K for controller

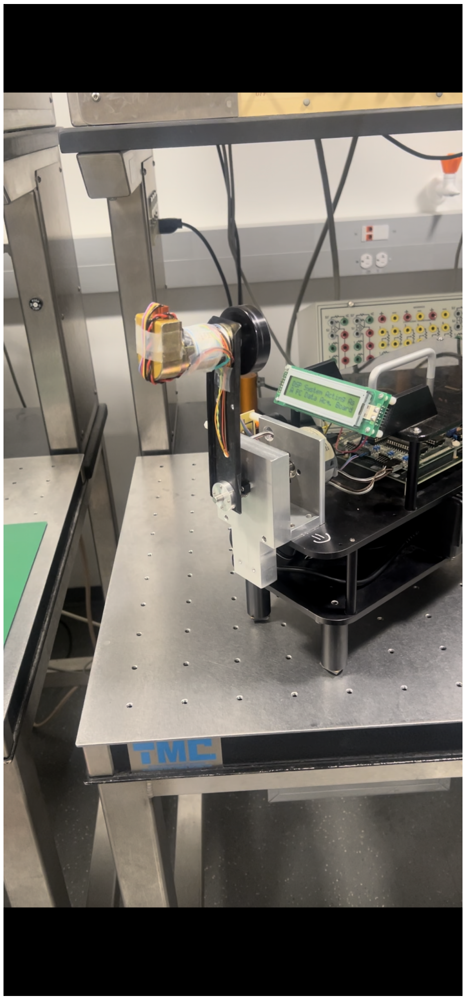
	1	2	3	4
1	-91.8208	9.0386	0	-0.0438

also, we need to consider the position of swinging rotor maybe towards right, because we assume swing would be clockwise if right swing would be the case, the θ_p would be negative, thus we find out the solution will be add a mod block before the input of multipluxer which make the θ_p be in range of 0 to 2π



after several tuning and testing, we find out the best ζ and ω_n for the up controller

The pendulum swing to the uprigtht and hold still.


TMC