ECE 147B Control Systems

Lab 3 Report:

Balancing the Inverted Pendulum

Student : Corey Gravelle

Perm Number: 8094211

E-mail: cgravelle@umail.ucsb.edu

3/6/2010

Department of Electrical and Computer Engineering, UCSB. ECE158

Abstract:

In this lab, I will balance a rod vertically, by controlling a motorized cart beneath it using feedback loops. This will be accomplished by designing discrete-time state feedback controller and state-feedback estimator, with input measurements of the current angle of the rod and position of the cart. In part 1 of the lab, I will design a discrete-time state-feedback controller, as well as a simple estimator using a differentiator. Full-order and reduced-order state estimators will be designed and compared in parts 2 and 3. In addition to simply balancing the rod upright, I will experiment with more challenging problems such as recovering from unexpected offsets, and traversing the cart a desired distance horizontally.

Introduction:

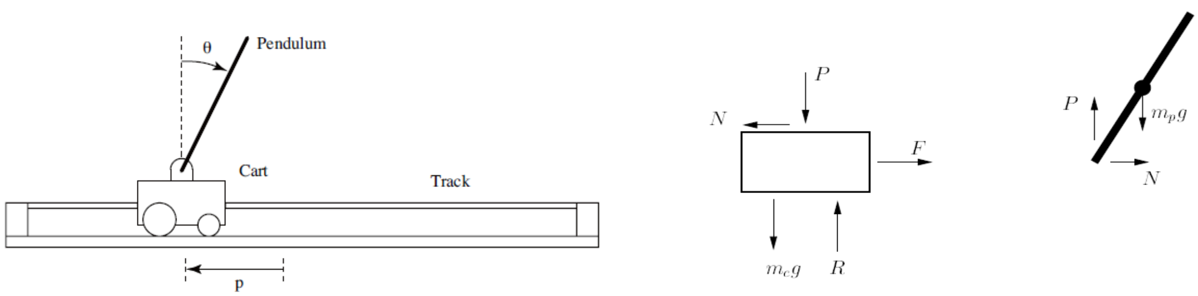


Figure 1: Model of the inverted pendulum system and free-body diagram of cart and pendulum

Using Newton's laws of mechanics with the free-body diagrams of Figure 2 and balancing the moments about the pendulum's center of mass, one may arrive to the following equations:

$$m_c \ddot{p} = F - N$$

(1)

$$N \;=\; m_p(\ddot{p}+l\cos(\theta)\ddot{\theta}-l\sin(\theta)(\dot{\theta})^2)$$

(2)

$$P \;=\; m_p g+m_p l(-\sin(\theta)\ddot{\theta}-\cos(\theta)(\dot{\theta})^2).$$

(3)

$$I\ddot{\theta}=Pl\sin(\theta)-Nl\cos(\theta)$$

(4)

Combining the above equations, one may find:

$$F=(m_c+m_p)\,p\ddot{\theta}+m_p l\cos(\theta)\ddot{\theta}-m_p l\sin(\theta)(\dot{\theta})^2.$$

(5)

$$\ddot{\theta}(I+m_p l^2)=m_p gl\sin(\theta)-m_p l\,\ddot{p}\cos(\theta)$$

(6)

From previous linerization models, one may recall the relationship between input voltage and force on the cart:

$$F \;=\; \frac{K_m K_g}{Rr}V-\frac{K_m^2 K_g^2}{Rr^2}\dot{p}$$

(7)

The above equations may be combined the following encompassing equations for acceleration:

$$\ddot{p} = \frac{\frac{K_m K_g}{Rr} V - \frac{K_m^2 K_g^2}{Rr^2} \dot{p} - \frac{m_p l g}{L} \cos(\theta) \sin(\theta) + m_p l sin(\theta)(\dot{\theta})^2}{\left(M - \frac{m_p l cos^2(\theta)}{L} \right)}$$

(8)

$$\ddot{\theta} = \frac{g sin(\theta) - \frac{m_p l (\dot{\theta})^2}{M} \cos(\theta) \sin(\theta) - \frac{\cos(\theta)}{M} \left(\frac{K_m K_g}{Rr} V - \frac{K_m^2 K_g^2}{Rr^2} \dot{p} \right)}{\left(L - \frac{m_p l cos^2(\theta)}{M} \right)}$$

(9)

The state vector (10) may be used to create a fully linearized model from the following (11) and (12):

$$x(s) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} p \\ \dot{p} \\ \theta \\ \dot{\theta} \end{bmatrix} \qquad \dot{x}(s) = Ax + Bu \qquad y = Cx + Du$$

(10) (11) (12)

For the pendulum-up case, linearizing about the equilibrium $[0\ 0\ 0\ 0]^T$ gives the following state-space model:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -15.14 & -3.04 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 37.23 & 31.61 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 3.39 \\ 0 \\ -8.33 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

(13)

For the pendulum-up case, linearizing about the equilibrium $[0\ 0\ 180\ 0]^T$ gives the following state-space model:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -15.14 & -3.04 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -37.23 & -31.61 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 3.39 \\ 0 \\ 8.33 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

(14)

For <u>Week 1</u> of the experiment, I designed a state-feedback controller for the system using the state space of equation (13) and using a simple differencing scheme to estimate the position (15) and angular (16) velocities:

$$\dot{p}(k) = \frac{p(k) - p(k-1)}{Ts}$$

(15)

$$\dot{\theta}(k) = \frac{\theta(k) - \theta(k-1)}{Ts}$$

(16)

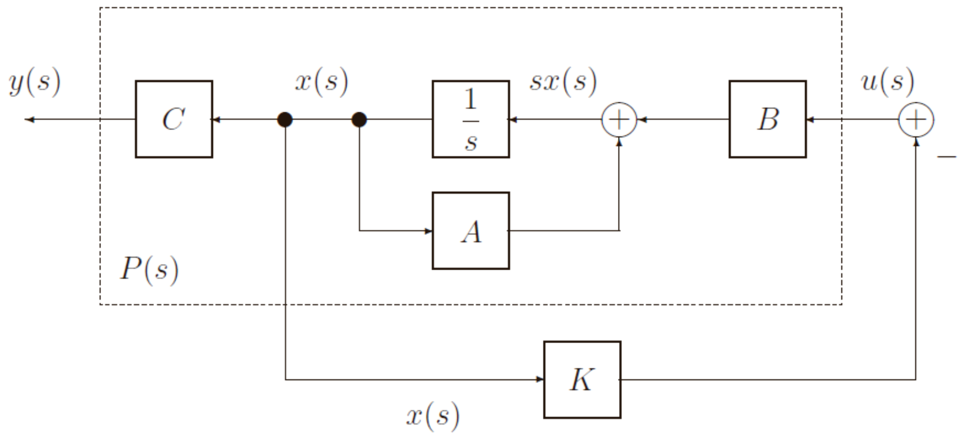


Figure 2: General Diagram of state-feedback controller as used in Week 1.

Using a sampling time of T_s = 0.001 seconds, I designed pendulum-up and pendulum-down controllers $K_{1\text{up}}$, $K_{1\text{dn}}$ with Matlab commands and trial and error techniques to place the poles at a fairly stable location (17). I used the place command to systematically find our first state-feedback controller matrices for both the pendulum-down case (18) and pendulum-up case (19):

$$\textit{poles of } K_1 = \begin{bmatrix} 0.996 & 0.997 & 0.893 & 0.894 \end{bmatrix} \tag{17}$$

$$K_{1\text{dn}} = \begin{bmatrix} 23.6400 & 10.6922 & 20.7348 & -2.8010 \end{bmatrix} \tag{18}$$

$$K_{1\text{up}} = \begin{bmatrix} -23.6399 & -19.6531 & -47.7477 & -9.5521 \end{bmatrix} \tag{19}$$

For Week 2, I designed a full-state estimator to reduce the noise of both the input position and angle measurements and the estimated linear and angular velocities.

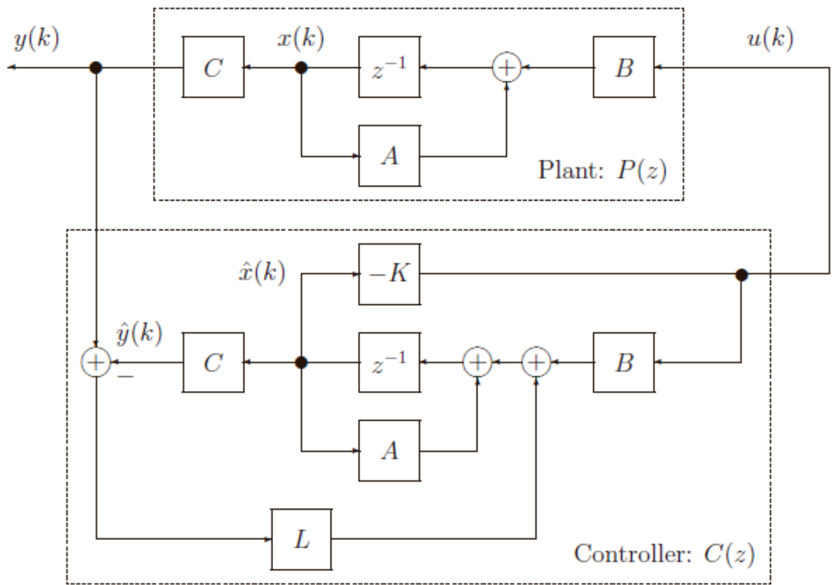


Figure 3: General diagram of a full state estimator, as used with Week 2

Using similar techniques as in Week 1, the controller gain matrix K

$$\textit{poles of } K_{1\text{dn}} \text{ and } K_{1\text{up}} = [0.996 \quad 0.997 \quad 0.893 \quad 0.894]$$

(20)

$$K_{1\text{dn}} = [1675.8 \quad 1003.0 \quad 853.7 \quad -384.2]$$

(21)

$$K_{1\text{up}} = [-1675.8 \quad -1013.5 \quad -2234.5 \quad -436.5]$$

(22)

The poles of the estimator gain matrix L were selected such that the estimator is much faster than the controller (averaging about ten times faster). L was found using similar pole-placing techniques (see Matlab code) which this time gives a two-column matrix from the estimation of linear and angular velocity:

$$\textit{poles of } L_{full} = \begin{bmatrix} 0.7404 & 0.7982 & 0.7905 & 0.7368 \end{bmatrix}$$

(23)

$$L_{full} = \begin{bmatrix} 0.4463 & -0.0003 \\ 46.0171 & -0.0667 \\ -0.0352 & 0.4727 \\ -15.2196 & 55.1204 \end{bmatrix}$$

(24)

For Week 3, I designed a reduced-order estimator to use the actual measured position and angle vectors as inputs to the plant and estimated only their velocities.

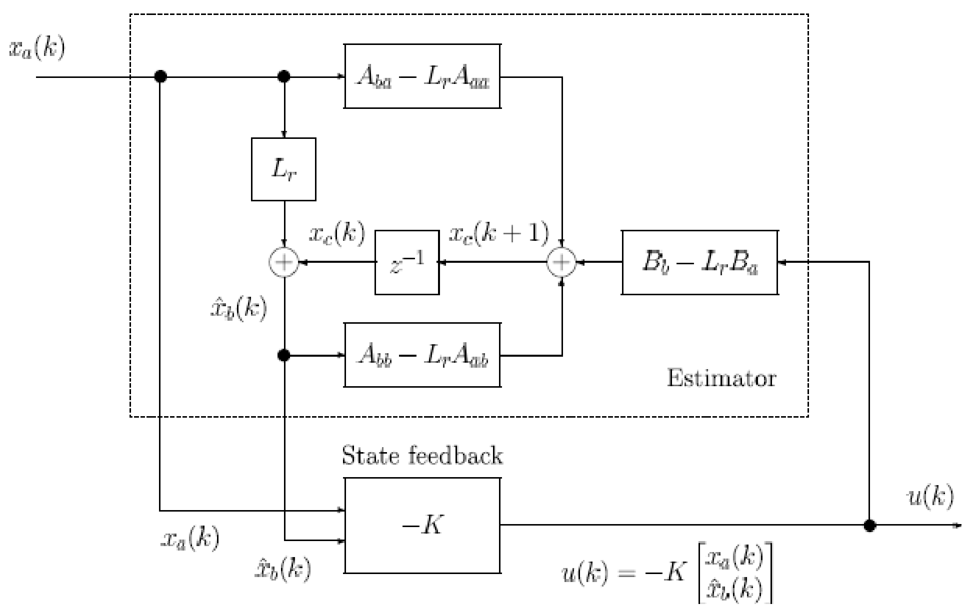


Figure 4: General diagram of a reduced-order state estimator, as used with Week 3

After rearranging the state-space matrices to splitting so the known measurement states x_a are split from the estimated states x_b, a new state-space is formed :

$$x_a = \begin{bmatrix} p \\ \theta \end{bmatrix} \qquad (25) \qquad\qquad x_b = \begin{bmatrix} \dot{p} \\ \dot{\theta} \end{bmatrix} \qquad (26)$$

$$A_t = T^{-1} A T \qquad B_t = T^{-1} B \qquad C_t = C T \qquad D_t = D$$

$$A_t = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & -3.04 & -15.14 & 0 \\ 0 & 31.61 & 37.23 & 0 \end{bmatrix} \quad B_t = \begin{bmatrix} 0 \\ 0 \\ 3.39 \\ -8.33 \end{bmatrix} \quad C_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad D_t = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

(27)

To interpret these new state elements with the model in Figure 4, A_t may be split into four 2x2 matrices and B_t into two 2x1 matrices. Thus, the following equation may be set up:

$$\begin{bmatrix} x_a(k+1) \\ x_b(k+1) \end{bmatrix} = \begin{bmatrix} A_{aa} & A_{ab} \\ A_{ba} & A_{bb} \end{bmatrix} \begin{bmatrix} x_a(k) \\ x_b(k) \end{bmatrix} + \begin{bmatrix} B_a \\ B_b \end{bmatrix} u(k)$$

(28)

The new estimator matrix L_r may be found to estimate only the velocities. I used similar techniques as before to find:

$$poles\ of\ L_r = \begin{bmatrix} 0.959 & 0.958 \end{bmatrix}$$

(29)

$$L_r = \begin{bmatrix} 26.1712 & -0.0015 \\ 36.4649 & 42.0156 \end{bmatrix}$$

(30)

Now, to idealize our new controller K_{lqr}, I used a slightly different technique: the *lqr* Matlab command (see code). This uses user-input parameters for maximum state error allowed (for each state) to design for the controller. Our maximum allowed errors for the states were as follows: position = 20cm, angle = 1/10 radians, velocity = 3.16m/s, and angular velocity = 1rad/sec.

$$K_{lqr} = \begin{bmatrix} -67.7964 & -177.1506 & -53.8102 & -31.5911 \end{bmatrix}$$

(31)

Results and Discussion

Week 1: Simple state-feedback controller without estimator

After linearizing the system for both pendulum-down and pendulum-up cases, the following state-space representations were found in the configuration of Figure 3:

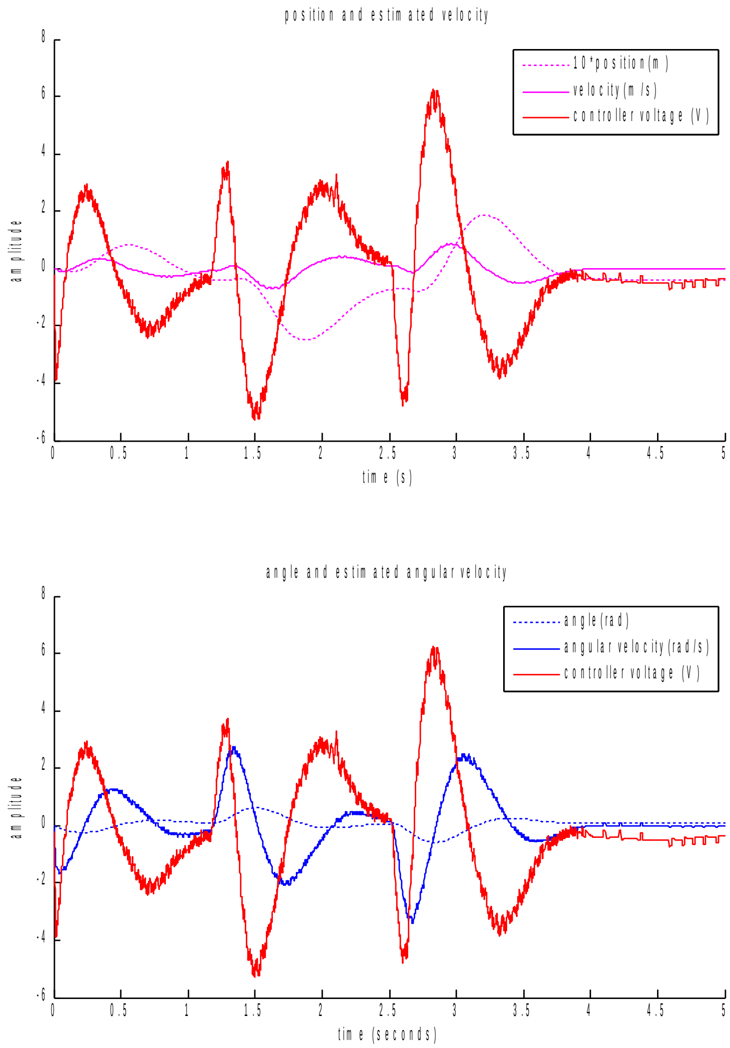


Figure 5: Position, Velocity, Angle, Angular Velocity, and Controller outputs for experimentally tested pendulum-down case with moving average filter implemented. Pendulum was tapped at 0, 1.2, and 2.5 seconds. Noise of the controller was largely a contribution from the noisy angular velocity measurements.

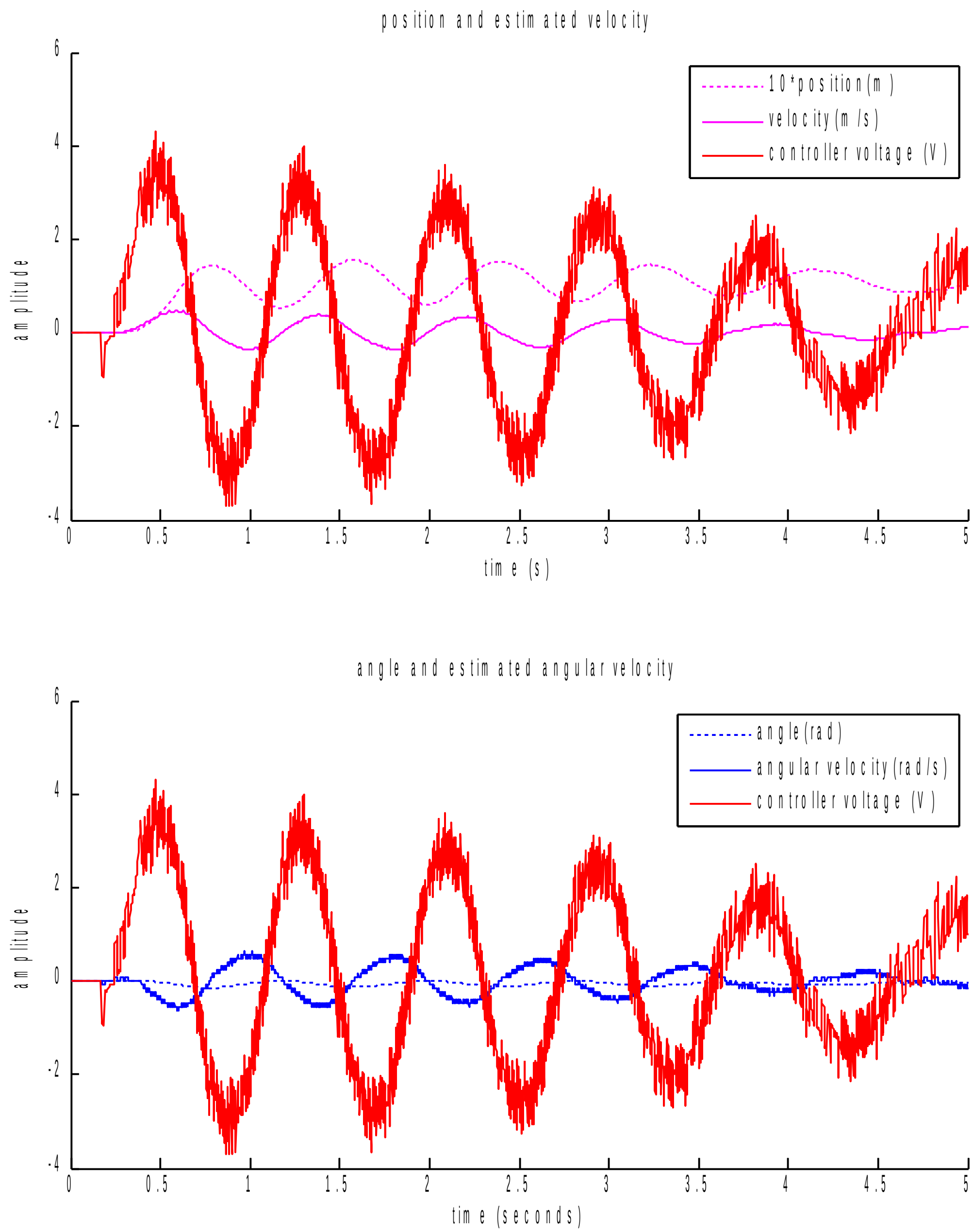


Figure 6: Position, Velocity, Angle, Angular Velocity, and Controller outputs for experimentally tested pendulum-up case. Like the pendulum-down case, the noise of the controller was largely a contribution from the noisy angular velocity measurements.

Week 2: State-feedback controller with estimator

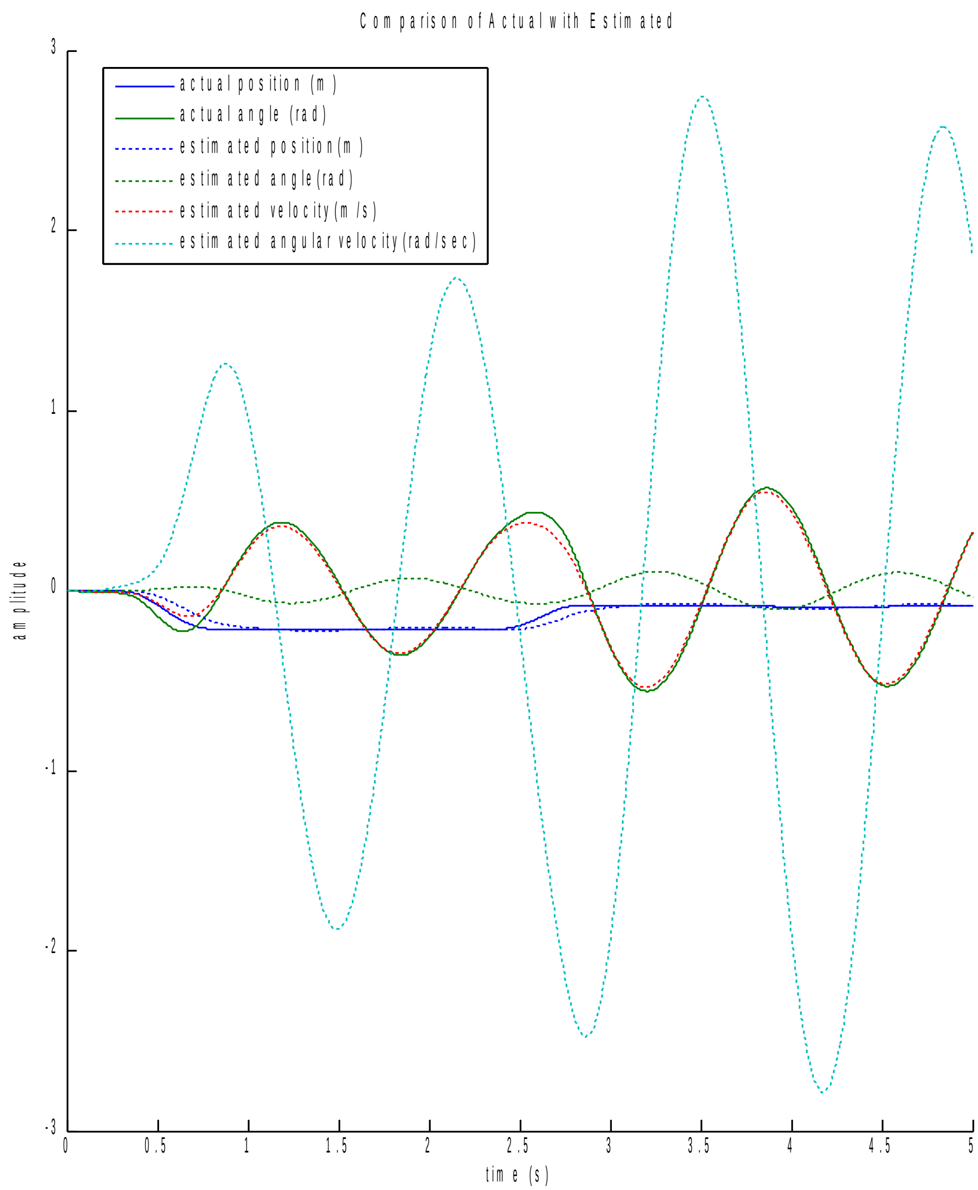


Figure 7: Estimator tested with pendulum down case and controller turned off. Testing commenced by tapping the cart one direction and then the other, as shown by the position graph. The controller was off so the pendulum was allowed to swing freely, a control for comparing with the response of our controller.

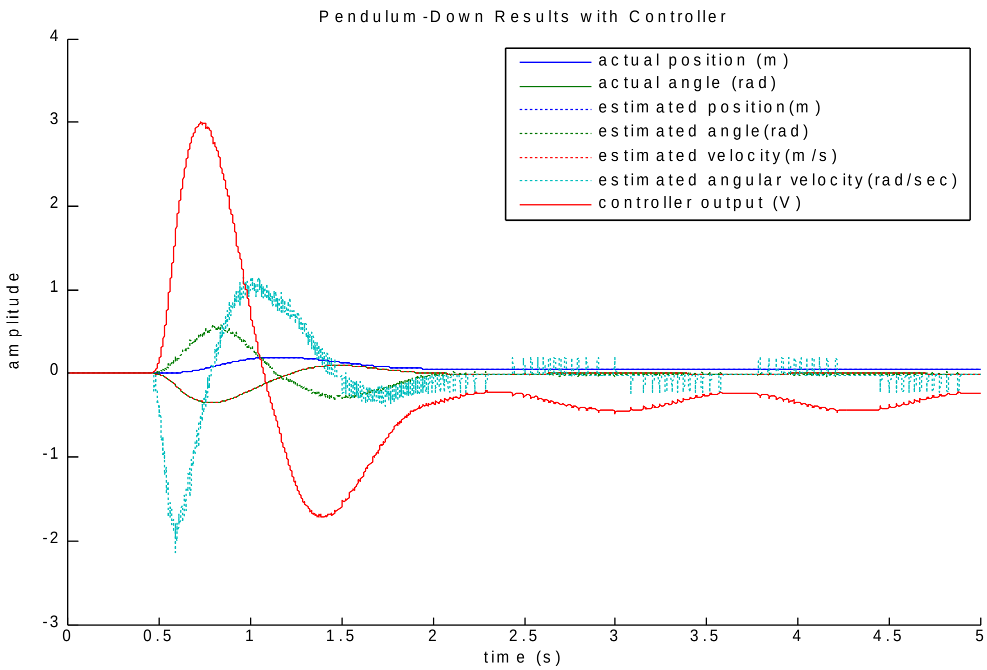


Figure 8: Response for pendulum-down controlled case with estimator (without moving average filter) in response to a tap at 0.5 seconds.

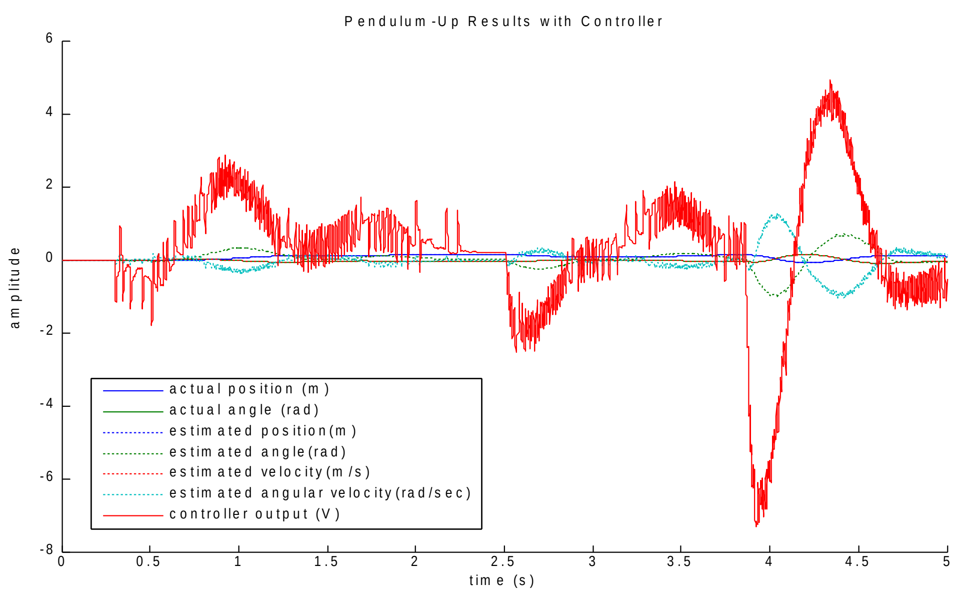


Figure 9: Pendulum-Up controlled case results with estimator (without moving average filter) in response to taps at 0.3, 2.5, and 3.7 seconds.

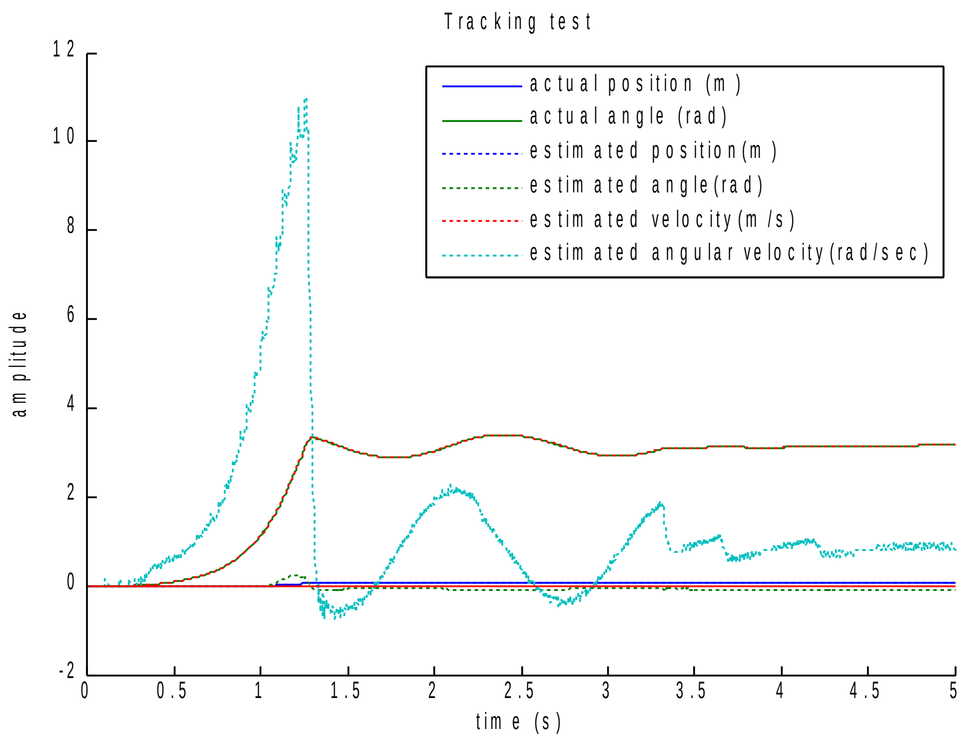


Figure 10: Results from a tracking test, with unexpected initial offset.

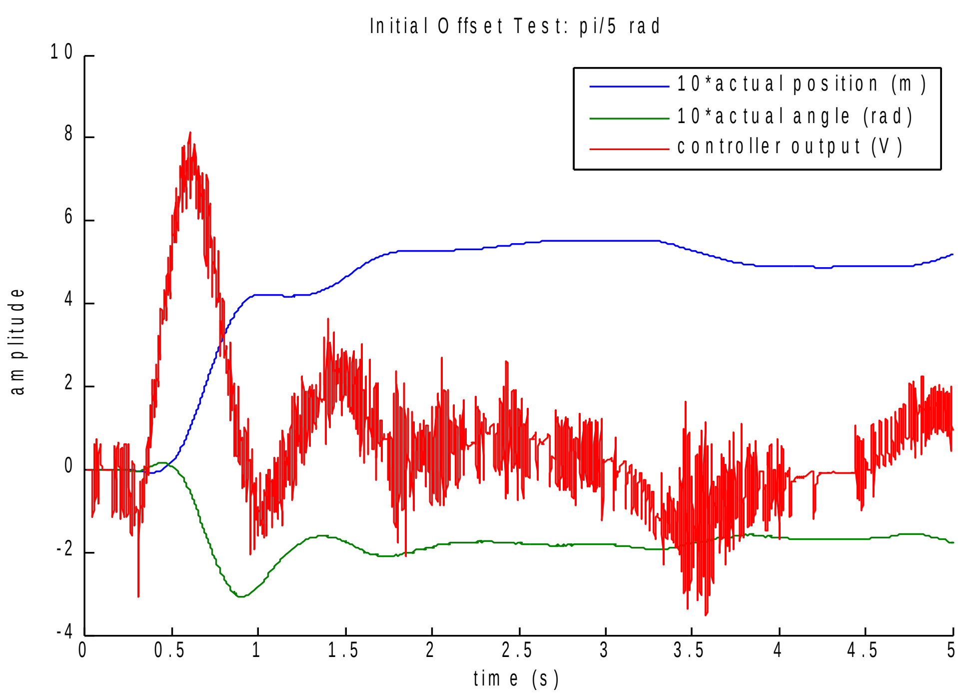


Figure 11: Testing of initial angle offset of pi/5 radians.

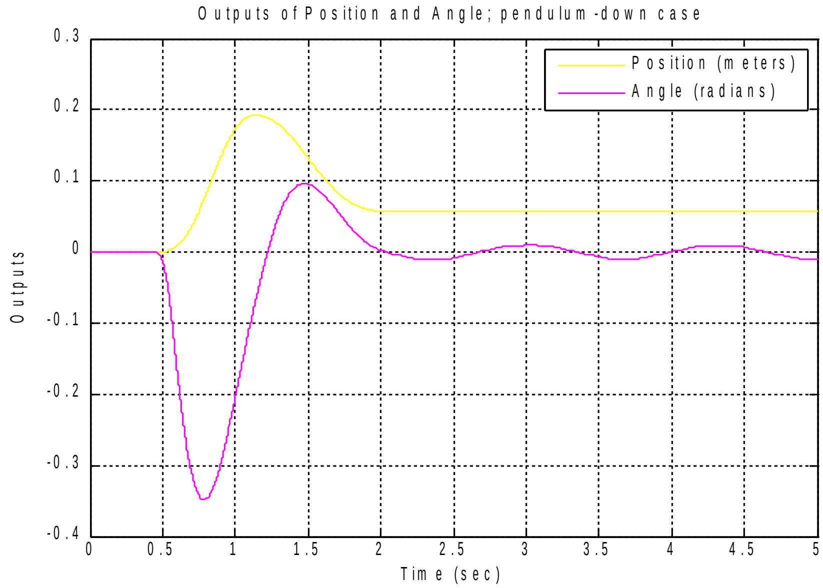


Figure 12: Stabilizing a position change in the pendulum-down case

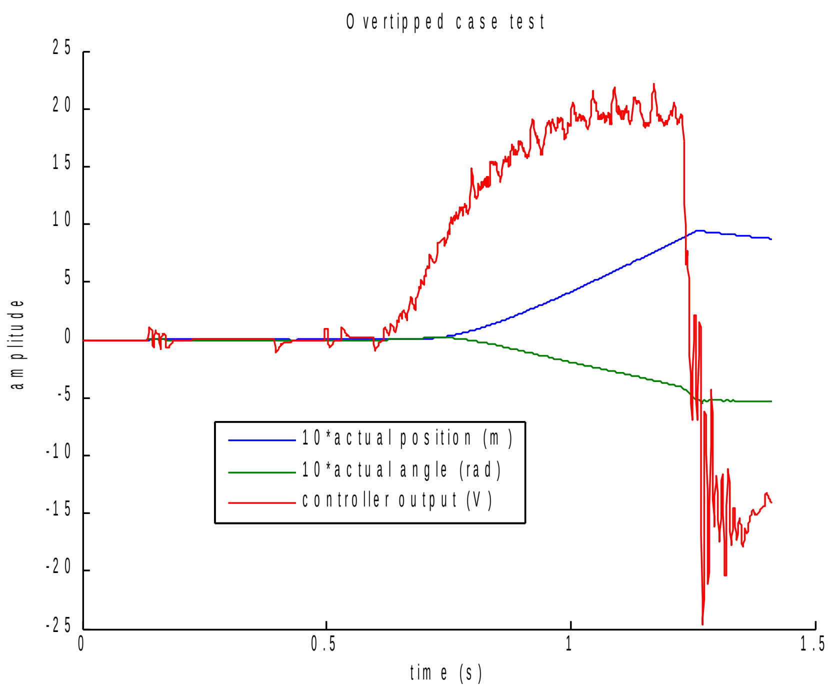


Figure 12: Testing of overtipped case. Controller voltage peaks at around 20V maximum which is not enough to stop the pendulum from falling. To note: the angle began with a ten degree or so offset, and the graph shows the deviation from that angle, NOT the deviation from zero.

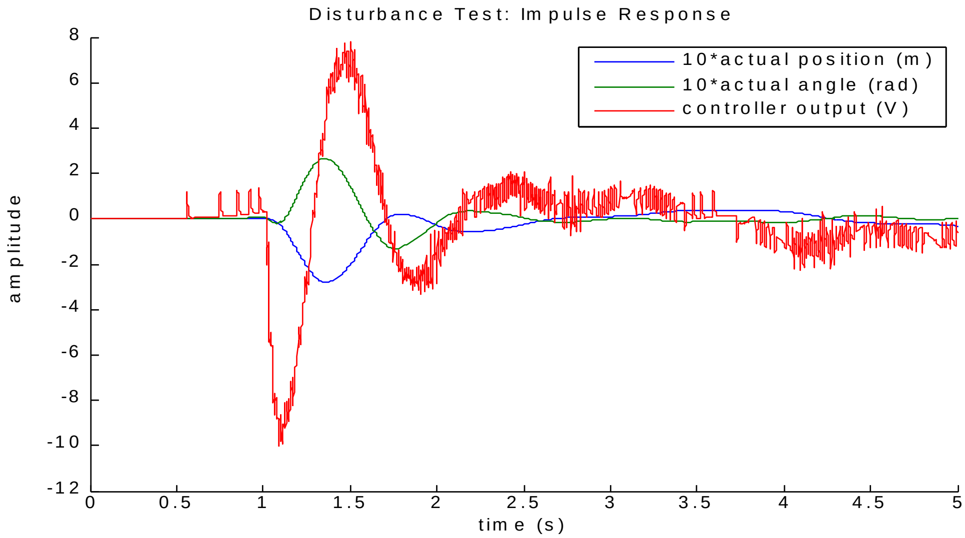
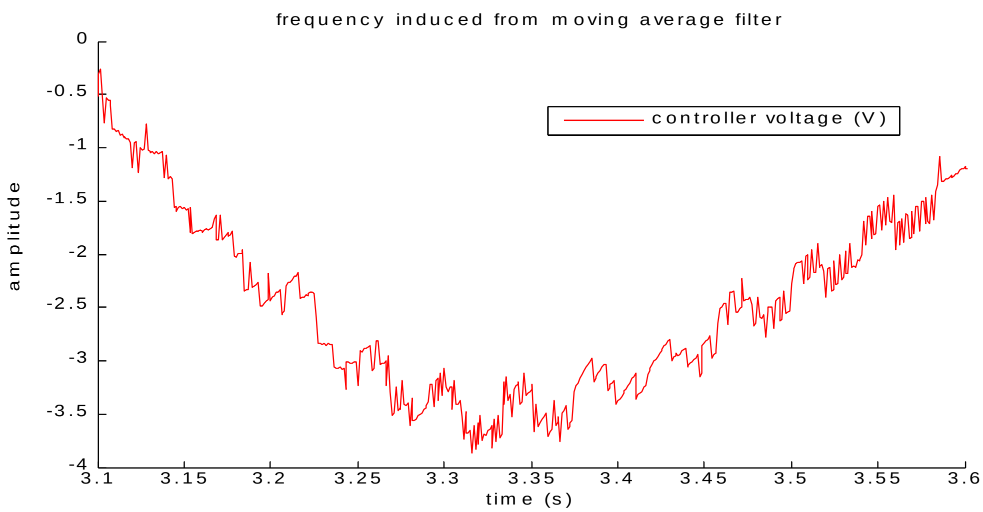


Figure 13: Testing of impulse disturbance response

In Weeks 1 and 2, the cart velocity and the pendulum angular velocity will be approximated by a simple differentiator, but for the reduced order estimator of Week 3, I simulated the experimentally measured position and pendulum angles and estimate the derivatives with a more accurate estimator. The state-feedback configuration of Figure 3 is the primary design for choosing controller gain matrices.

Discussion

In Week 1's experiments, there was a large amount of noise at the angular velocity which carried over into the controller voltage outputs. The noise originates from a set amount of radial ticks with which our system reads the angle of the pendulum. If the angle/distance conversion had a greater accuracy then there would be less noise. If this amount of noise is too great, it results in a chattering and a loss of efficiency, especially in the pendulum-up case. The noise in the system would cause the error between to the measured states and the estimated states to increase. As the error increases, the time it takes for the estimated states to catch up with the measured states would increase as well.



To decrease this noise, I placed a moving average filter after the velocity and angular velocity estimations to smooth out the response. The downside of implementing this filter is a phase shift; as one can see in Figure 5, there appears to be an oscillating motion along the estimated velocity plots. The frequency of this oscillation is dependent on the width of the moving average filter. Increasing its width would increase the frequency and decrease the amplitude of the oscillating waveform while smoothing out the estimated velocity. However, the greater one would increase the width of this filter, the greater the phase lag of the controller and the more difficult it is to control the system, resulting in a tradeoff between controller noise and speed.

The first-order system of Week 1 worked fine to stabilize the pendulum in the downward-swinging position. With the pendulum up case, however, there was an underdamped ringing response as the controller found it difficult to predict the inputs necessary to result in an equilibrium position. This may have been a result from controller poles that were too aggressive and caused the cart to overshoot its intended position target.

The full-order system takes longer to calculate the response, but is much more accurate and is the desired stabilizing control for our system. This system, especially with a moving-average filter implemented, takes that extra split-second to muffle the input measurement noise and result in a consistently accurate performance.