Inverted Pendulum Controllers

Garrett Wilson and Nathan Zimmerly

ENGR 352

June 9, 2016

Contents

1 PID Controller

1.1 Transfer Function of θ / F Without Friction Derivation

Below is the derivation of the transfer function with an input of F and an output of θ.

$$\ddot{\theta}(I + my_c^2) + \ddot{x}(my_c) + b\dot{\theta} - mgy_c\theta = 0 \tag{1}$$

$$\ddot{\theta}(my_c) + \ddot{x}(M + m) = F \tag{2}$$

Laplace transforms of equations 29 and 30 gives

$$\theta(I + my_c^2)s^2 + X(my_c)s^2 + b\theta s - mgy_c\theta = 0 \tag{3}$$

$$\theta(my_c)s^2 + X(M + m)s^2 = F \tag{4}$$

Next, solving 4 for x and substituting it into 3

$$X = \frac{F - \theta(my_c)s^2}{(M + m)s^2} \tag{5}$$

$$\theta(I + my_c^2)s^2 + \frac{(F - \theta(my_c)s^2)(my_c)s^2}{(M + m)s^2} + b\theta s - mgy_c\theta = 0 \tag{6}$$

Simplifying 31 leads to

$$\theta(I + my_c^2)s^2 + \frac{Fmy_c}{(M + m)} - \frac{\theta(my_c)^2s^2}{(M + m)} + b\theta s - mgy_c\theta = 0 \tag{7}$$

$$\theta[(I + my_c^2)s^2 - \frac{(my_c)^2s^2}{(M + m)} + bs - mgy_c] + \frac{Fmy_c}{(M + m)} = 0 \tag{8}$$

$$\theta[(I + my_c^2)s^2 - \frac{(my_c)^2s^2}{(M + m)} + bs - mgy_c] = -\frac{Fmy_c}{(M + m)} \tag{9}$$

$$\theta[(I + my_c^2)(M + m)s^2 - (my_c)^2s^2 + bs(M + m) - mgy_c(M + m)] = -Fmy_c \tag{10}$$

$$\frac{\theta}{F} = \frac{-my_c}{(I + my_c^2)(M + m)s^2 - (my_c)^2s^2 + bs(M + m) - mgy_c(M + m)} \tag{11}$$

$$\frac{\theta}{F} = \frac{-my_c}{[(I + my_c^2)(M + m) - (my_c)^2]s^2 + b(M + m)s - mgy_c(M + m)} \tag{12}$$

Equation 12 is the transfer function of θ / F without friction.

1.2 Transfer Function X / θ Derivation

Using 3

$$X(my_c)s^2 = -\theta[(I + my_c^2)s^2 + bs - mgy_c] = 0 \tag{13}$$

$$\frac{X}{\theta} = \frac{-[(I + my_c^2)s^2 + bs - mgy_c]}{(my_c)s^2} \tag{14}$$

Equation 14 is the transfer function of X / θ.

1.3 Transfer Function of θ / F with Friction Derivation

Below is the derivation of the transfer function with an input of F and an output of θ.

$$\ddot{\theta}(I + my_c^2) + \ddot{x}(my_c) + b\dot{\theta} - mgy_c\theta = 0 \tag{15}$$

$$\ddot{\theta}(my_c) + \ddot{x}(M + m) = F + F_{fric} \tag{16}$$

These equations are equivalent to 29 and 30 except 30 has F_{fric} added.

$$F_{fric} = -b_c art\dot{x} \tag{17}$$

Substituting 40 into 34

$$\ddot{\theta}(my_c) + \ddot{x}(M + m) = F - b_{cart}\dot{x} \tag{18}$$

Laplace Transforms of 33 and 41 give

$$\theta(I + my_c^2)s^2 + X(my_c)s^2 + b\theta s - mgy_c\theta = 0 \tag{19}$$

$$\theta(my_c)s^2 + X(M + m)s^2 = F - b_{cart}Xs \tag{20}$$

Solving 20 for X and solving substituting the result into 19 gives

$$X[(M + m)s^2 + b_{cart}s] = F - \theta(my_c)s^2 \tag{21}$$

$$X = \frac{F - \theta(my_c)s^2}{(M + m)s^2 + b_{cart}s} \tag{22}$$

$$\theta(I + my_c^2)s^2 + \frac{[F - \theta(my_c)s^2]my_cs^2}{(M + m)s^2 + b_{cart}s} + b\theta s - mgy_c\theta = 0 \tag{23}$$

Solving 42 for the transfer function

$$\theta(I + my_c^2)s^2[(M + m)s^2 + b_{cart}s] + Fmy_cs^2 - \theta(my_c)^2s^4 + b\theta s[(M + m)s^2 + b_{cart}s] - mgy_c\theta[(M + m)s^2 + b_{cart}s] = 0 \tag{24}$$

$$\theta[(I + my_c^2)s^2((M + m)s^2 + b_{cart}s) - (my_c)^2s^4 + bs((M + m)s^2 + b_{cart}s) - mgy_c((M + m)s^2 + b_{cart}s)] = -Fmy_cs^2 \tag{25}$$

$$\frac{\theta}{F} = (-my_cs^2)/((I + my_c^2)s^2((M + m)s^2 + b_{cart}s) - (my_c)^2s^4 + bs((M + m)s^2 + b_{cart}s) - mgy_c((M + m)s^2 + b_{cart}s)) \tag{26}$$

$$\frac{\theta}{F} = (-my_cs^2)/((I + my_c^2)s^2((M + m)s^2 + b_{cart}s) - (my_c)^2s^4 + bs((M + m)s^2 + b_{cart}s) - mgy_c((M + m)s^2 + b_{cart}s)) \tag{27}$$

$$\frac{\theta}{F} = (-my_cs)/(s^3(mMy_c^2 + Im + IM) + s^2(bm + bM + Ib_{cart} + mb_{cart}y_c^2) + s(bb_{cart} - gm^2y_c - gmMy_c) - gmb_{cart}y_c) \tag{28}$$

Equation 28 is the transfer function of θ / F with friction.

1.4 Figures

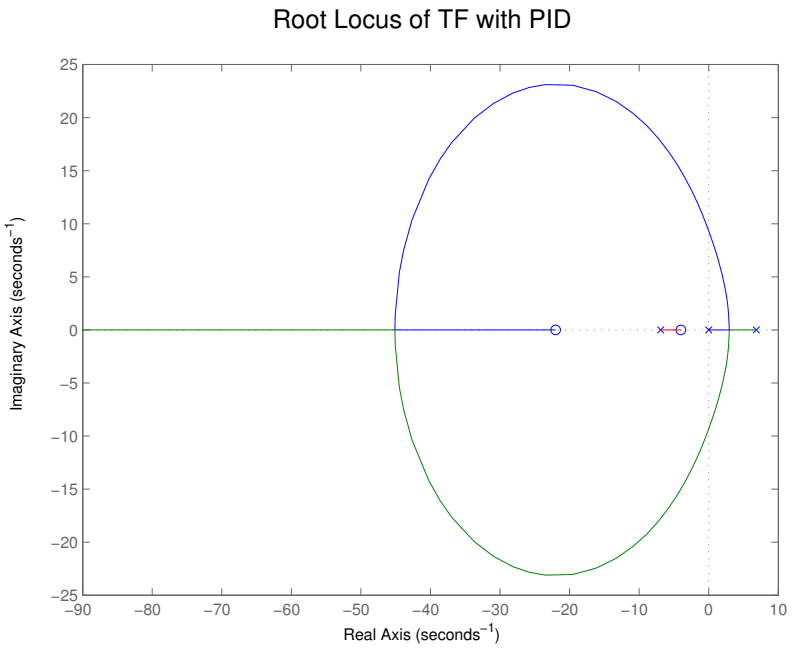


Figure 1: Rlocus of TF with PID Controller

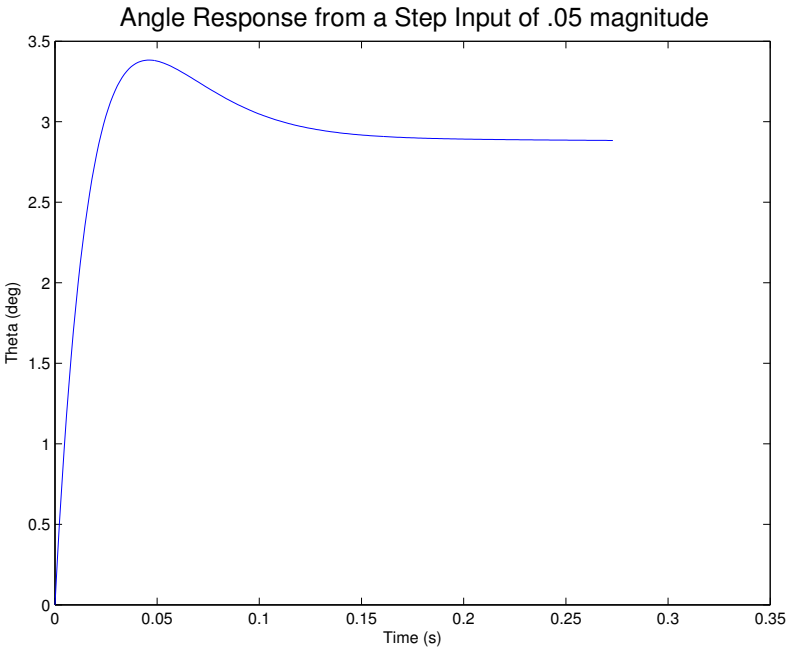


Figure 2: Angle Response for a Step Input ($\theta_0 = 0°$)

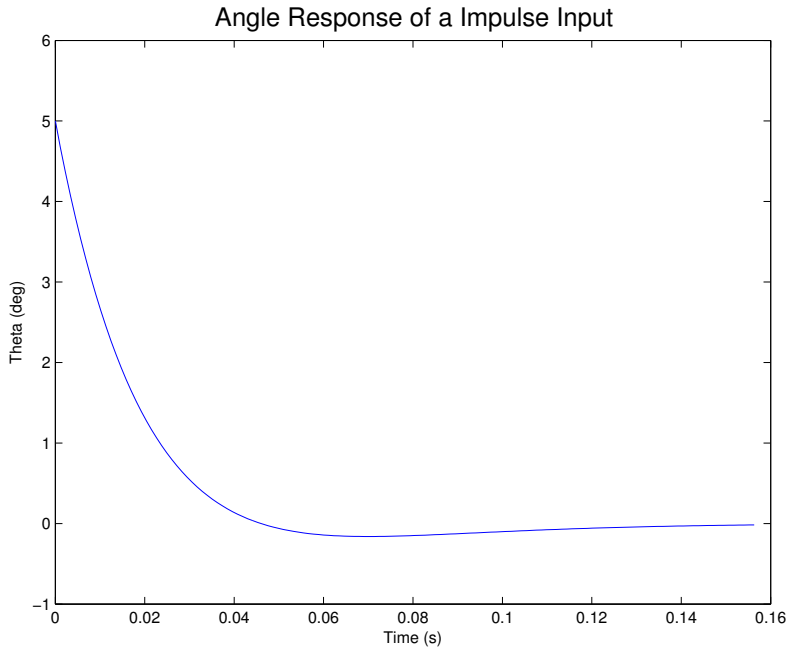


Figure 3: Angle Response for an Impulse Input ($\theta_0 = 5°$)

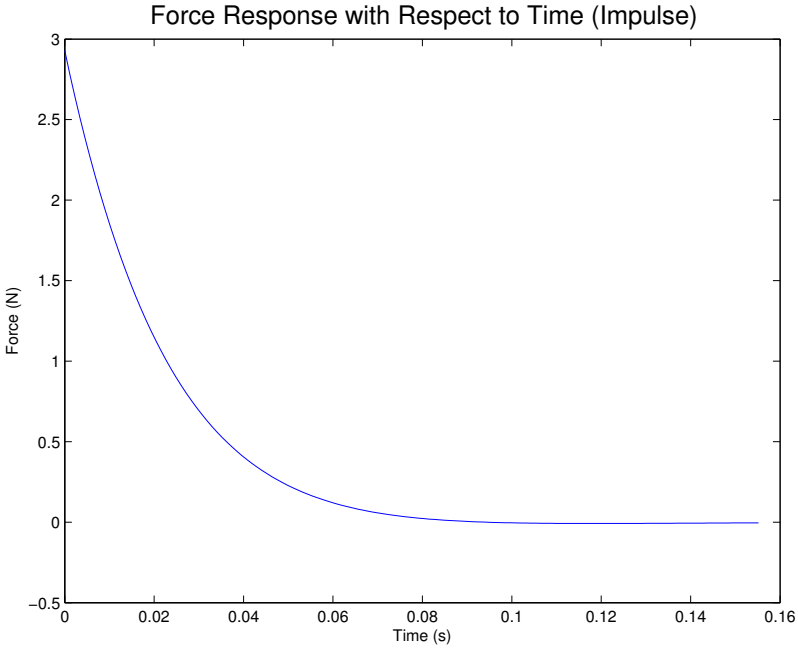


Figure 4: Force Response of an Impulse Input ($\theta_0 = 5°$)

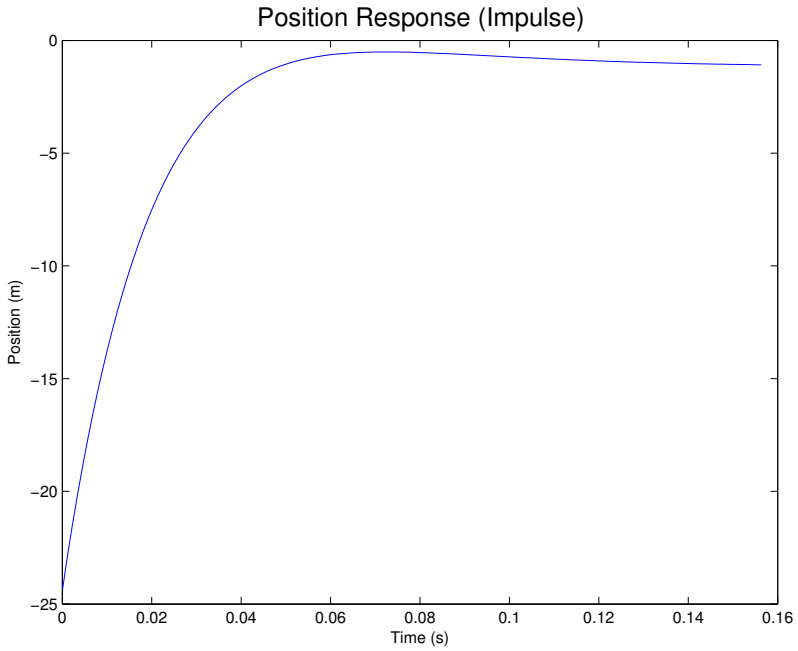


Figure 5: Position Response of an Impulse Input ($\theta_0 = 5°$)

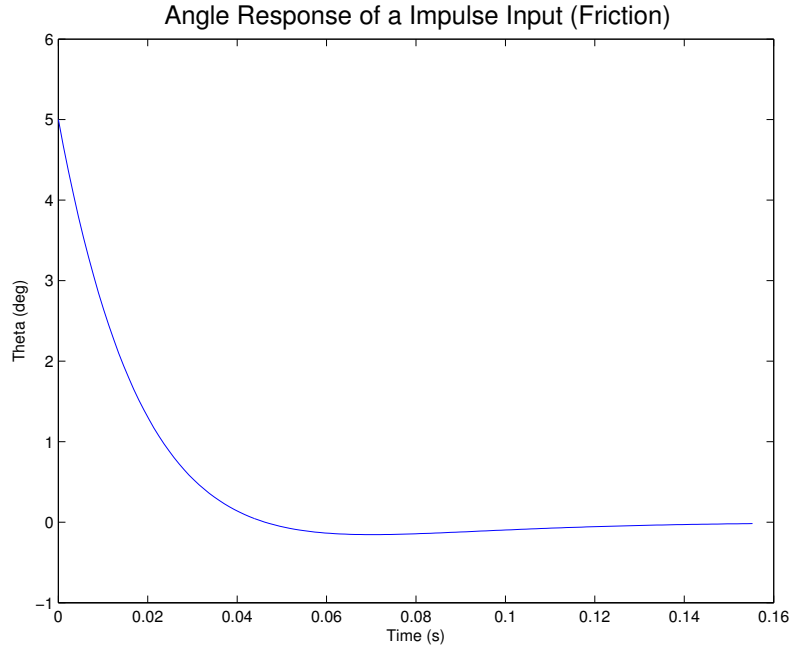


Figure 6: Angle Response of an Impulse Input with Friction ($\theta_0 = 5°$)

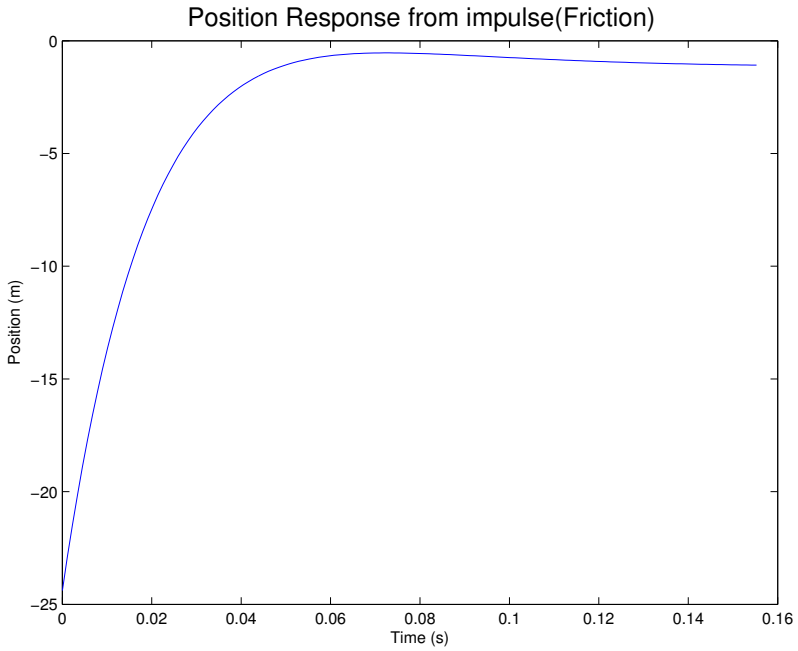


Figure 7: Position Response of an Impulse Input with Friction ($\theta_0 = 5°$)

1.5 Discussion

1.5.1 Requirements

My controller is stable, has 0% steady state error for step inputs, a settling time that is .292s of the maximum, an overshoot that is 51.4% of the maximum, and a force that is 97.7% below the maximum. All of these values are quite satisfying except for the force that is used. I think that if I had more time, I could adjust the poles for a better value. Note that a step input is impractical for the angle because we want it to be straight up. An impulse force is much more practical for the real world.

	Goal	Actual
Stability	Stable	Stable
Steady State Error (step input)	0%	0%
Settling Time (Max)	.5 s	.146 s
Overshoot (Max)	35%	18.1%
Force (Max)	3N	2.929N

1.5.2 Friction

Accounting for friction kept all my parameters the same or improved them except for the force used. This makes sense because it would take a larger force to overcome friction and move the pendulum the same distance. I am unsure of why the settling time and overshoot improved, my guess is that it is because the friction slows down the cart so that it the overshoot is less and the settling time improves because of this. The position response seems to be the same whether friction is accounted for or not. This is interesting because I thought that friction would slow down the cart until it stopped, but it keeps moving as long as the simulation runs. My current theory on this is that the model controls the position, so even if friction is added, the model is going to add more force to overcome the friction.

	Goal	Actual	Actual (With Friction)
Stability	Stable	Stable	Stable
Steady State Error (step input)	0%	0%	0%
Settling Time (Max)	.5 s	.146 s	.143 s
Overshoot (Max)	35%	18.1%	17.3%
Force (Max)	3N	2.929N	2.97N

2 LQR State Space Controller

2.1 4th Order

2.1.1 State-Space Modeling

Below is the free-body diagram of the inverted pendulum and cart

Summation of carts forces in the horizontal direction

$$F = M\ddot{x} + b\dot{x} + N \tag{29}$$

Summation of the pendulum forces in the horizontal direction

$$N = m\ddot{x} + ml\ddot{\phi}cos(\phi) - ml\dot{\phi}^2 sin(\phi) \tag{30}$$

Substituting 30 into 29

$$F = M\ddot{x} + b\dot{x} + m\ddot{x} + ml\ddot{\phi}cos(\phi) - ml\dot{\phi}^2 sin(\phi) \tag{31}$$

Next, the sum of forces perpendicular to the pendulum

$$Psin(\phi) + Ncos(\phi) - mgsin(\phi) = ml\ddot{\phi} + m\ddot{(}x)cos(\phi) \tag{32}$$

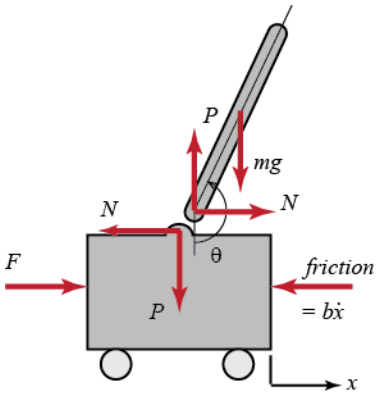


Figure 8: Source: Inverted Pendulum: System Modeling

Summing the moments about the centroid of the pendulum

$$-Psin(\phi) - Ncos(\phi) = I\ddot{\phi}/l \tag{33}$$

Combining 32 and 33 leads to

$$(I + ml^2)\ddot{\phi} + mglsin(\phi) = -ml\ddot{x}cos(\phi) \tag{34}$$

Small angle approximations. θ replaces ϕ so that the angle $\theta = 0\ ^\circ$ is straight up.

$$cos(\phi) = cos(\pi + \theta) \approx -1 \tag{35}$$

$$sin(\phi) = sin(\pi + \theta) \approx -\theta \tag{36}$$

$$\dot{\phi} = \dot{\theta} \approx 0 \tag{37}$$

Plugging 35 and 36 into 31 and 34 leads to

$$F = (M + m)\ddot{x} + b\dot{x} - ml\ddot{\theta} \tag{38}$$

$$(I + ml^2)\ddot{\theta} - mgl\theta = ml\ddot{x} \tag{39}$$

Motor Equations

$$Li + Ri = V - k_m\dot{\theta} \tag{40}$$

$$\tau = k_m * i \tag{41}$$

To arrive at state space equations add 40 and 41 into 38 and 39. Note: L = 0.

Plugging 41 into 40 leads to

$$F = \frac{Rk_m}{r(V - k_m\dot{\theta})} \tag{42}$$

Finally, plugging 42 into 38 and 39 and solving for $\ddot{x}$ and $\ddot{\theta}$ leads to the state space equations below

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{(I+ml^2)/(ml)(-(b+K^2/(Rr^2))}{(M+m)(I+ml^2)/(ml)-ml} & \frac{(I+ml^2)/(ml)(M+m)g}{[(M+m)(I+ml^2)/(ml)-ml]-g} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-(b+K^2/(Rr^2))}{(M+m)(I+ml^2)/(ml)-ml} & \frac{(M+m)g}{(M+m)(I+ml^2)/(ml)-ml} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{(I+m*l^2)/(m*l)*(K/(R*r)}{(M+m)(I+ml^2)/(ml)-ml} \\ 0 \\ \frac{K/(R*r)}{(M+m)(I+ml^2)/(ml)-ml} \end{bmatrix} v \tag{43}$$

$$\begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{44}$$

2.1.2 Poles, Stability, and Observability

Now that the state-space equations are found, the poles of the model are equal to the eigen values of the A matrix. The original Poles are shown below:

1. 0.00000 + 0.00000i
2. -9.9916 + 0.00000i
3. -4.9981 + 0.00000i
4. 5.1345 + 0.00000i

Because one of the poles is positive, the system is unstable. We will fix this with a controlled matrix A in the next section.

If the rank of the controllability matrix shown below equals the number of states the system (n) is made up of, then the system is controllable. In our case the rank of R equals 4 so our system is controllable. This means that with our controller, we can make the system reach any state.

$$R = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} \tag{45}$$

If the rank of the observability matrix shown below equals the number of states the system is made up of, then the system is observable. In our case the rank of O equals 4 so our system is observable. This means that in any possible situation, the current state of our system can be determined using only the inputs and outputs.

$$O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{46}$$

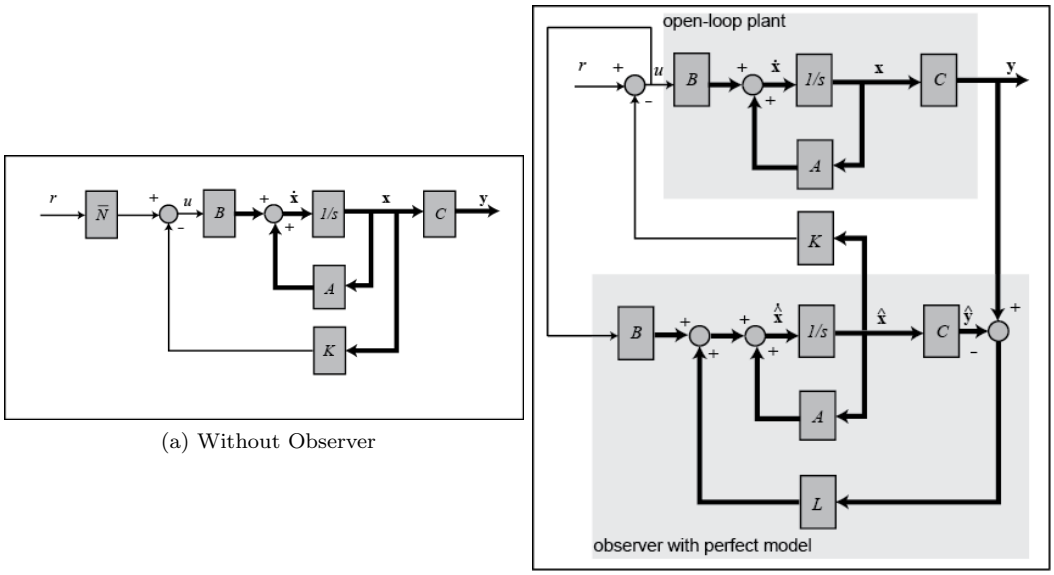
2.1.3 LQR Controller

A linear-quadratic regulator (LQR) weights factors depending on their importance using the n by n Q matrix, while minimizing another factor using the variable R. In our case, the weighted factors were the position of the cart (x) and the angle of the pendulum (θ). Also, the factor that we minimized is the velocity of the cart. A good starting point for Q is C'C evenly weighting the factors. However, we care much more about the pendulum staying vertical than we do the position staying at 0 m. Because of this, after some tests we weighted x at 1000 and θ at 100,000. leaving the Q matrix below. Also, a good place to start the limiting factor is 1, but we change R = 0.1.

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 100,000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{47}$$

In LQR controllers, the state-feedback control gains vector k is used to move the poles in the system. It can be calculated by minimizing the cost function J. We used Octave's LQR function to calculate our k vector. Once calculated, the corrected A matrix can be calculated using the equation below.

$$A_c = [A - Bk] \tag{48}$$



(a) Without Observer

(b) With Observer

Figure 9: Adding the observer,
Source: Inverted Pendulum: State-Space Methods for Controller Design

2.1.4 Precompensator

At this point, with a reference input commanding the pendulum to a certain position, the cart would move somewhere, but not to the desired position. Because of this, we had to add a precompensator to account for this error in position. We chose the C matrix below so that we would be controlling the position with our reference input:

$$C_n = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{49}$$

By replacing the state space C matrix with C_n, $\bar{N}$ can be calculated. We used the *rscale* function provided by Control Tutorials for Matlab & Simulink to calculate $\bar{N}$. $\bar{N}$ is a scale factor that eliminates the steady-state error of a system to a step input. It should be noted that $\bar{N}$ only works for single input systems, which is why we could use it to only eliminate the steady state error in the carts position. Once $\bar{N}$ is calculated, the state-space equation uses the corrected A matrix, B $\bar{N}$, C, and D for the A, B, C, and D matrices respectively.

2.1.5 Adding an Observer-Based Estimator

Previous work assumes full-state feedback, which means that x, $\dot{x}$, θ, and $\dot{\theta}$ are all being measure. However, only x and θ are being measured so we must add observer based control. Figure 9a is the previous block diagram, and Figure 9b is the updated block diagram with the observer-based estimator.

The speed of convergence depends on the poles of the estimator. Since we are going to use the estimation before the actual event occurs, we want the estimation to converge must faster. Because of this we want the poles of our estimator to be at least four times faster than the slowest controller pole. If the estimator poles are too fast it can cause error in the measurement, so usually the poles of the estimator should not be more than 10 times larger than the smallest controller pole. Also, the estimator poles should be close together. The current poles of our C matrix are listed below. Placing the estimator poles P at [-30, -31, -32, -33] was found to work after several trials. Note: we did not place all the estimator poles in the same position because the Octave *place* function used to find L does not support that.

1. -50.8860 +50.1382i
2. -50.8860 -50.1382i
3. -0.7136 + 0.6863i
4. -0.7136 - 0.6863i

Based on the estimator poles, we used the Octave place function which computes L. Based on the Original A, B, C, and D matrices for state space, the final state space equations are shown below:

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - Bk & Bk \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B\bar{N} \\ 0 \end{bmatrix} r \tag{50}$$

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} X \\ e \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} r \tag{51}$$

This state space model successfully keeps the pendulum up and the cart centered on the track. However, there is some noise which is not accounted for that causes the pendulum to oscillate more than it should. Also, originally this model only works with the wrong radius value for the motor which we used from a previous model that worked. However, after measuring the PWM input provided to the motor, we discovered the signal ranged from -30V to 30V instead of -20V to 20V like we were told. Once we made this change, the inverted pendulum worked with the correct radius.

2.2 6th Order

Our 6th order system takes account of the angle of the long pendulum and the position of the cart just like the 4th order system, but also the position of the short pendulum. We did this to see if it would eliminate some of the faulty state estimates we were having which caused our pendulum to oscillate. Also, these equations could be used to make both pendulums stay up.

We derived the equations almost exactly like the 4th order equations, except that there was another pendulum. The equations that we derived which are equivalent to 39, 38, and 42 are shown below

$$(m_{p1} l_1 We^2 + J_1)\ddot{\theta_1} - m_{p1} g l_1 \theta_1 = m_{p1} l_1 \ddot{x} \tag{52}$$

$$(m_{p2} l_2^2 + J_2)\ddot{\theta_2} + m_{p2} g l_2 \theta_2 = -m_{p2} l_2 \ddot{x} \tag{53}$$

$$(M_c + m_{p1} + m_{p2})\ddot{x} + b_c \dot{x} - m_{p1} l_1 \ddot{\theta_1} + m_{p2} l_2 \ddot{\theta_2} = \frac{KmV}{Rrd} - \frac{Km^2 \dot{x}}{Rrd^2} \tag{54}$$

Any variable with subscript 1 represents the long pendulum and subscript 2 the short pendulum. All other variables are the same as stated previously. Note that θ_2 is zero when pointing straight down. By solving these equations for $\ddot{x}$, $\ddot{\theta_1}$, and $\ddot{\theta_2}$ in Maple we found

$$\ddot{x} = (-(Rb_c l_1^2 l_2^2 m_{p1} m_{p2} r_d^2 + J_1 Rb_c l_2^2 m_{p2} r_d^2 + J_2 Rb_c l_1^2 m_{p1} r_d^2 + K_m^2 l_1^2 l_2^2 * m_{p1} m_{p2} + J_1 J_2 Rb_c r_d^2 + J_1 K_m^2 l_2^2 m_{p2} + J_2 K_m^2 l_1^2 m_{p1} + J_1 J_2 K_m^2)/p_1)\dot{x} - ((-Rgl_1^2 l_2^2 m_{p1}^2 m_{p2} r_d^2 - J_2 Rgl_1^2 m_{p1}^2 r_d^2)/p_1)\theta_1 - ((Rgl_1^2 l_2^2 m_{p1} m_{p2}^2 r_d^2 + J_1 Rgl_2^2 m_{p2}^2 r_d^2)/p_1\theta_2 - (-K_m l_1^2 l_2^2 m_{p1} m_{p2} r_d - J_1 K_m l_2^2 m_{p2} r_d - J_2 K_m l_1^2 m_{p1} r_d - J_1 J_2 K_m r_d)V/p_1 \quad (55)$$

$$\ddot{\theta_1} = -(Rb_c l_2^2 m_{p2} r_d^2 + J_2 Rb_c r_d^2 + K_m^2 * l_2^2 * m_{p2} + J_2 * K_m^2) * l_1 * m_{p1}\dot{x}/p_1 - (-M_c Rgl_2^2 m_{p2} r_d^2 - Rgl_2^2 m_{p1} m_{p2} r_d^2 - 2Rgl_2^2 m_{p2}^2 r_d^2 - J_2 M_c Rgr_d^2 - J_2 Rgm_{p1} r_d^2 - J_2 Rgm_{p2} r_d^2) l_1 m_{p1}\theta_1/p_1 - gl_2^2 m_{p2}^2 l_1 m_{p1}\theta_2/p_1 - (-K_m l_2^2 m_{p2} r_d - J_2 K_m r_d) l_1 m_{p1} V/p_1 \quad (56)$$

$$\ddot{\theta_2} = -m_{p2} l_2 (Rb_c l_1^2 m_{p1} r_d^2 + J_1 Rb_c r_d^2 + K_m^2 l_1^2 m_{p1} + J_1 K_m^2)\dot{x}/p_1 + m_{p2} l_2 gl_1^2 mp1^2 \theta_1/p_2 - m_{p2} l_2 (-M_c Rgl_1^2 m_{p1} r_d^2 - Rgl_1^2 m_{p1} m_{p2} r_d^2 - J_1 M_c Rgr_d^2 - J_1 Rgm_{p1} r_d^2 - J_1 Rgm_{p2} r_d^2)\theta_2/p_1 + m_{p2} l_2 (-K_m l_1^2 m_{p1} r_d - J_1 K_m r_d)/p_1 \quad (57)$$

where

$$p_1 = (Rr_d^2 (M_c l_1^2 l_2^2 m_{p1} m_{p2} + 2l_1^2 l_2^2 m_{p1} m_{p2}^2 + J_1 M_c l_2^2 m_{p2} + J_1 l_2^2 m_{p1} m_{p2} + 2J_1 l_2^2 m_{p2}^2 + J_2 M_c l_1^2 m_{p1} + J_2 l_1^2 m_{p1} m_{p2} + J_1 J_2 M_c + J_1 J_2 m_{p1} + J_1 J_2 m_{p2})) \quad (58)$$

and

$$p_2 = (M_c l_1^2 l_2^2 m_{p1} m_{p2} + 2l_1^2 l_2^2 m_{p1} m_{p2}^2 + J_1 + M_c l_2^2 m_{p2} + J_1 l_2^2 m_{p1} m_{p2} + 2J_1 l_2^2 m_{p2}^2 + J_2 M_c l_1^2 m_{p1} + J_2 l_1^2 m_{p1} m_{p2} + J_1 J_2 M_c + J_1 J_2 m_{p1} + J_1 J_2 m_{p2})$$

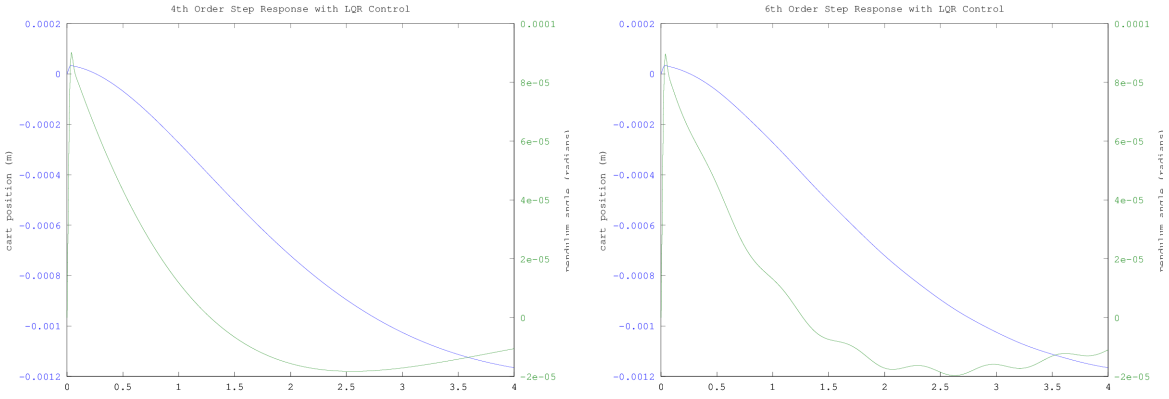
After plugging these equations into state space form and actually running it on the pendulum, we had more oscillation than on the 4th order when using the wrong voltage and radius. We are not sure why this is – the 6th order model should have better results than the 4th order model. Possible explanations include our 6th order model not being tuned as well as our 4th order or that there is more error in our small pendulum model than if we just ignore it. However, the oscillation decreased with this model using the correct values.

2.3 8th Order

We derived 8th order equations which include the springiness of the wire which moves the cart only to discover that the matrix had a rank of 7 meaning that the state space equations were not controllable. This is because when modeling the movement of the wire as a spring, the position of the cart is not known.

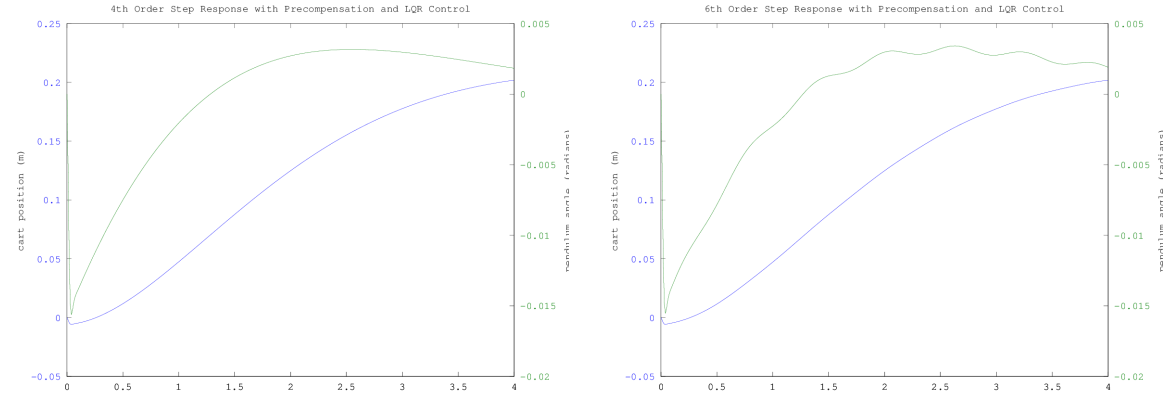
3 Results

Figure 10a and 10b show the step response to a reference input telling the cart to go to 0.2 meters. You will notice that it doesn't actually go to 0.2 meters. This is fixed by multiplying the reference input by $\bar{N}$. The result is Figure 11a and 11b, where the cart does go to a position of 0.2 meters. Also note that Figure 11b has some higher frequency components that is due to it caring about the short pendulum in addition to the long pendulum.



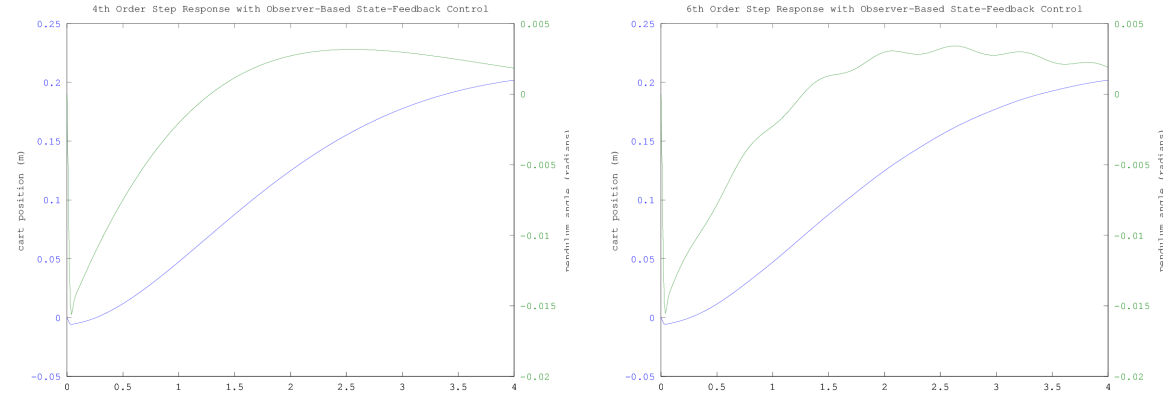
(a) 4th Order (b) 6th Order

Figure 10: Step Response with LQR Control



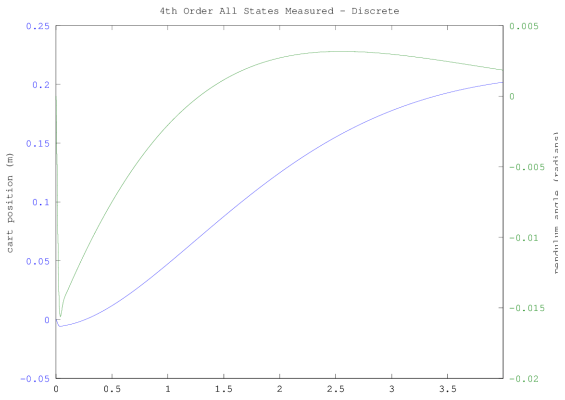
(a) 4th Order (b) 6th Order

Figure 11: Step Response with Precompensation and LQR Control

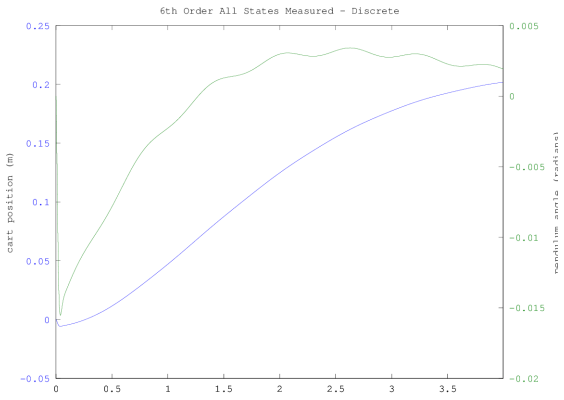


(a) 4th Order (b) 6th Order

Figure 12: Step Response with Observer Based State Feedback Control

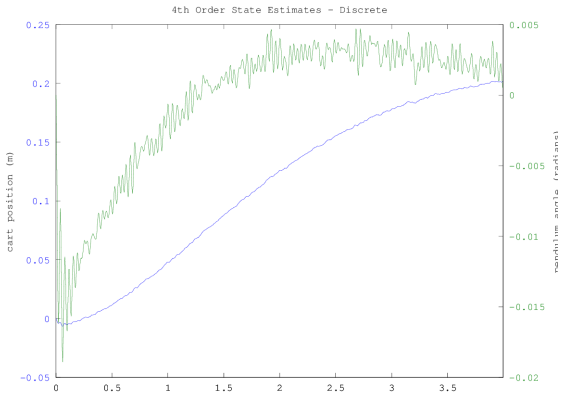


(a) 4th Order

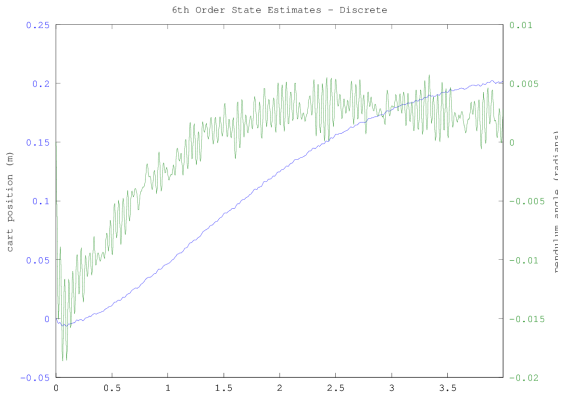


(b) 6th Order

Figure 13: Discrete: All States Measured



(a) 4th Order



(b) 6th Order

Figure 14: Discrete: All State Estimations

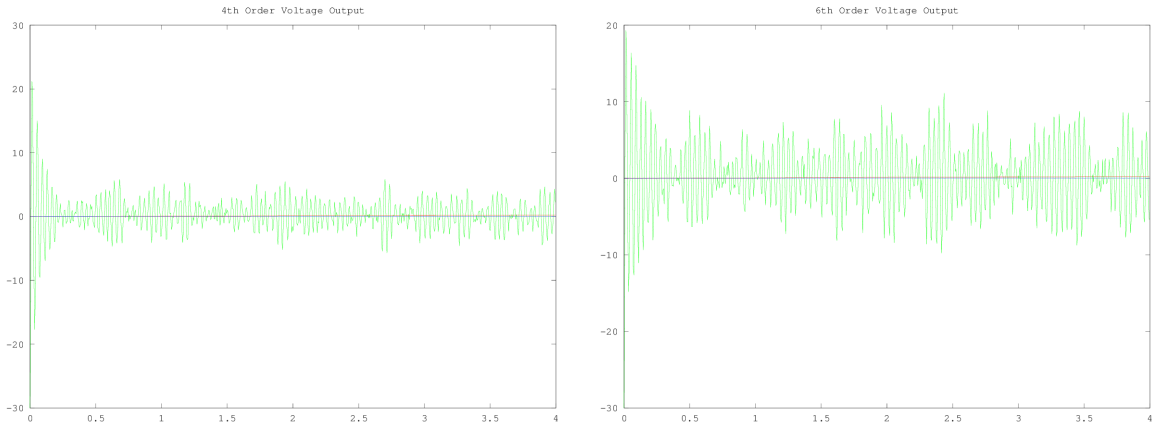
However, we don't have sensors for all of the states of these systems. We are only measuring the cart position and the pendulum angles. Figure 12a and 12b show the result when using a full-order observer. It is very close to what happened when using the actual state measurements in simulation.

When we do this on the inverted pendulum in the real world, we will be doing it digitally, so we need to discretize our system. We are doing this with *c2d* in Octave. To verify that our sampling rate won't cause issues, we can run the simulation on this discrete system. If the sampling rate is too low, the effect will be visible on these plots.

Next, Figure 14a and 14b show using our discretized estimator and also adding some Gaussian noise to our measurements to make this more realistic since the actual sensors might not have perfect readings.

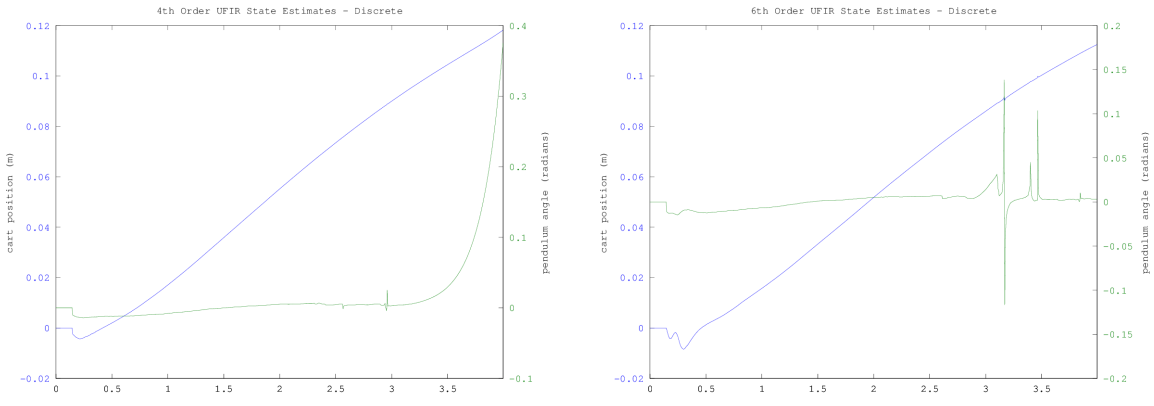
Since we are dealing with a real motor and power supply, there are limits to what voltage we can apply to the motor. Figure 15a and 15b show what voltage is output to the motor from our system in simulation. This will allow us to see if we are driving the motor too hard, which will be visible on this plot as a square wave oscillating between the max and min output voltages.

We wanted to estimate our states since our full-order observer in our test runs didn't quite match reality. Thus we tried a Kalman-like unbiased FIR filter as shown in Figure 16a and 16b. However, more work is required on this to get correct state estimates.



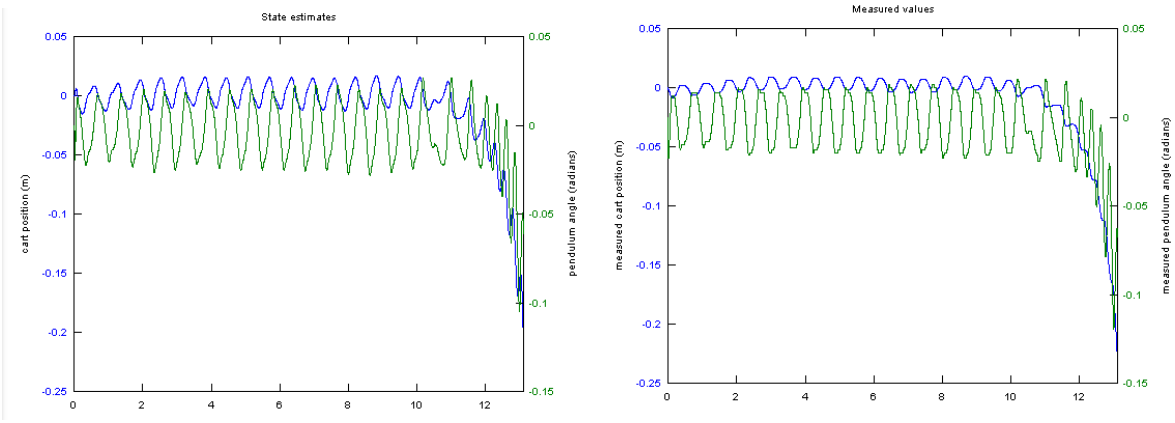
(a) 4th Order (b) 6th Order

Figure 15: Discrete: Voltage Output to Motor



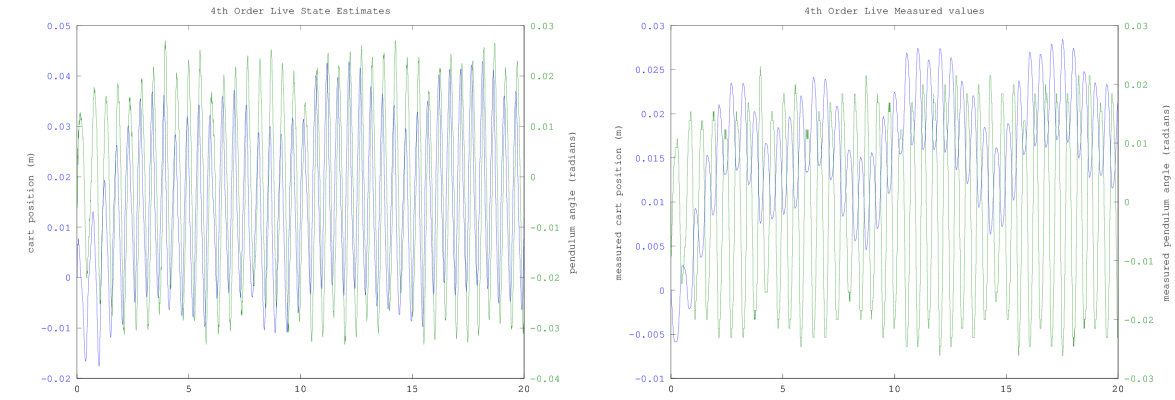
(a) 4th Order (b) 6th Order

Figure 16: Discrete: UFIR State Estimates



(a) Estimated (b) Measured

Figure 17: 4th Order With 20V Max Voltage



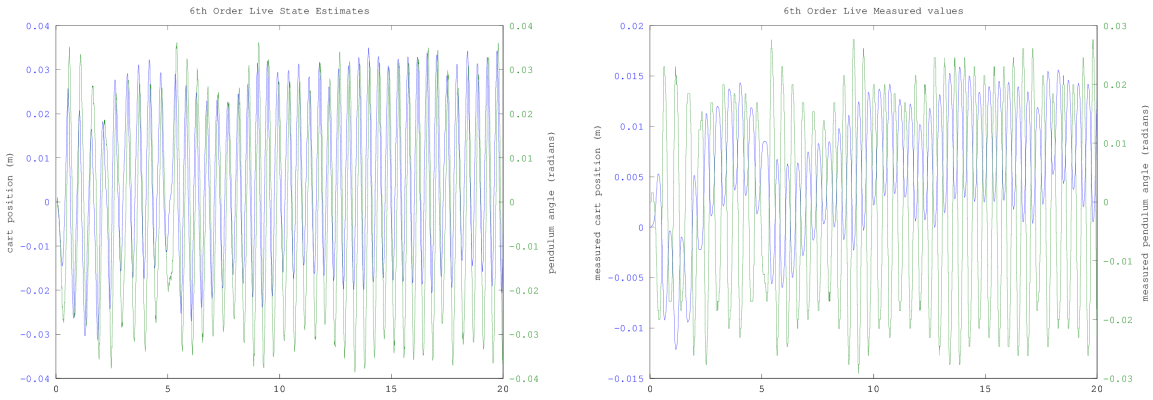
(a) Estimated (b) Measured

Figure 18: 4th Order With 30V Max Voltage

Figure 17a and 17b use the voltage level we were given in the list of system parameters, which was 20 V. These graphs have the correct radius for the motor. They show the long pendulum staying inverted for approximately 12 seconds.

Figure 18a, 18b, 19a, and 19b use the voltage level we measured on the output of the motor, which was 30 V. The state estimations are far from the actual values we measured as shown in Figure 18b and 19b. The 6th order system keeps the pendulum centered around a position of $x = 0$ much more consistently than the 4th order system in our tests. This is what we would think should happen since the 6th order system takes into account the small pendulum hanging down. However, note that we used the small angle approximation for the both pendulums, so if the small pendulum is swinging wildly, then our state estimates might not follow as close to reality as we would hope.

In addition, we found that the program that is generally used to demo the inverted pendulum, which was created by Andrew Roth, multiplied the measured motor angle by r_d in two places, which is the same as setting r_d to be the square of what we would expect. When we made this alteration, both the 4th and 6th order system worked better. The correct value is $r_d = 0.0127$ m, but in this case we used $r_d = 0.00016129$ m. However, the system acted similarly when we used the correct radius and used $Maxvoltage = 30$ V instead of $Maxvoltage = 20$ V.



(a) Estimated

(b) Measured

Figure 19: 6th Order With 30V Max Voltage

4 Future Work

- Fix the UFIR state estimates. This requires figuring out why the algorithm depends on $inv(A*F*A')$, which is a problem since all of our A matrices are non-invertible. An alternative is using the Octave package called ALS, which uses a Kalman filter and provides a method to determine the variances. However, in theory, the UFIR will outperform the Kalman filter in the majority of cases; though, it is more computationally intensive.
- We could verify that the system parameters we are using are correct. For example, we could measure the velocity of the motor when we spin the motor at a certain velocity by an external force, which would allow us to check the motor's K value.
- Instead of verifying system parameters, we could also do system identification to determine all the system parameters by providing white noise as the input and doing a best fit.
- While it won't necessarily provide any improvement, it would be fairly simple to take the PID in State-Space form equations to test a PID controller with our existing code.
- We could try getting the short pendulum to stay up while the long one hangs down, or we could work more on getting both pendulums to stay up simultaneously.

5 References

- A Kalman-like FIR estimator ignoring noise and initial conditions
- Iterative Unbiased FIR State Estimation: A Review of Algorithms
- ALS Octave Package
- Explanation of Autocovariance Least-Squares (ALS) Method