ELEC ENG 4CL4

Lab Project Phase 3-4 Report
State-space Control of Inverted Pendulum

Max Poddoubnyi – poddoum

Sara Jamil – jamils2

Abstract

In order to control the dynamics of the inverted pendulum mechanism a state space controller was implemented. The state space controller consisted of a linear state-feedback controller and a linear state observer based on the linearized model of the system at the two equilibrium states. The purpose a state-feedback controller is to determine the states of the system and use the measured states as the input to the system to maintain an equilibrium position. The need for a state observer in this system was to determine the angular velocities of each link. The angular velocities could not be measured directly from the encoders on the system which only measure the angle of each link. The state space controller was simulated in Simulink and then tested on the actual inverted pendulum mechanism.

Open-loop Response

The using the following linearized model:

$$\dot{x} = \boldsymbol{A}x + \boldsymbol{B}u$$

$$where\ x = \begin{matrix} q1 \\ q2 \\ \dot{q}1 \\ \dot{q}2 \end{matrix} = \begin{matrix} x1 \\ x2 \\ x3 \\ x4 \end{matrix} and\ u = \tau$$

$$y = \boldsymbol{C}x$$

$$Where\ \boldsymbol{A} = \begin{bmatrix} 0_{2x2} & I_{2x2} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{F} \end{bmatrix} \in \mathbb{R}^{4\,x\,4}\ ,\ \boldsymbol{B} = \begin{bmatrix} 0 \\ \boldsymbol{M}^{-1} \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \mathbb{R}^{4\,x\,1}\ ,\ \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{2\,x\,4}$$

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{\theta_1} + 2\boldsymbol{\theta_2}\cos(x_{20}) & \boldsymbol{\theta_3} + \boldsymbol{\theta_2}\cos(x_{20}) \\ \boldsymbol{\theta_3} + \boldsymbol{\theta_2}\cos(x_{20}) & \boldsymbol{\theta_3} \end{bmatrix},$$

$$\boldsymbol{K} = \begin{bmatrix} -\boldsymbol{\theta_4} * g * sin(x_{10}) - sin(x_{10} + x_{20})\boldsymbol{\theta_5} * g & -sin(x_{10} + x_{20})\boldsymbol{\theta_5} * g \\ -sin(x_{10} + x_{20})\boldsymbol{\theta_5} * g & -sin(x_{10} + x_{20})\boldsymbol{\theta_5} * g \end{bmatrix},\ \boldsymbol{F} = \begin{bmatrix} \boldsymbol{\theta_4} & 0 \\ 0 & \boldsymbol{\theta_4} \end{bmatrix},$$

$$\Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \\ \Theta_4 \\ \Theta_5 \\ \Theta_6 \\ \Theta_7 \end{bmatrix} = \begin{bmatrix} (m_1\, l_{c1}^2 + \, m_2(l_1^2 + l_{c2}^2) + I_1 \, + \, I_2)/k \\ m_2 l_1 l_{c2}/k \\ (m_2 l_{c2}^2 + \, I_2)/k \\ (m_1 l_{c1} + m_2 l_1)/k \\ m_2 l_{c2}/k \\ b_{f1}/k \\ b_{f2}/k \end{bmatrix} = \begin{bmatrix} 11.8421 \\ 0.4216 \\ 0.1005 \\ 5.6072 \\ 1.1523 \\ 28.1160 \\ 0.0111 \end{bmatrix}$$

Station 3 values of Θ were used

Using the following linearized model the poles of the system were obtained for each equilibrium. At the equilibrium when both links are vertically upright.

$$x0 = [\pi/2 \quad 0 \quad 0 \quad 0]^T$$

The eigenvalues of **A** which are the poles of the system are located at:

$$eig(A) = \begin{bmatrix} 11.5450 \\ -11.5450 \\ 2.1574 \\ -2.1574 \end{bmatrix}$$

The system is unstable because the poles are located to the right of the $j\omega$-axis which grow exponentially. This result is consistent with the responses obtained from the full nonlinear model.
The Simulink model used is as shown Figure 1: (The input to the system is $u = 0.01 * \sin(t) + \frac{\pi}{2}$)

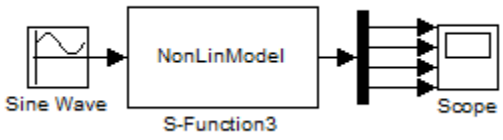


Figure 1: Non Linear System Model: Stability Test

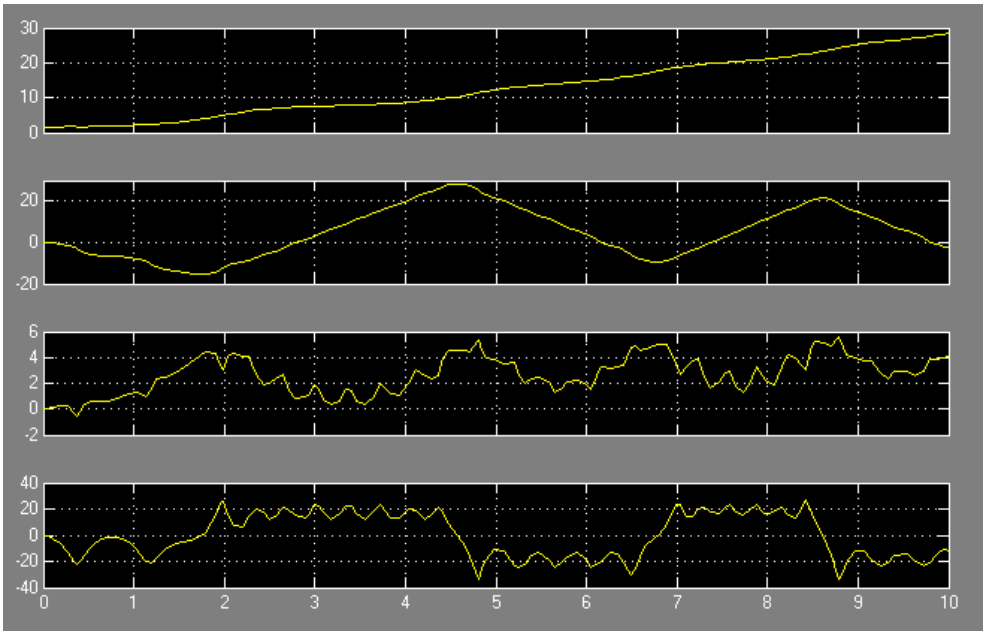


Figure 2: Output of Stability Test at Equilibrium 1

The result of inserting a bounded input results in the angle of Link 1 growing exponential (as seen in Figure 2) which is consistent with the prediction that the open loop system is unstable at this equilibrium. At the equilibrium when the first link is vertically upright and the second link is hanging downward.

$$x0 = [\pi/2 \quad \pi \quad 0 \quad 0]^T$$

The eigenvalues of **A** which are the poles of the system are located at:

$$eig(A) = \begin{bmatrix} j11.4727 \\ -j11.4727 \\ 2.1710 \\ -2.1710 \end{bmatrix}$$

The system is also unstable because the poles are located to the right of the $j\omega$-axis which grow exponentially. Using the same Simulink model in Figure 1 with the same input as before the nonlinear

model confirms that the this system is unstable. The output of the nonlinear system is shown in Figure 3. In figure 3, it is evident that q1 and q2 grow exponentially with a bounded input which makes the system unstable.

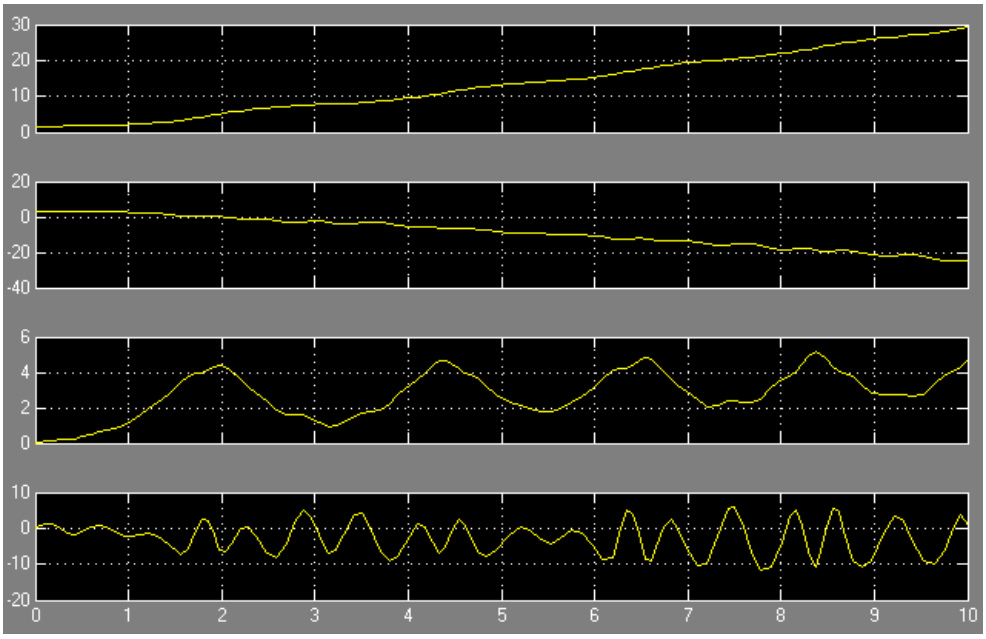


Figure 3: Output of Stability Test at Equilibrium 2

To determine e^{At}:

1. Determine the eigenvalues of A: $|\lambda I - A|$ = the characteristic polynomial and the roots are the eigenvalues (λ_i) for i = 1, 2, … , n.
2. $e^{At} = T\, e^{\lambda_i t} T^{-1}$ *where T is the eigenvectors of* λ_i *with*

$$e^{\lambda_i t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & \dots & 0 \\ 0 & e^{\lambda_2 t} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & e^{\lambda_n t} \end{bmatrix}$$

3. $x(t) = e^{At}\, x(0)$
4. $y(t) = Cx(t) + Du(t)$

Again for each equilibrium e^{At} was determined using the linearized dynamics.
For the unstable equilibrium when both links are vertically upright:

MATLAB:

```
%% Linearization Point
q1 = pi/2;
q2 = 0;
X0=[q1+.1 q2-.2 0 0];

%% Computing e^At
syms t
[V,D] = eig(A) % returns eigenvectors and eigenvalues
eDiagonal = [exp(D(1,1)*t) 0 0 0;
```

```
        0 exp(D(2,2)*t) 0 0;
        0 0 exp(D(3,3)*t) 0;
        0 0 0 exp(D(4,4)*t)];
eAt = V*eDiagonal*inv(V);

old = digits(5); % changes accuracy to 5 decimal points
vpa(eAt)
digits(old);

xOfT = eAt*X0.';
yOfT = C*xOfT;
time = 0:0.001:2.5;
yoftime = subs(yOfT,t,time);

figure()
plot(time, real(yoftime(1,:)),'r',time, real(yoftime(2,:)),'b');
legend('q1','q2','Location','northwest');
title('Output to Initial State')
xlabel('Time (seconds)')
ylabel('angle (rads)')
```

eAt =

[0.51514*exp(2.1574*t) - 0.015136*exp(11.545*t) - 0.015136*exp(-11.545*t) + 0.51514*exp(-2.1574*t), 0.018483*exp(2.1574*t) - 0.018483*exp(11.545*t) - 0.018483*exp(-11.545*t) + 0.018483*exp(-2.1574*t), 0.23877*exp(2.1574*t) - 0.001311*exp(11.545*t) + 0.001311*exp(-11.545*t) - 0.23877*exp(-2.1574*t), 0.008567*exp(2.1574*t) - 0.0016009*exp(11.545*t) + 0.0016009*exp(-11.545*t) - 0.008567*exp(-2.1574*t)]

[0.42185*exp(11.545*t) - 0.42185*exp(2.1574*t) + 0.42185*exp(-11.545*t) - 0.42185*exp(-2.1574*t), 0.51514*exp(11.545*t) - 0.015136*exp(2.1574*t) + 0.51514*exp(-11.545*t) - 0.015136*exp(-2.1574*t), 0.03654*exp(11.545*t) - 0.19553*exp(2.1574*t) - 0.03654*exp(-11.545*t) + 0.19553*exp(-2.1574*t), 0.04462*exp(11.545*t) - 0.0070156*exp(2.1574*t) - 0.04462*exp(-11.545*t) + 0.0070156*exp(-2.1574*t)]

[1.1114*exp(2.1574*t) - 0.17474*exp(11.545*t) + 0.17474*exp(-11.545*t) - 1.1114*exp(-2.1574*t), 0.039875*exp(2.1574*t) - 0.21338*exp(11.545*t) + 0.21338*exp(-11.545*t) - 0.039875*exp(-2.1574*t), 0.51514*exp(2.1574*t) - 0.015136*exp(11.545*t) - 0.015136*exp(-11.545*t) + 0.51514*exp(-2.1574*t), 0.018483*exp(2.1574*t) - 0.018483*exp(11.545*t) - 0.018483*exp(-11.545*t) + 0.018483*exp(-2.1574*t)]

[4.8702*exp(11.545*t) - 0.91011*exp(2.1574*t) - 4.8702*exp(-11.545*t) + 0.91011*exp(-2.1574*t), 5.9472*exp(11.545*t) - 0.032654*exp(2.1574*t) - 5.9472*exp(-11.545*t) + 0.032654*exp(-2.1574*t), 0.42185*exp(11.545*t) - 0.42185*exp(2.1574*t) + 0.42185*exp(-11.545*t) - 0.42185*exp(-2.1574*t), 0.51514*exp(11.545*t) - 0.015136*exp(2.1574*t) + 0.51514*exp(-11.545*t) - 0.015136*exp(-2.1574*t)]

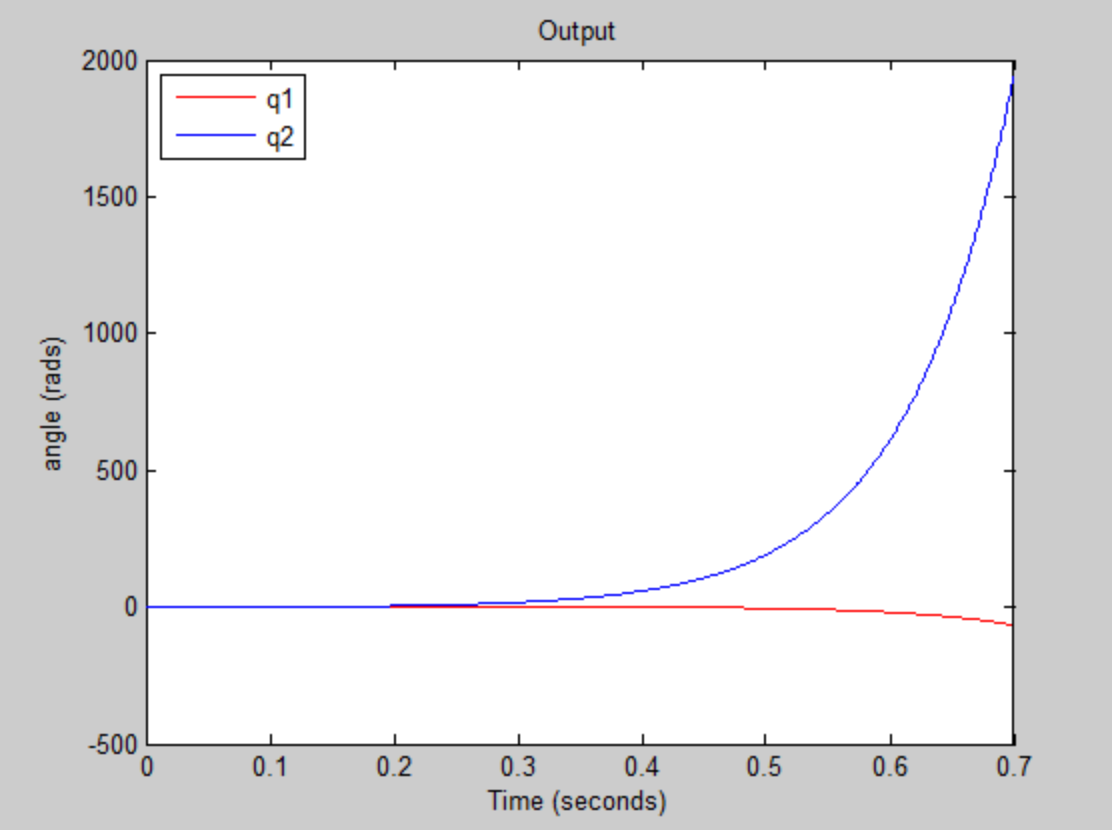


Figure 4

Using the linearized dynamics in the Simulink environment with the same initial states X0 the output of the system is shown in figure 6 which is identical to the MATLAB response in figure 4.

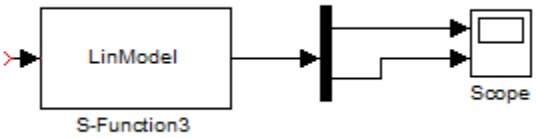


Figure 5 Block Diagram of Linearized Model in Simulink

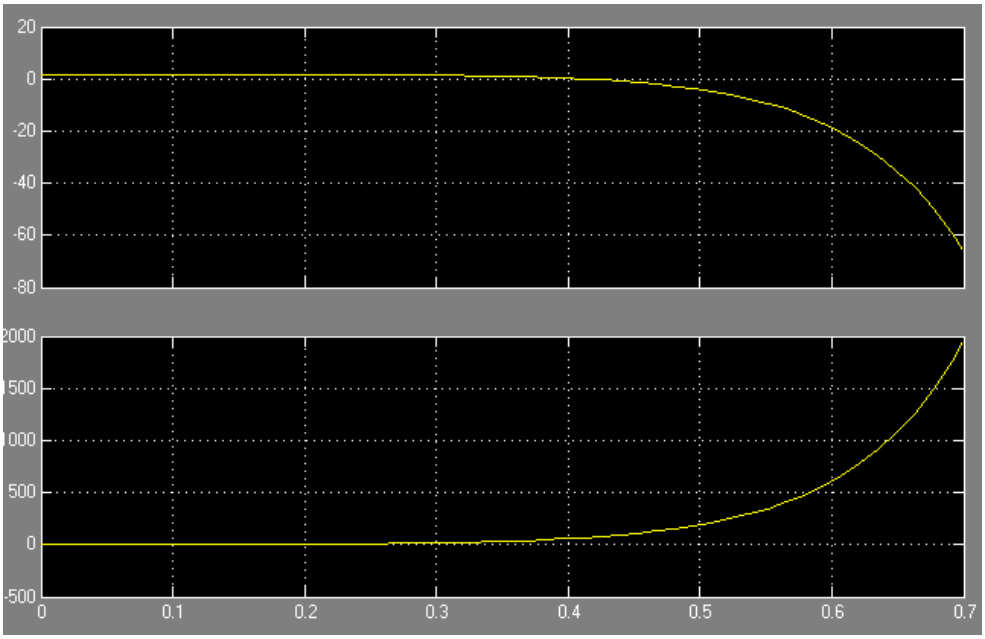


Figure 6 Simulink Output for Equilibrium 1

For the second equilibrium when the first link is vertically upright and the second link is hanging downward. To find e^{At} for the second equilibrium the only changes required to the MATLAB code above was to change q2 to equal to pi. The resulting e^{At} is:

eAt =

[exp(t*(- 1.2282e-15 + 11.473*i))*(0.014496 - 1.6618e-17*i) + exp(2.171*t)*(0.4855 + 1.5887e-19*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(0.014496 + 1.6627e-17*i) + exp(-2.171*t)*(0.4855 + 1.5887e-19*i), exp(t*(- 1.2282e-15 + 11.473*i))*(0.017438 - 2.4097e-17*i) + exp(2.171*t)*(- 0.017438 - 5.7063e-21*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(0.017438 + 2.4097e-17*i) + exp(-2.171*t)*(- 0.017438 - 5.7063e-21*i), exp(t*(- 1.2282e-15 + 11.473*i))*(- 4.061e-19 - 0.0012635*i) + exp(2.171*t)*(0.22363 + 7.3178e-20*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(- 4.061e-19 + 0.0012635*i) + exp(-2.171*t)*(- 0.22363 + 7.3178e-20*i), exp(t*(- 1.2282e-15 + 11.473*i))*(- 1.7314e-18 - 0.00152*i) + exp(2.171*t)*(- 0.0080323 + 2.7905e-19*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(- 1.7314e-18 + 0.00152*i) + 0.0080323*exp(- 2.171*t)]

[exp(t*(- 1.2282e-15 + 11.473*i))*(0.40359 + 9.4628e-19*i) + exp(2.171*t)*(- 0.40359 - 1.3207e-19*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(0.40359 - 6.8215e-19*i) + exp(-2.171*t)*(- 0.40359 - 1.3207e-19*i), exp(t*(- 1.2282e-15 + 11.473*i))*(0.4855 - 1.1319e-16*i) + exp(2.171*t)*(0.014496 + 4.7435e-21*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(0.4855 + 1.1318e-16*i) + exp(-2.171*t)*(0.014496 + 4.7435e-21*i), exp(t*(- 1.2282e-15 + 11.473*i))*(2.9104e-17 - 0.035178*i) + exp(2.171*t)*(- 0.1859 - 6.0832e-20*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(2.9104e-17 + 0.035178*i) + exp(-2.171*t)*(0.1859 - 6.0832e-20*i), exp(t*(- 1.2282e-15 + 11.473*i))*(4.061e-19 - 0.042318*i) + exp(2.171*t)*(0.0066771 - 2.3197e-19*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(4.061e-19 + 0.042318*i) - 0.0066771*exp(-2.171*t)]

[exp(t*(- 1.2282e-15 + 11.473*i))*(1.9509e-16 + 0.16631*i) + exp(2.171*t)*(1.054 + 3.4491e-19*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(1.952e-16 - 0.16631*i) + exp(-2.171*t)*(- 1.054 - 3.4491e-19*i), exp(t*(- 1.2282e-15 + 11.473*i))*(2.818e-16 + 0.20006*i) + exp(2.171*t)*(- 0.037859 - 1.2388e-20*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(2.818e-16 - 0.20006*i) + exp(-2.171*t)*(0.037859 + 1.2388e-20*i), exp(t*(- 1.2282e-15 + 11.473*i))*(0.014496 - 5.046e-18*i) + exp(2.171*t)*(0.4855 + 1.5887e-19*i) +

exp(t*(- 1.2282e-15 - 11.473*i))*(0.014496 + 5.046e-18*i) + exp(-2.171*t)*(0.4855 - 1.5887e-19*i), exp(t*(- 1.2282e-15 + 11.473*i))*(0.017438 - 2.033e-17*i) + exp(2.171*t)*(- 0.017438 + 6.0582e-19*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(0.017438 + 2.033e-17*i) - 0.017438*exp(-2.171*t)]

[exp(t*(- 1.2282e-15 + 11.473*i))*(- 1.2525e-15 + 4.6303*i) + exp(2.171*t)*(- 0.8762 - 2.8672e-19*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(- 1.2495e-15 - 4.6303*i) + exp(-2.171*t)*(0.8762 + 2.8672e-19*i), exp(t*(- 1.2282e-15 + 11.473*i))*(- 1.9509e-16 + 5.57*i) + exp(2.171*t)*(0.031471 + 1.0298e-20*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(- 1.952e-16 - 5.57*i) + exp(-2.171*t)*(- 0.031471 - 1.0298e-20*i), exp(t*(- 1.2282e-15 + 11.473*i))*(0.40359 + 4.4213e-16*i) + exp(2.171*t)*(- 0.40359 - 1.3207e-19*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(0.40359 - 4.4213e-16*i) + exp(-2.171*t)*(- 0.40359 + 1.3207e-19*i), exp(t*(- 1.2282e-15 + 11.473*i))*(0.4855 + 1.3485e-16*i) + exp(2.171*t)*(0.014496 - 5.0361e-19*i) + exp(t*(- 1.2282e-15 - 11.473*i))*(0.4855 - 1.3485e-16*i) + 0.014496*exp(-2.171*t)]

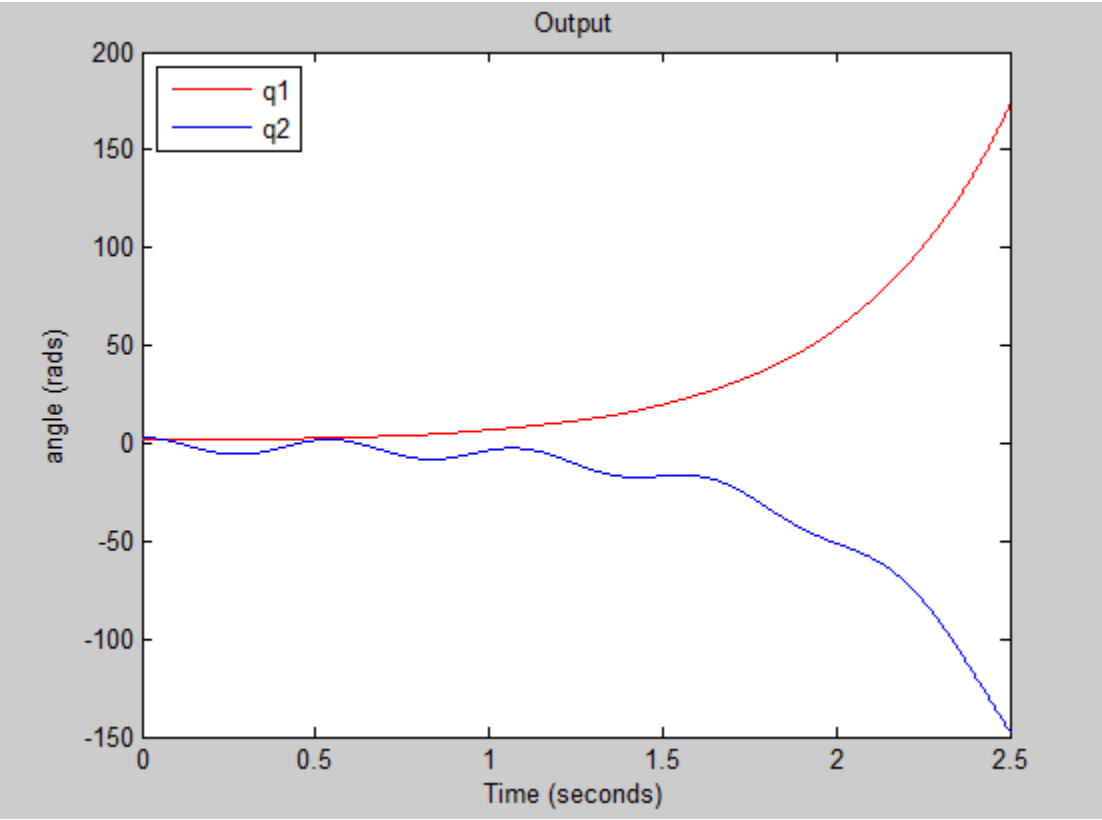


Figure 7

Using the linearized dynamics in the Simulink environment with the same initial states the output of the system is shown in figure 8 which also vertically identical to the MATLAB response in figure 7.

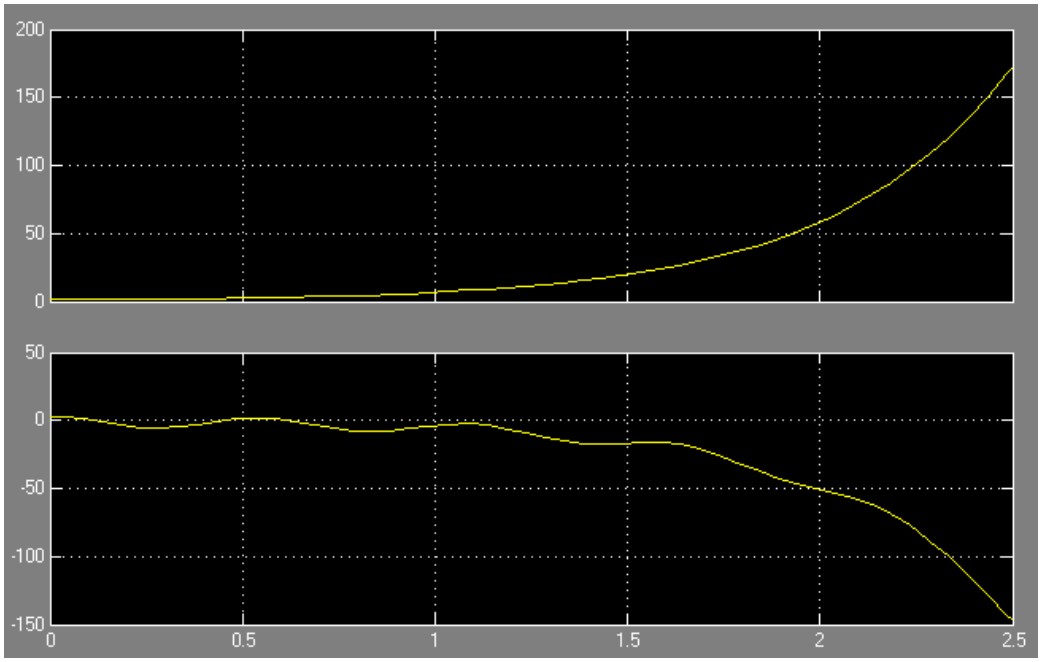


Figure 8 Simulink Output for Equilibrium 2

State-feedback Control

To determine if a system is fully controllable, the rank of the following controllability matrix has to be full rank which is the same as the number of states.

$controllability\ matrix: P_c = [B\ \ AB\ \ A^2B \ldots\ A^{n-1}B],\ rank(P_c) = n\ if\ fully\ controllable$

For the first equilibrium when both links are vertical upright:
The MATLAB function ctrb(A,B) was used to determine the controllability matrix

```
Pc =

         0    0.1003         0    2.5532
         0   -0.5209         0  -60.5765
    0.1003         0    2.5532         0
   -0.5209         0  -60.5765         0
```

And $rank(P_c) = 4$ which means that this system is fully controllable at this equilibrium.
For the second equilibrium when the first link is vertical upright and the second link is hanging downwards:

```
Pc =

         0    0.1003         0   -1.4471
         0    0.3204         0  -51.9361
    0.1003         0   -1.4471         0
    0.3204         0  -51.9361         0
```

And $rank(P_c) = 4$ which also means that this system is fully controllable at this equilibrium.

If the system is fully controllable that means that all the poles in the system can be changed in the following way.

For a state-feedback controller the input is in the following form: $\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}$
By substituting $\boldsymbol{u}$ in the linear state-space dynamics the result is: $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}$
From the new linear state-space equation it can be seen that the original 'A' matrix is replaced with 'A-BK' which changes the systems poles because the eigenvalues are now determined from 'A-BK'.
The choice of the 'K' matrix will put the poles of system in the desired location. To determine the K matrix the MATLAB place() function was used. K = place(A,B,p) which computes the state-feedback gains that would put the closed-loop poles in their desired locations specified by the vector p.

For the first equilibrium when both links are vertically upright:
The closed pole positions were chosen to be:

```
p =

   -15   -14   -11   -12
```

The state-feedback gain K resulted in:

```
K =

   1.0e+03 *

   -5.7022   -3.2994   -1.0743   -0.3066
```

The impact of the new poles of this system is that they are all to the left of the $j\omega$-axis and are all real decaying exponential which causes any transient behaviour to become come zero quickly. Also the poles are not extremely close to the zero which results in faster response time to disturbances or position changes.

For the second equilibrium when the first link is upright and the second link is hanging downwards:
The closed pole positions were chosen to be:

$$\mathrm{p} = [-2.0000 + 1.0000\mathrm{i}, -2.0000 - 1.0000\mathrm{i}, \quad -1.0000, \quad -9.0000]$$

The state-feedback gain K resulted in:

```
k =

  -129.4226 -188.4192   10.8141   31.1349
```

The impact of the closed poles of this system from the open loop poles is that they are all to the left of the $j\omega$-axis and are decaying exponentials which will decay the transient behaviour to zero. Also the imaginary component of conjugate pair poles was reduced from 11.431 to 1 which will result in slower oscillations. The slowest pole in the system is at -1 which limits the response time of the system.

To visualize the closed loop response of the system, a Simulink model was used for the linearized model and the non-linear model. Figure 9 shows the block diagram of the linearized model used and figure 10 is

the block diagram of the non-linear model.

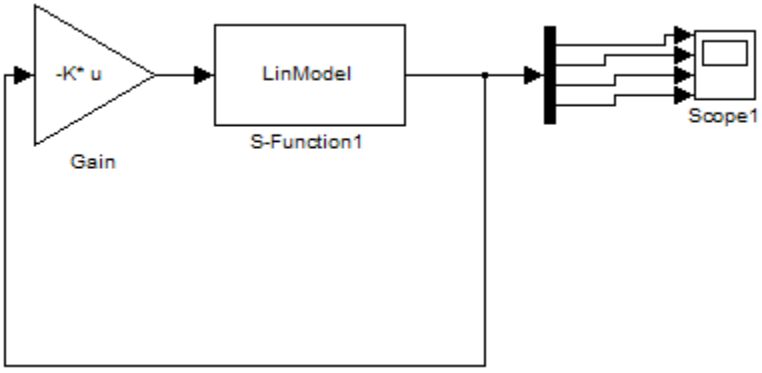


Figure 9 Block Diagram of Linearized Model in Simulink using state-feedback coontrol

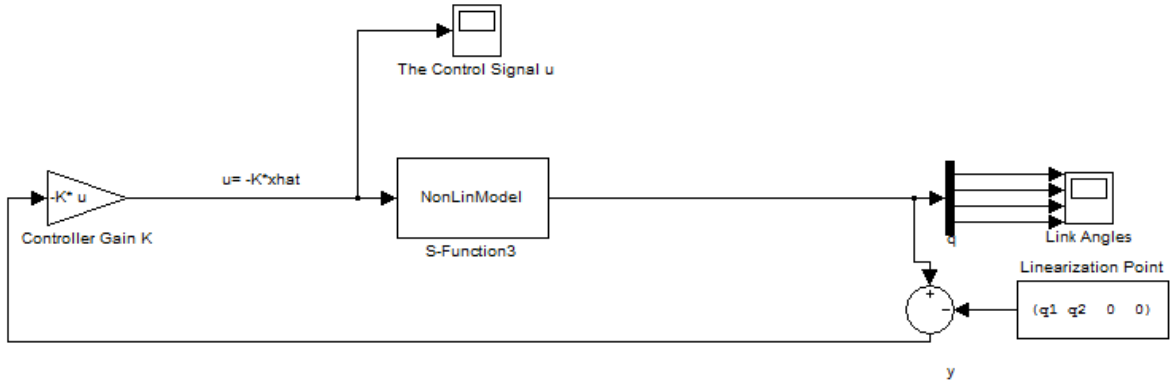


Figure 10 Block Diagram of Non-Linear Model in Simulink using state-feedback coontrol

For the first equilibrium with both links vertically upright the closed loop response of the linearized system is shown in figure 11 when the initial state is $x0 = [\frac{\pi}{2} + 0.1 \quad -0.2 \quad 0 \quad 0]^T$ and the nonlinear system response for the same initial state is shown in figure 12. The subplots in each figure corresponded to x1, x2, x3, and x4 from top to bottom. As seen in the linearized response all states converge to zero after less than 1 second which makes the system responses quick and stable due to the convergence to zero. From the non-linear model the response that is produced shows that the state feed-back controller is able to stabilize the system at the equilibrium position $x = [\frac{\pi}{2} \quad 0 \quad 0 \quad 0]^T$ in less than one second with a quick response that was predicted from the eigenvalues chosen. It is in equilibrium because the angular velocities of both links are zero after one second.

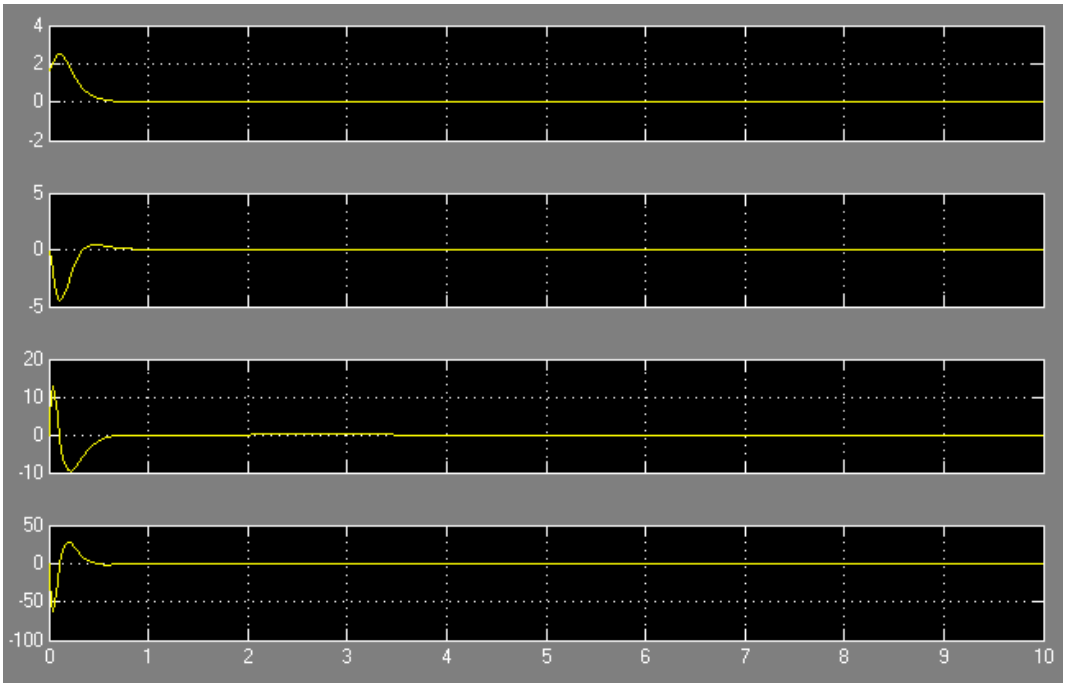


Figure 11: Closed Loop Response for Linearized Model

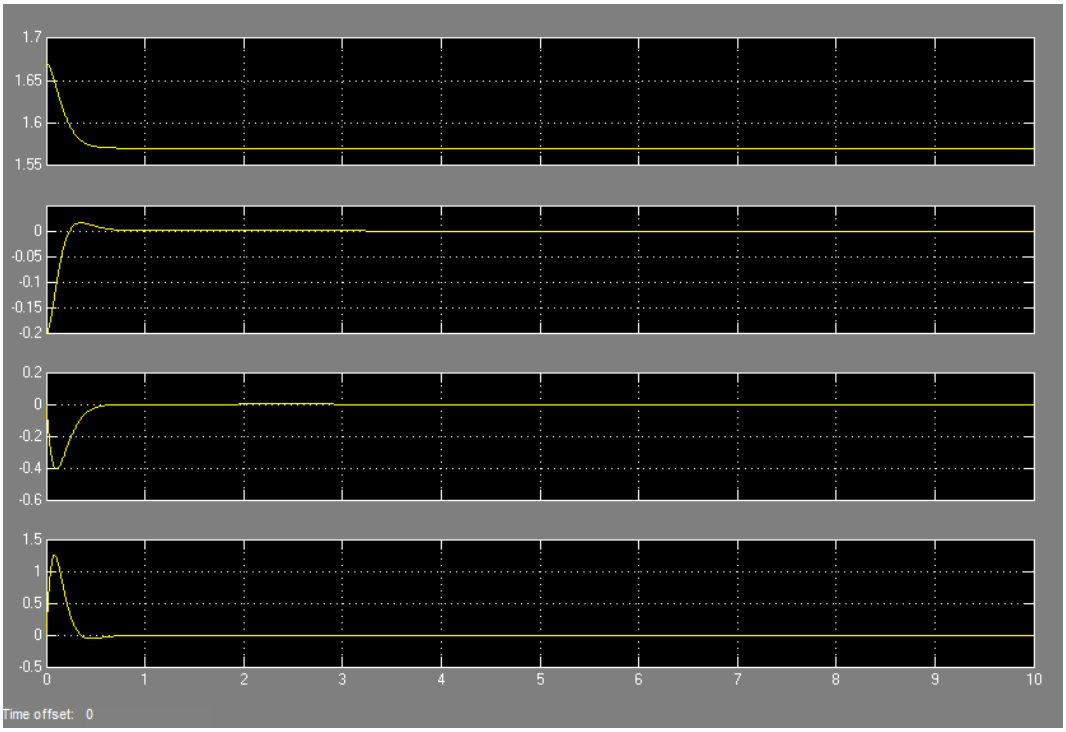


Figure 12: Closed Loop Response for Non-Linear Model

For the second equilibrium with first links vertically upright and the second link hanging downwards the closed loop response of the linearized system is shown in figure 13 when the initial state is $x0 = [\frac{\pi}{2} + 0.1 \quad \pi - 0.2 \quad 0 \quad 0]^T$ and the nonlinear system response for the same initial state is shown in figure 14. The subplots in each figure corresponded to x1, x2, x3, and x4 from top to bottom. As seen in the linearized response all states converge to zero after around 6 seconds which makes the system responses slow which is due to the pole at -1, but it is stable due to the convergence of all states to zero. From the non-linear model the response that is produced shows that the state-feedback controller is also able to stabilize the system at the second equilibrium position$x = [\frac{\pi}{2} \quad \pi \quad 0 \quad 0]^T$. The response of this controller is slow due to the pole at -1 and produces slow oscillation because of the complex conjugate poles with small imaginary components. However it reaches equilibrium because the angular velocities of both links are zero after five seconds. The limitation of the state-feedback controller is that if the rank controllability matrix was less the number of states then there would be a state that would be uncontrollable. And if the uncontrollable was to the right of the $j\omega$-axis then this type of controller would not be able to make the system stable. Furthermore if the K values chosen are very large due to very fast poles then the physical system may not be able to produce these gains causing the input to saturate resulting in non-linear responses.

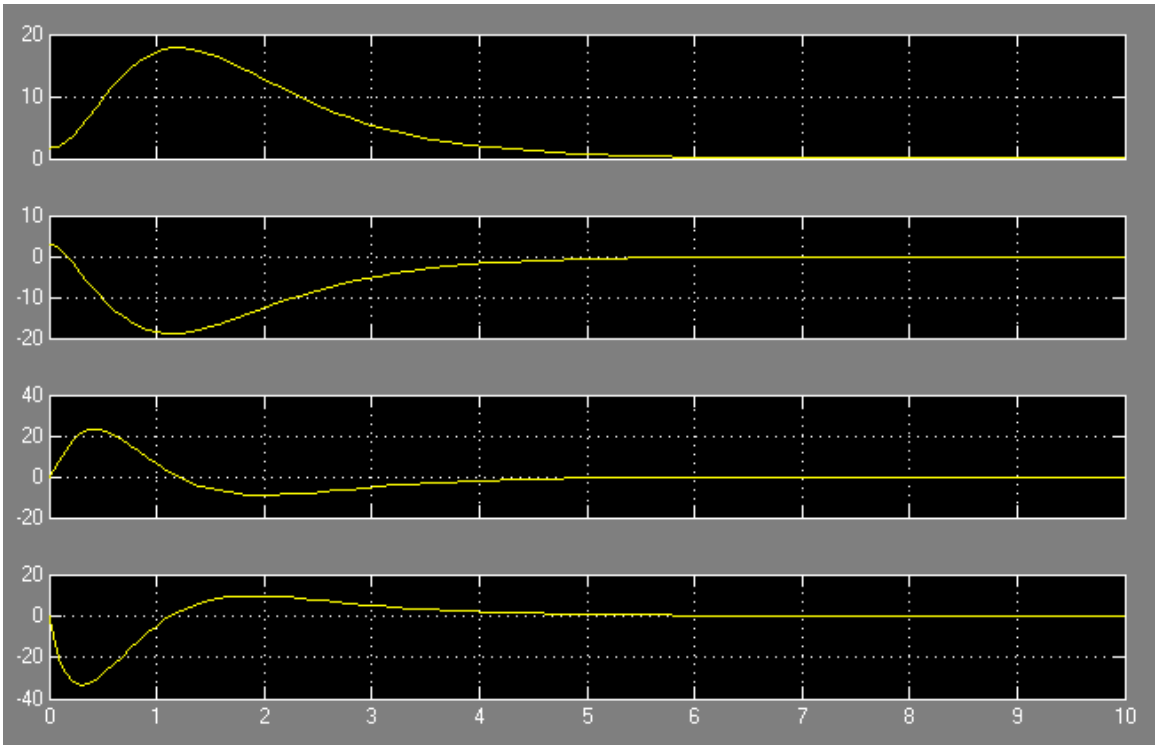


Figure 13: Closed Loop Response for Linearized Model

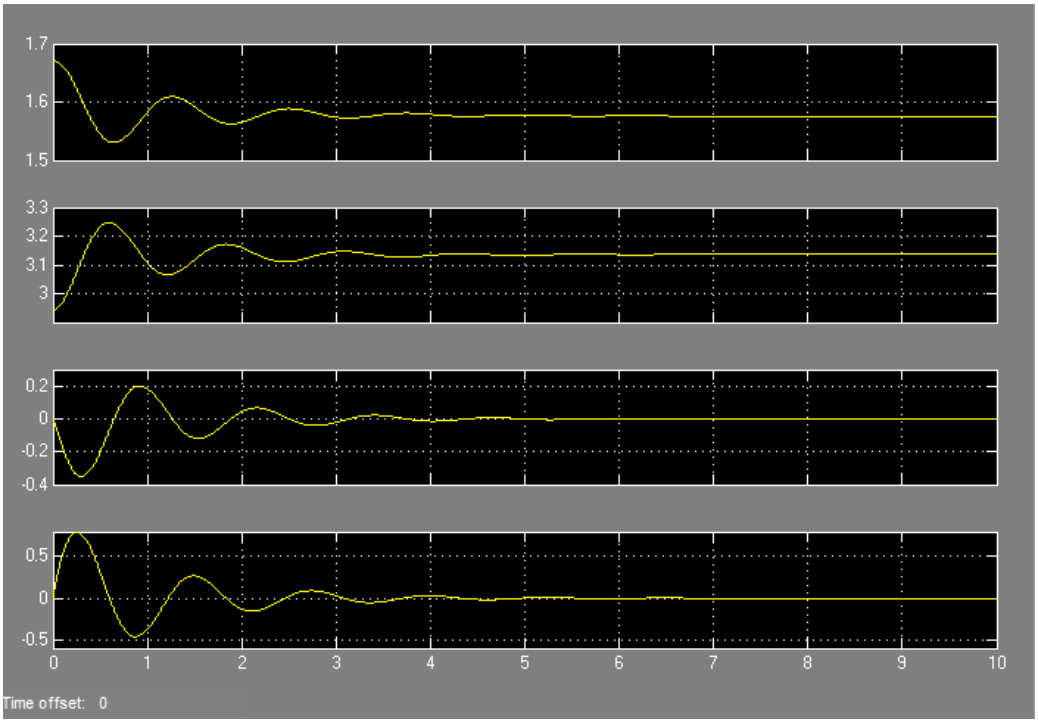


Figure 14: Closed Loop Response for Non-Linear Model

State Estimator (Observer)

There is a need for a state estimator in this system because the inverted pendulum only measures the angles of link 1 and link 2, but the state space controller also requires the angular velocities of each link. In the linear state-space dynamics the unknown states in x can be estimated from the output measurement of y using the following equation:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \; where \; \hat{x} \; is \; the \; estimated \; state \; and \; L \; is \; the \; oberver \; gain$$

The error in the estimated state is commuted as:

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

$$\dot{e} = A\hat{x} + Bu + L(y - C\hat{x}) - [Ax + Bu] = A(\hat{x} - x) + L(Cx - C\hat{x}) = [A - LC](\hat{x} - x) = [A - LC]\, e$$

The solution the above equation is: $e(t) = exp(\Lambda t)e0 \; where \; \Lambda \; is \; eignvalues \; of \; A - LC$
Thus if the eigenvalues are negative then the error will converge to zero, but in order to move all these poles to a desired location the system the system must be full observable. In order for a system to be fully observable its observability matrix must be full rank which is equal to the number of states.

$$observability \; matrix: P_o = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}, \; rank(P_o) = n \; if \; fully \; observable$$

Now using the MATLAB function obsv(A,C) the observability matrix for each equilibrium was obtained.
For the first equilibrium when both links are vertical upright:

```
Po =

    1.0000         0         0         0
         0    1.0000         0         0
         0         0    1.0000         0
         0         0         0    1.0000
    0.7606   -4.7549         0         0
  108.5267  137.1802         0         0
         0         0    0.7606   -4.7549
         0         0  108.5267  137.1802
```

And $rank(P_o) = 4$ which means that this system is fully observable at this equilibrium.
For the second equilibrium when the first link is vertical upright and the second link is hanging downwards:

```
Po =

    1.0000         0         0         0
         0    1.0000         0         0
         0         0    1.0000         0
         0         0         0    1.0000
    0.7606   -4.7549         0         0
 -110.0480 -127.6704         0         0
         0         0    0.7606   -4.7549
         0         0 -110.0480 -127.6704
```

And $rank(P_o) = 4$ which also means that this system is also fully observable at this equilibrium.

With both equilibriums being fully observable the eigenvalues of 'A-LC' can be placed at any desired location. The pole positions for each equilibrium were determined to be about ten times that of the state-feedback control pole positions to ensure that the state estimates converge to the actual state quickly. The observer gain matrix 'L' can be determined using the MATLAB place function.
$L^T = place(A^T, C^T, p_e)$ *where* p_e *is the vector of the desired eigenvalues*

For the first equilibrium when both links are vertical upright:
The closed pole positions were chosen to be:

```
po =

  -150  -140  -110  -120
```

The observe gain L resulted in:

```
L =

   1.0e+04 *

    0.0230         0
         0    0.0290
    1.3201   -0.0005
    0.0109    2.1137
```

For the second equilibrium when the first link is upright and the second link is hanging downwards: The closed pole positions were chosen to be:

$$p0 = [-10.0000 + 1.0000i, -10.0000 - 1.0000i, \quad -15.000, \quad -90.000]$$

The observe gain L resulted in:

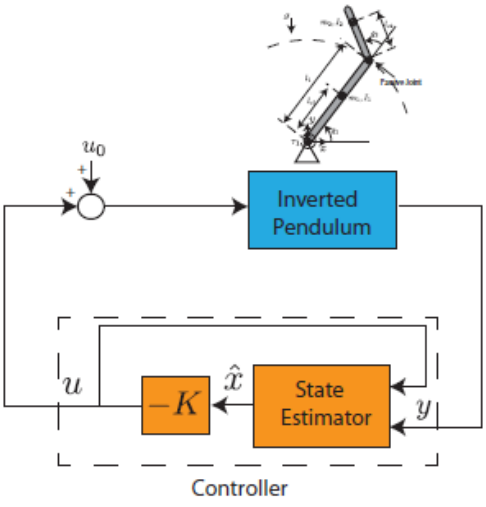
```
L =

   24.9564    1.1530
    0.1381  100.0436
  151.2324   85.6872
 -113.6955  776.2875
```

The desired poles chosen are all to the left of the $j\omega$-axis and are all fast decaying exponential.

State-feedback Control with State Estimation

Linearized Model

The final implementation of the closed loop system consisting of the state-feedback controller and the state estimator is shown to have the following system dynamics



$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) = (A - LC)\hat{x} + Bu + Ly$$
$$u = -K\hat{x}$$

Figure 2: Inverted Pendulum with Observer-based State-feedback Controller

In the simulation of the response of the system using the linearized model, the following Simulink configuration was used.

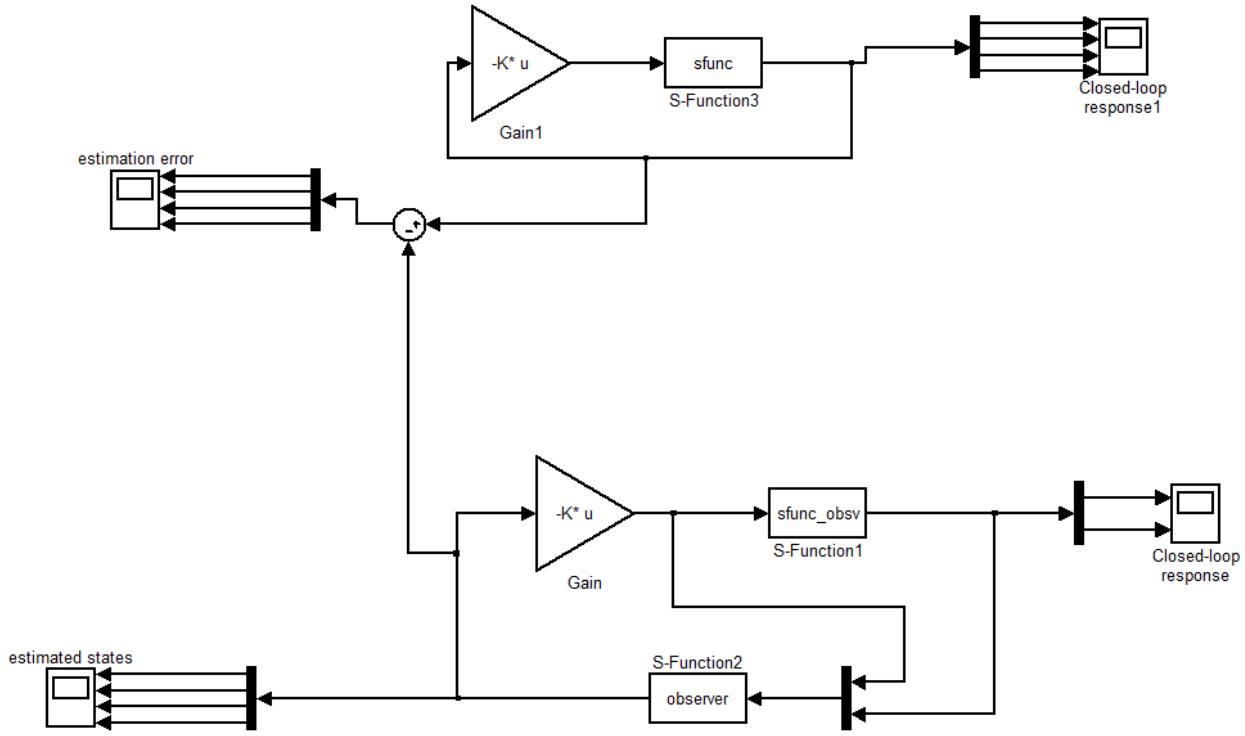


Figure 15: State-feedback Control with State Estimation for Linearized Model

The following figures show the response of the scope output on the simulation in the linearized Simulink model above.

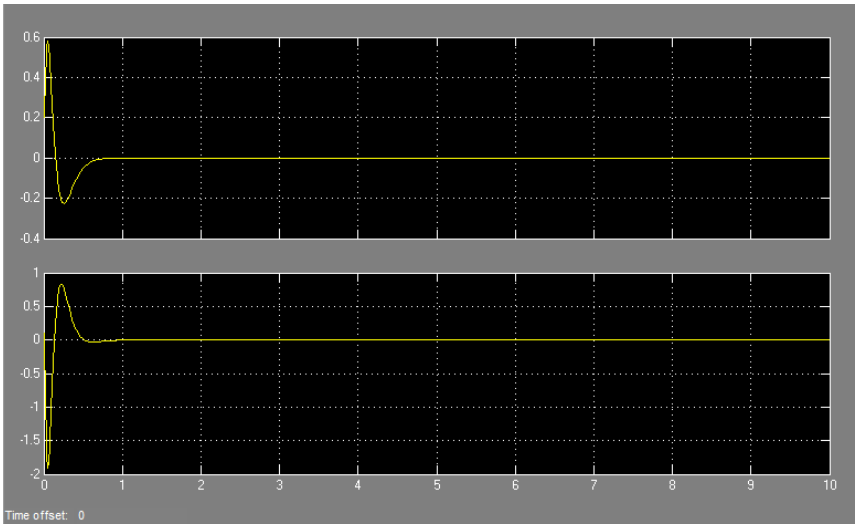


Figure 16: Closed loop response of State-feedback Control with State Estimation – first equilibrium

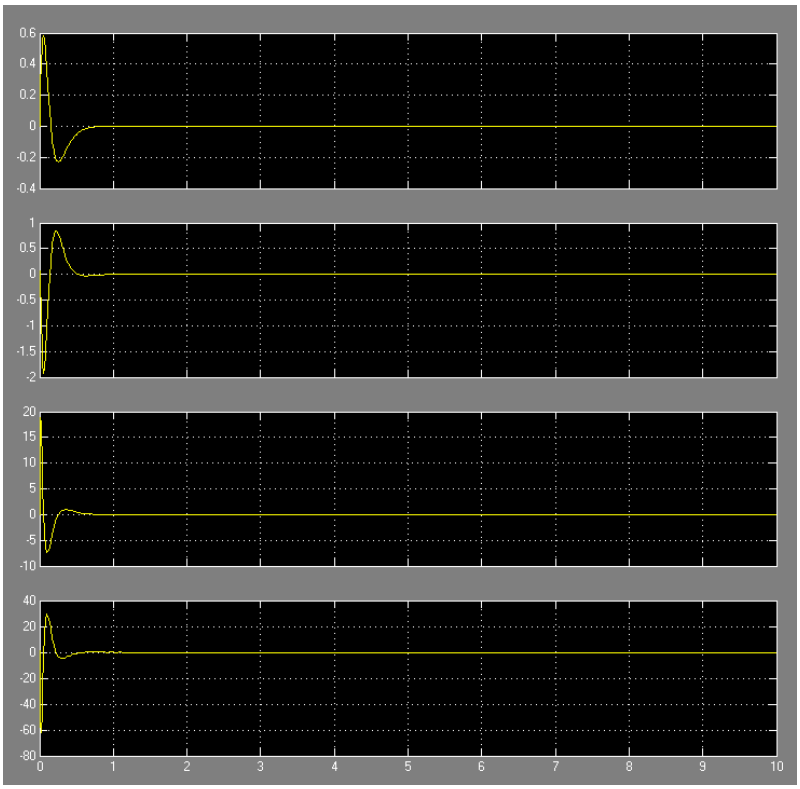


Figure 17: Estimated States of State-feedback Control with State Estimation – first equilibrium

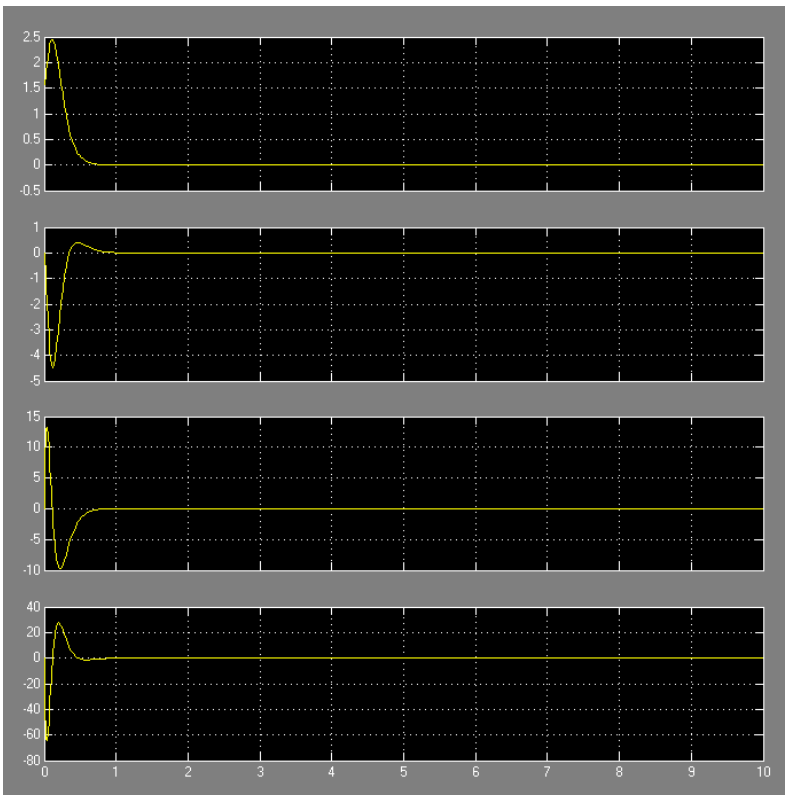


Figure 18: Closed loop response of State-feedback Control – first equilibrium

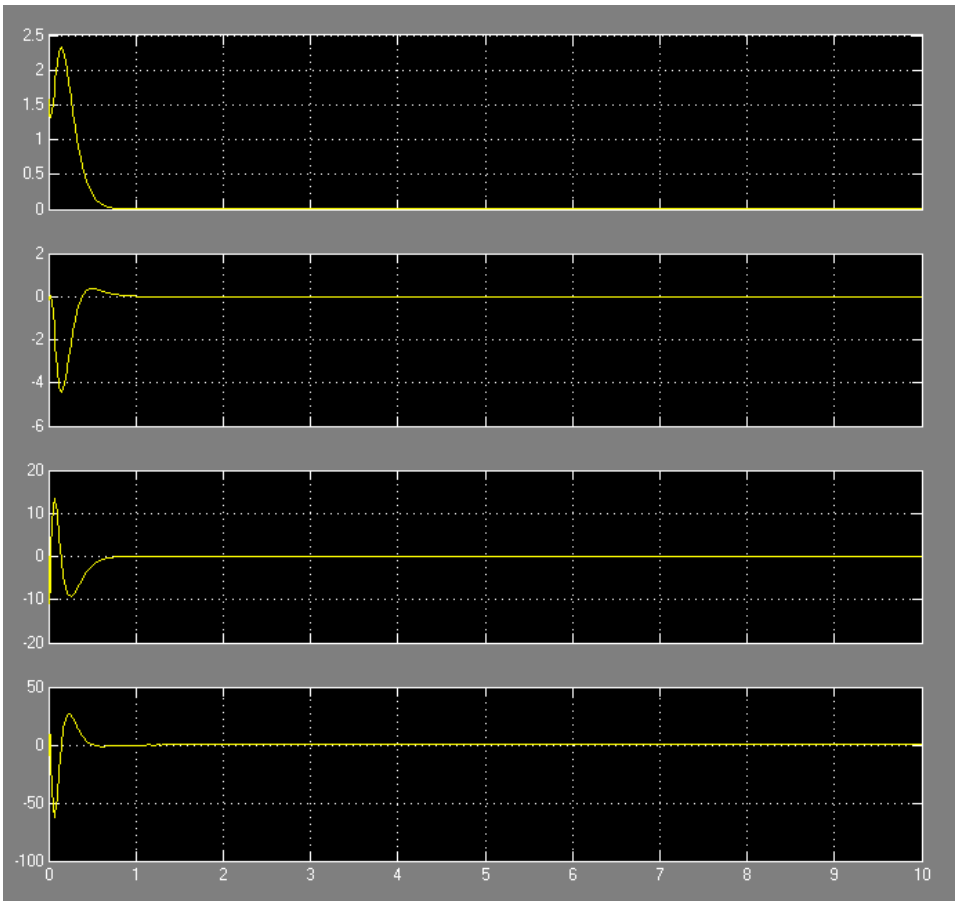


Figure 19: Estimated Error between State-feedback Control with and without State Estimation – first equilibrium

The figures above show the state-feedback control with state estimation and compare it to the results of state-feedback without an observer, assuming that all of the states can be measured. The simulations were tested for the first equilibrium point where the first joint is at pi/2 and the second joint is at zero, and the values of K are chosen by choosing the pole positions at po = [-15, -14, -11, -12], as described in the state estimator section. The plot of the closed loop response with the state estimator shows that the values for the states all return to zero after some initial perturbation. This means that the state-feedback controller is correctly changing the torque of the input to have the states return to their equilibrium position chosen. However, the initial perturbation can be quite large and the controller will still be able to have the states return to equilibrium, which is not always the case in working with the real, non-linear model as this will only work for perturbation values close to the equilibrium point.

The following figure shows the estimated states for the second equilibrium with the chosen controller K poles at po = [-10+1i, -10-1i, -15, -90]. After experimenting with several different pole placements that have complex conjugate poles, it was found that the responses had a relatively longer settling time and produced more oscillations. The response shown below does not produce a fast enough response for the second equilibrium pole so it was decided that complex conjugate poles did not produce a good enough response to test on the nonlinear system or to implement in our experiments.

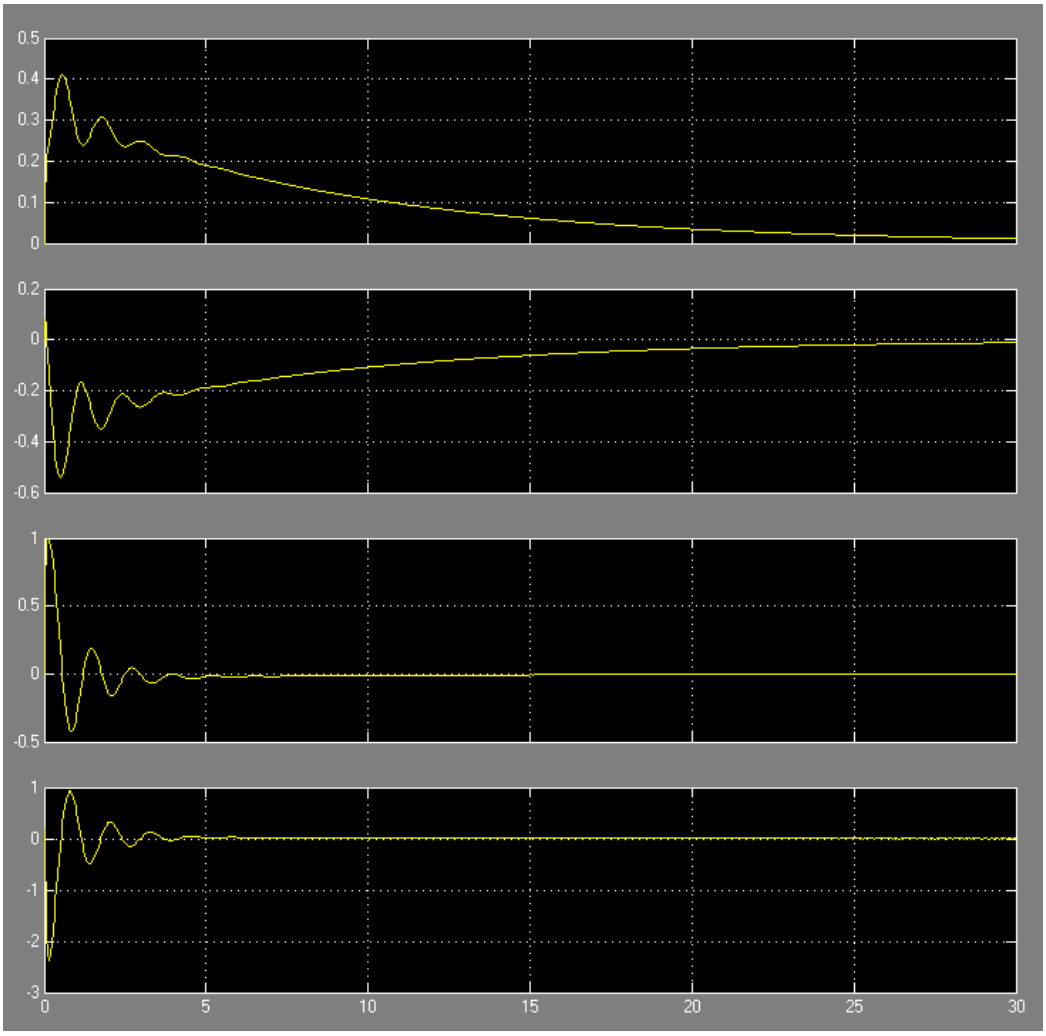


Figure 20: Estimated States of State-feedback Control with State Estimation – second equilibrium – complex conjugate pole placements

The second equilibrium was then tested with the same pole placements as the first equilibrium that did not contain complex conjugate poles and had faster poles at po = [-15, -14, -11, -12]. Similarly to the first equilibrium position, the pole placement for the observer L were chosen to be 10 times faster than that of the controller. The response of the closed loop system and estimated states for the second equilibrium position are shown below. These values produced a more desirable response and closely resemble that of the first equilibrium.

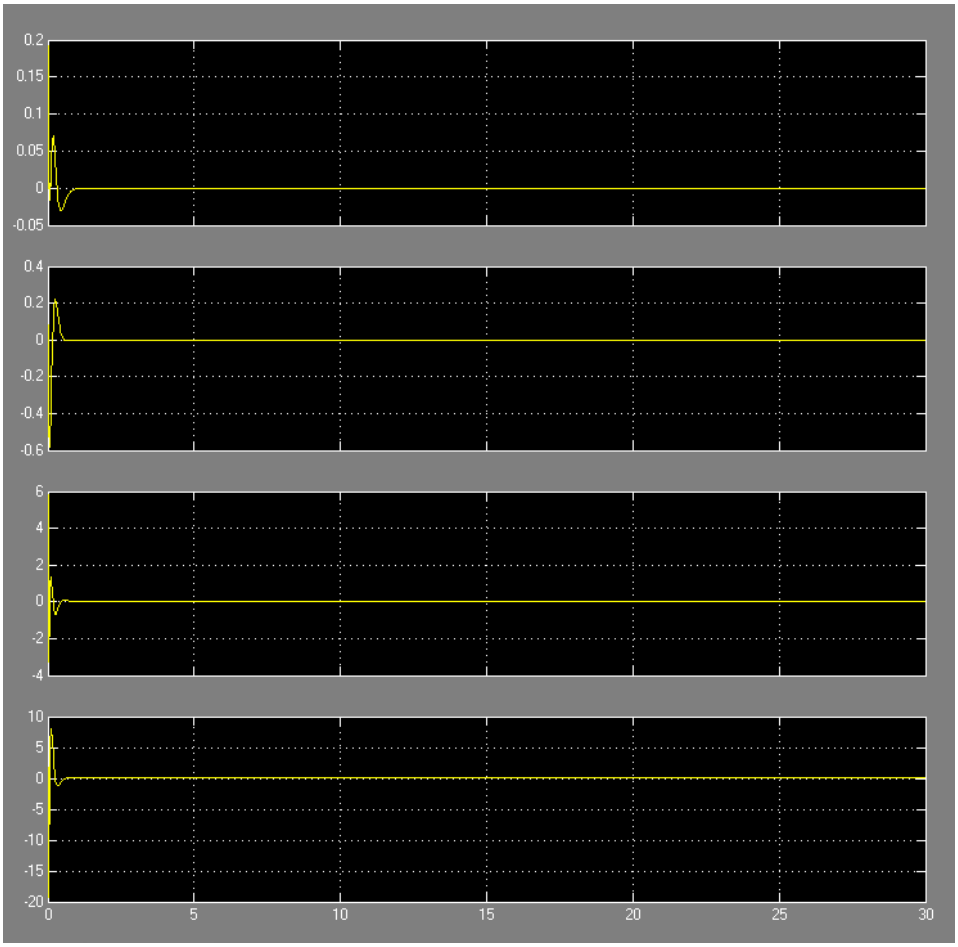


Figure 21: Estimated States of State-feedback Control with State Estimation – second equilibrium

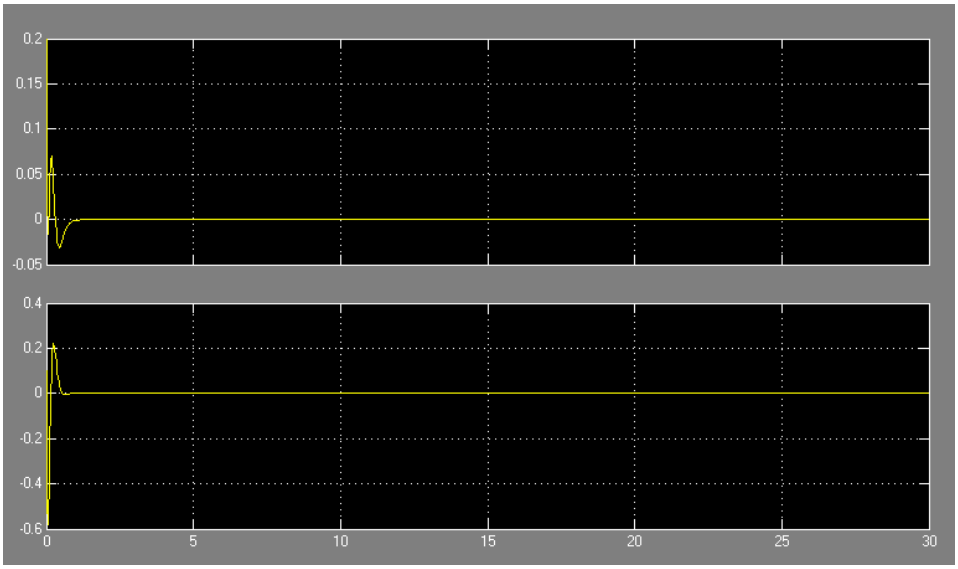


Figure 22: Closed loop response of State-feedback Control with State Estimation – second equilibrium

Non-linear Model

The controller and estimator were also tested on the nonlinear model to show how the real system may react. Although the values for K and L were designed based on the linearized system, the goal was to be able to attempt control of the nonlinear system around an equilibrium point. The Simulink model for the nonlinear model with state estimation is shown below. The nonlinear dynamics are implemented as an S-function based on second order differential equations, as was demonstrated in a previous lab phase.

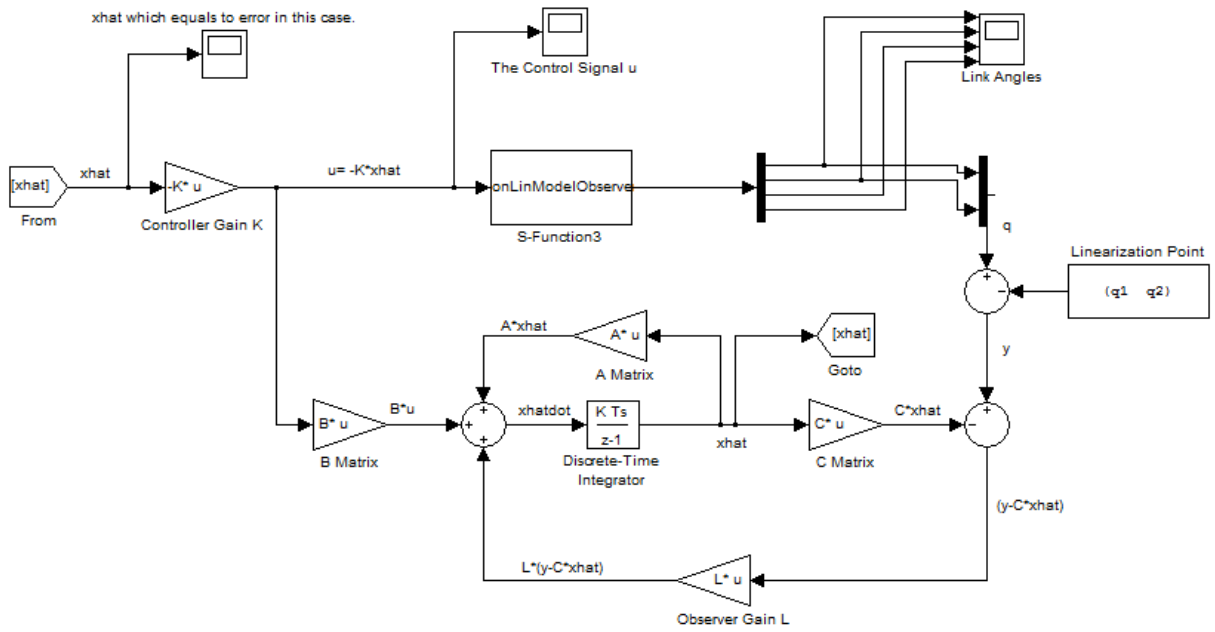


Figure 23: State-feedback Control with State Estimation for Linearized Model

The state-feedback controller here is a gain block, labelled as the Controller Gain K, and the state estimator block consists of gain blocks as well as a discrete time integrator. The observer estimates the state values from the outputs and must make estimations of the q1dot and q2dot values based on the outputs q1 and q2.

Before running the simulation, the parameters of A, B, and C, as well as the controller and observer gains K and L must be calculated. Simulations are run to test different values of K and L based on the chosen pole positions, in order to determine what values produce the best response in terms of minimizing overshoot and settling time as well as steady state errors.

The results of the simulations for the first equilibrium point are shown as follows.

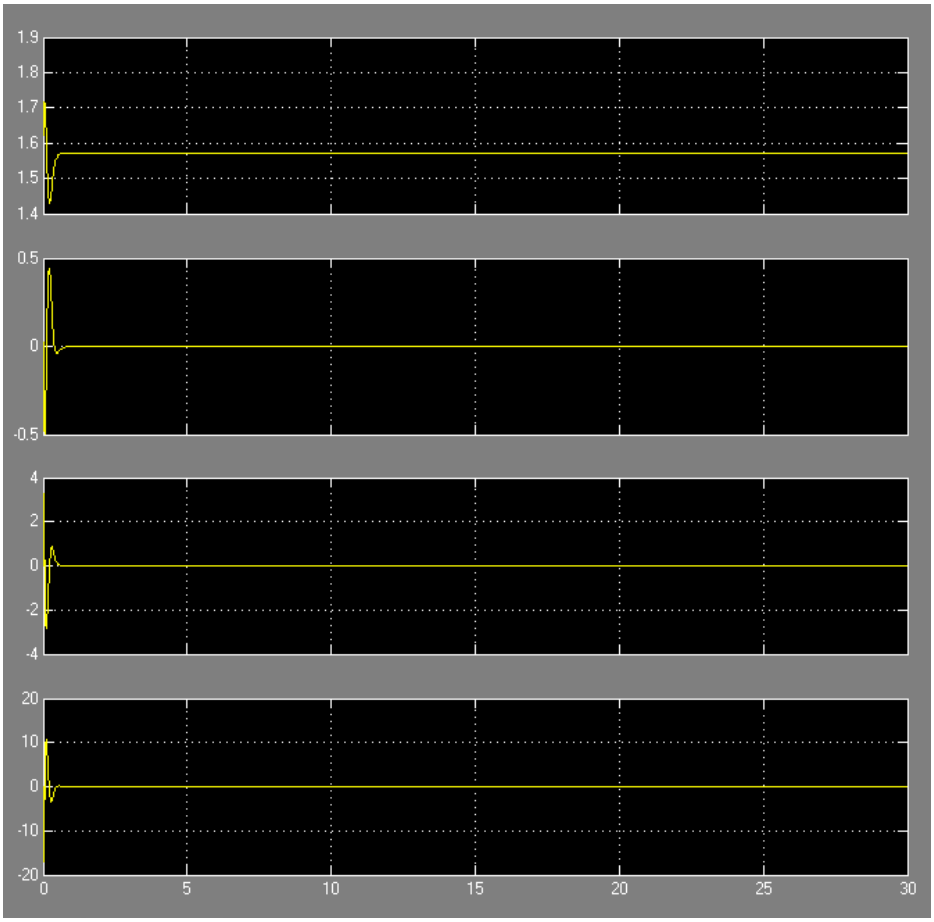


Figure 24: Closed loop response for State-feedback Control with State Estimation for Nonlinear Model – first equilibrium

The joint angles eventually return to the desired equilibrium points after a slight perturbation within 0.5 seconds and remain at the equilibrium point without steady state error. The first plot shows the value of q1 which returns to pi/2 and the second plot shows q2 which returns to its upright position at 0.

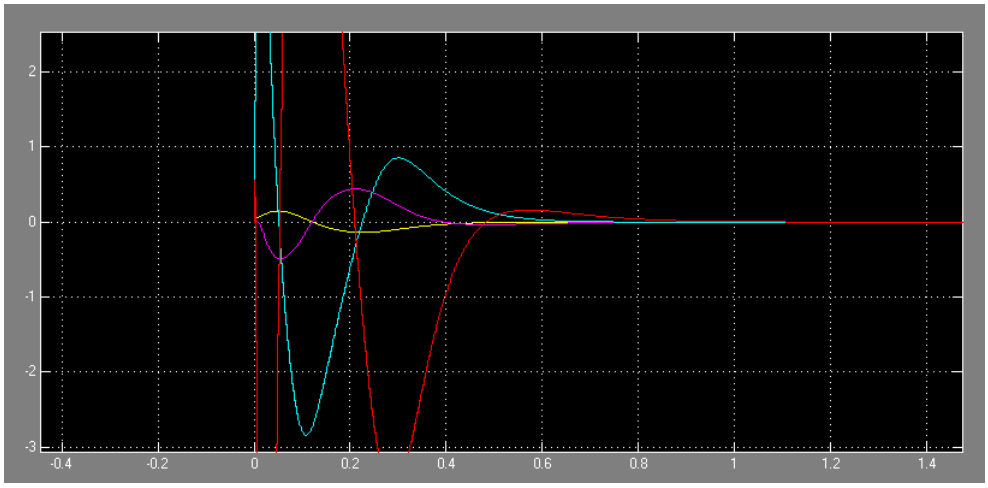


Figure 25: xhat (or error in this case)

The estimation error for each state is initially large, but as the control input is applied for compensation, the error eventually returns to zero. This shows that the system output is decaying, bounded, and the error returns to zero for all the states, meaning that the system is stable with the given input.

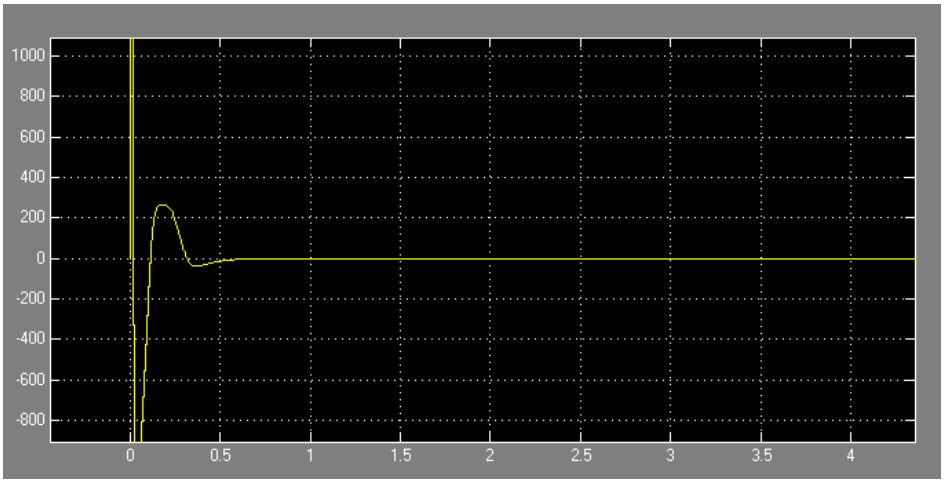


Figure 26: the control signal to u

The input is initially large because the error between the actual position and its desired location is large. The feedback controller gain determines how the system will react to the input and if it is able to reach the correct position. The system is able to reach steady state as the states reach zero and the control input also reaches zero.

The simulations showed that the nonlinear system is more sensitive to disturbances than the linearized model, as expected. The linearized model should be able to reach steady state with very large disturbances because it assumes that the system has the same characteristics for all inputs. The actual non-linear system does produce undesirable results for disturbances that are too large, as expected of the physical system.

The following figures show the nonlinear state-feedback response of the system with the second equilibrium position of q1 = pi/2 and q2 = pi. The same pole placements were used as that of the first equilibrium position as they were found to produce more desirable response when the poles are all placed on the negative real axis and had higher values (chosen within a reasonable limit). The results are shown below.

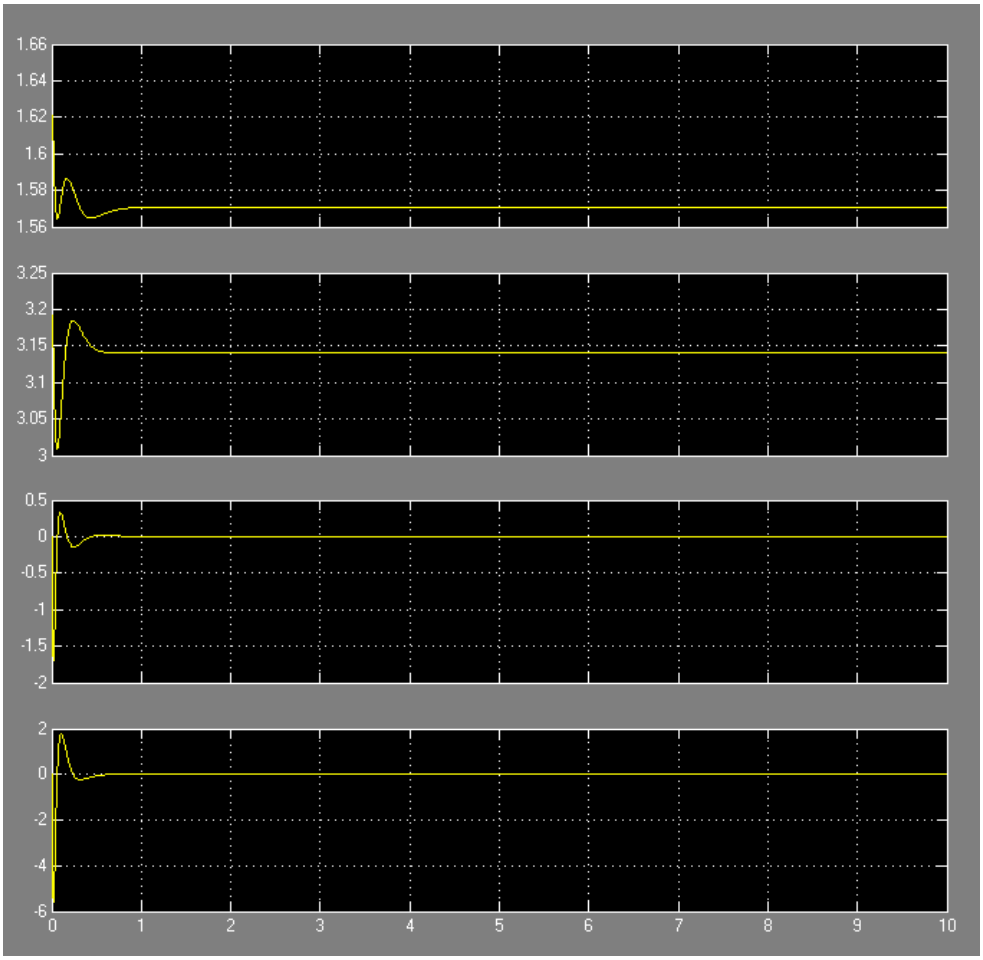


Figure 27: Closed loop response for State-feedback Control with State Estimation for Nonlinear Model – second equilibrium

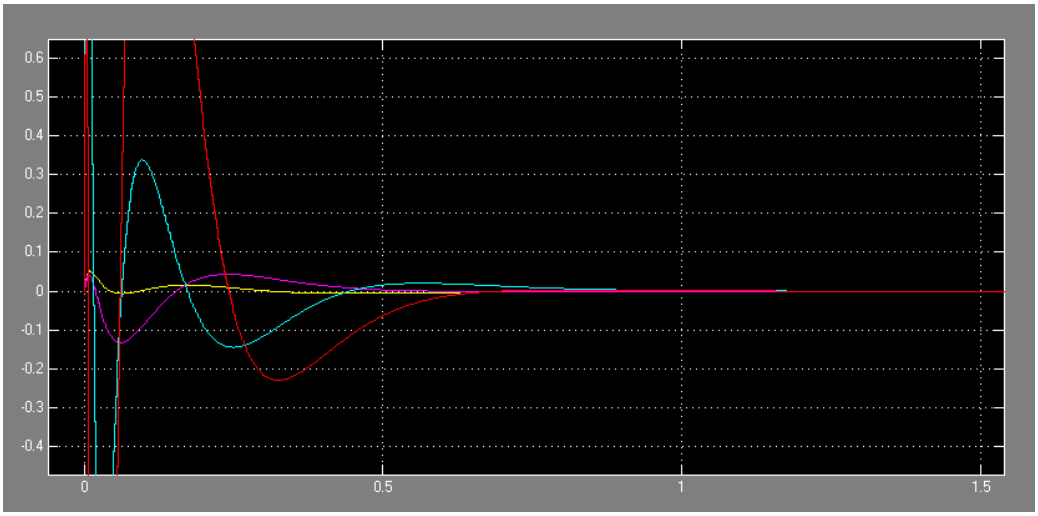


Figure 28: xhat (or error in this case)

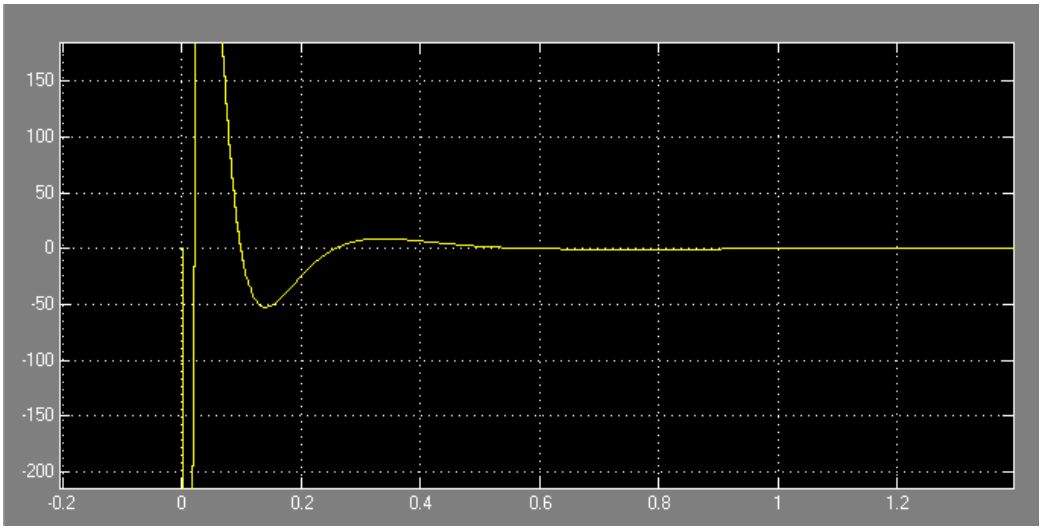


Figure 29: the control signal to u

Nonlinear Model using Numerical Differentiation

The system was also implemented using numerical differentiation instead of a state observer. This involves using a differentiator to obtain the values for the states q1 and q2 instead of designing a state estimator. The model is similar to the previous nonlinear model but with the C gain and L gain removed, as shown below.

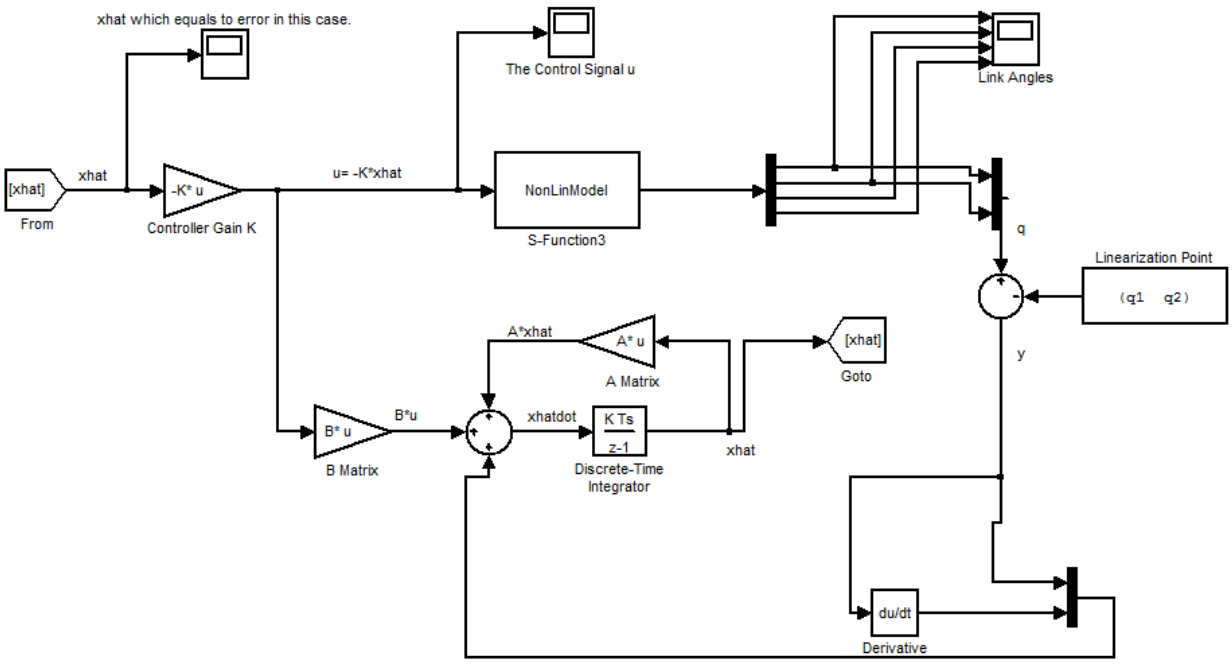


Figure 30: State-feedback controller on Nonlinear model using numerical differentiation

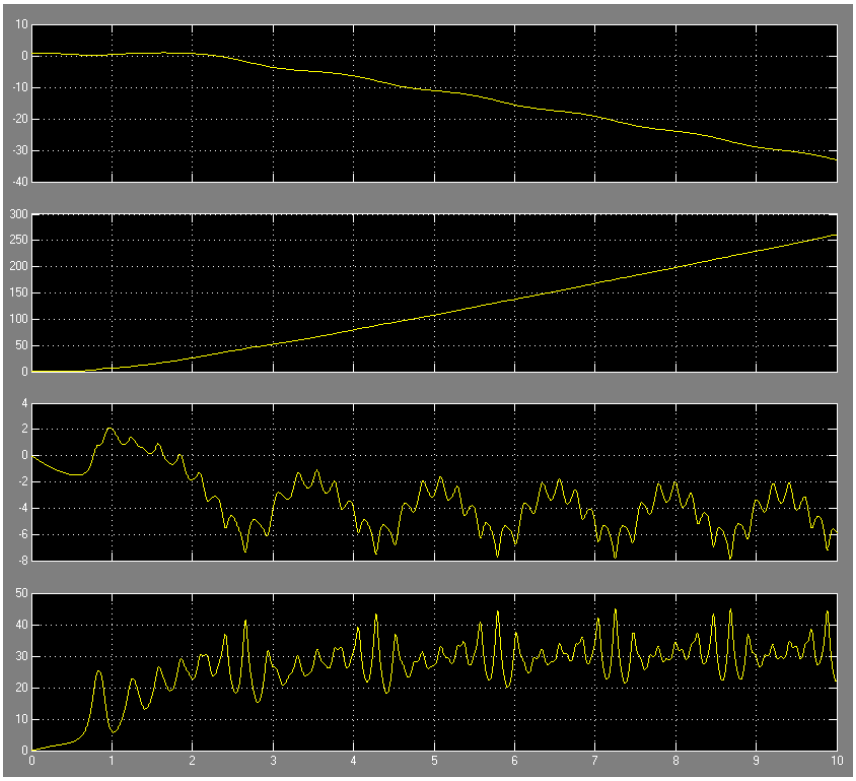


Figure 31: Closed loop response for State-feedback Control using numerical differentiation for Nonlinear Model

The results of the closed loop response of the system do not produce desirable results as the error continues to increase and never reaches a steady state. The joint positions diverge from the equilibrium and their angular velocities never reach zero and the system appears to be unstable. This shows that using a differentiator does not produce desirable results as it is very sensitive to changes in input. The differentiator is sensitive to noise and will result in the system reacting to noise and producing unbounded outputs.

Numerical differentiation was also tested without the additional discrete time integrator blocks included in the previous model. It was tested with the system shown below but this also did not produce desirable results.

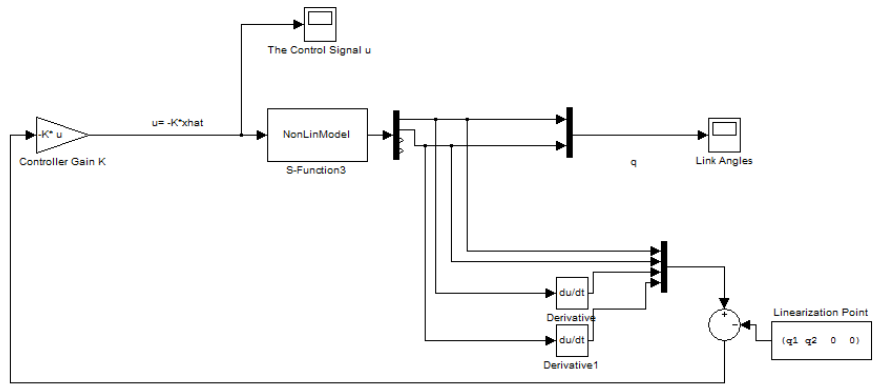


Figure 32: Second method of implementing Nonlinear numerical differentiation

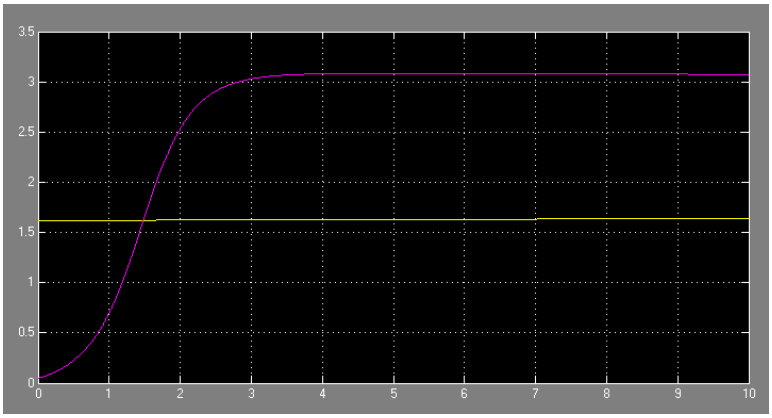


Figure 33: Closed loop response of nonlinear numerical differentiation method

These were the results for the first equilibrium position and it shows that for any perturbations it was not able to control the system as the previous nonlinear model had. It was able to maintain the position of the second equilibrium but was unable to handle any deviation from the first equilibrium. Thus the state estimator is able to produce much more desirable results.

Experiments

In the final implementation, the s-function block is replaced with a clock that controls the motor of the inverted pendulum. Once the parameters are initialized, the Simulink model is run and data of the behaviour of the actual system is obtained for the two equilibrium points.

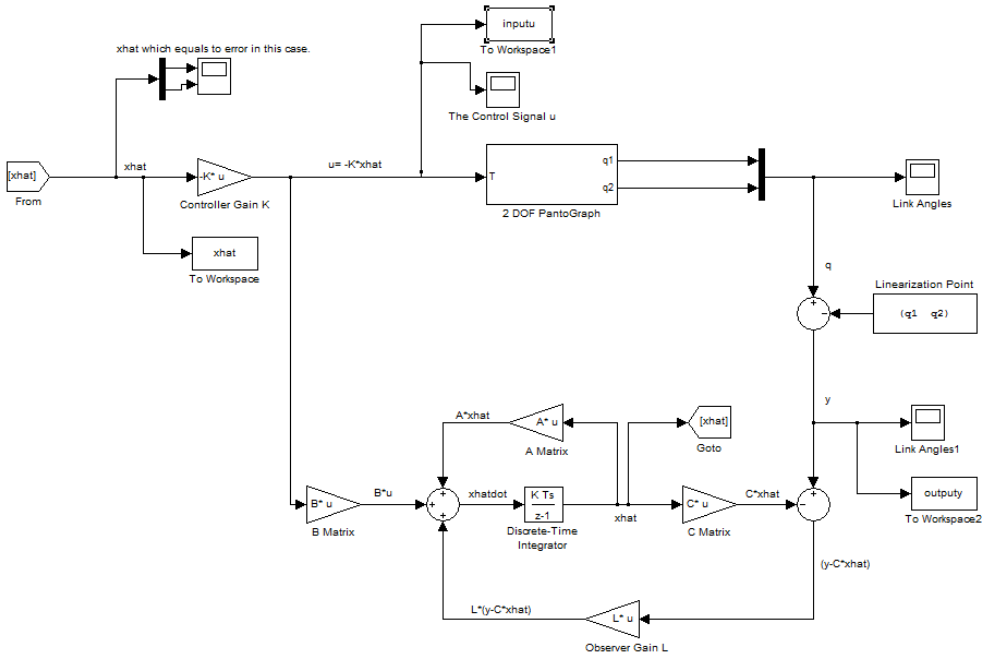


Figure 34: Final implementation model

The system was implemented using the pole positions that were on the negative, real axis only (no complex conjugate poles) since they produced better results in our simulations. The results for the actual experiment conducted for the first equilibrium position are shown below.

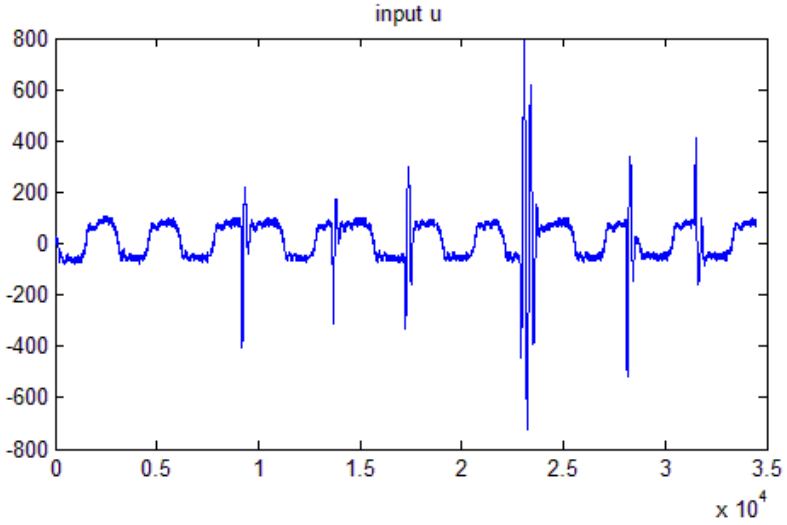


Figure 35: Control input – first equilibrium

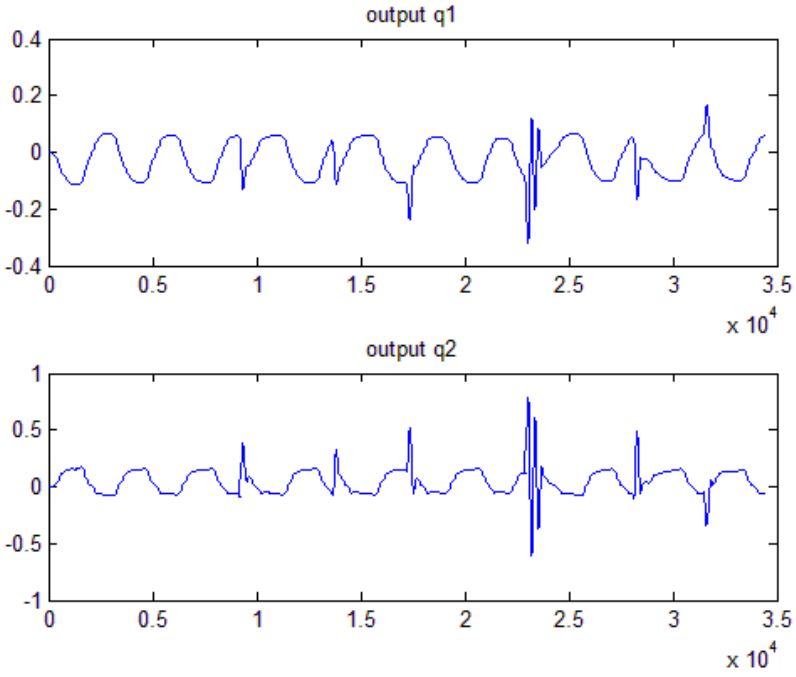


Figure 36: Output – first equilibrium

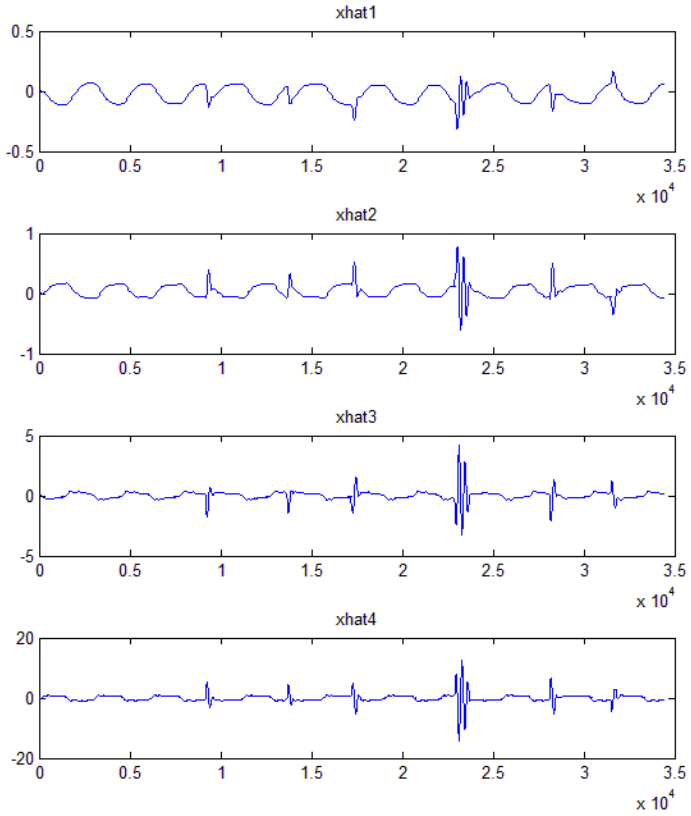


Figure 37: xhat – first equilibrium

The figures above show the response of the system to the inputs shown. The control input, output of the joint positions, and the xhat values are plotted. This clearly demonstrates that the equilibrium position of q1 at pi/2 and q2 at 0, the upright position, was maintained with the designed controller/observer.

The figures below show the results obtained from the second equilibrium position.

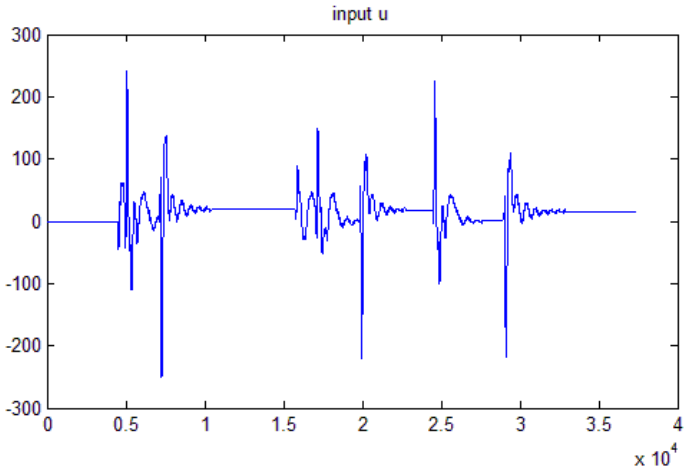


Figure 38: Control input – second equilibrium

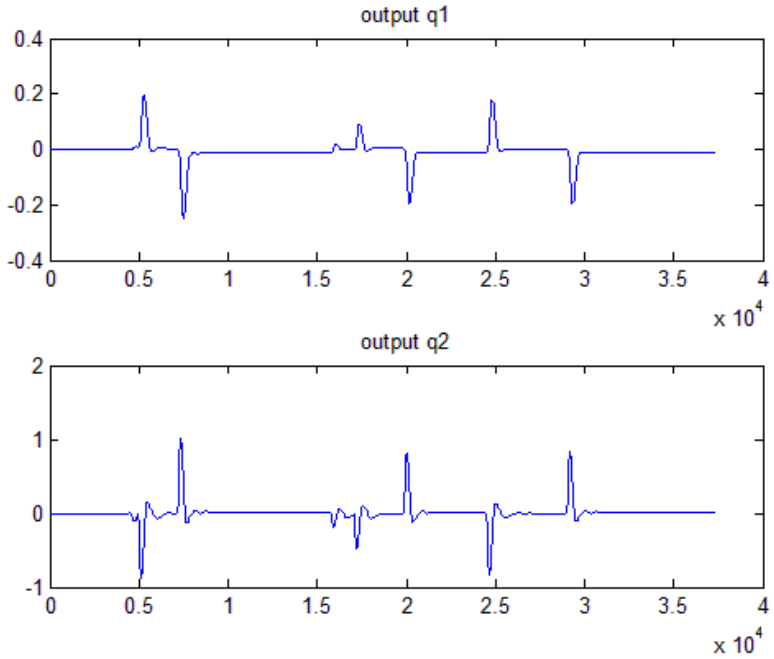


Figure 39: Output – second equilibrium

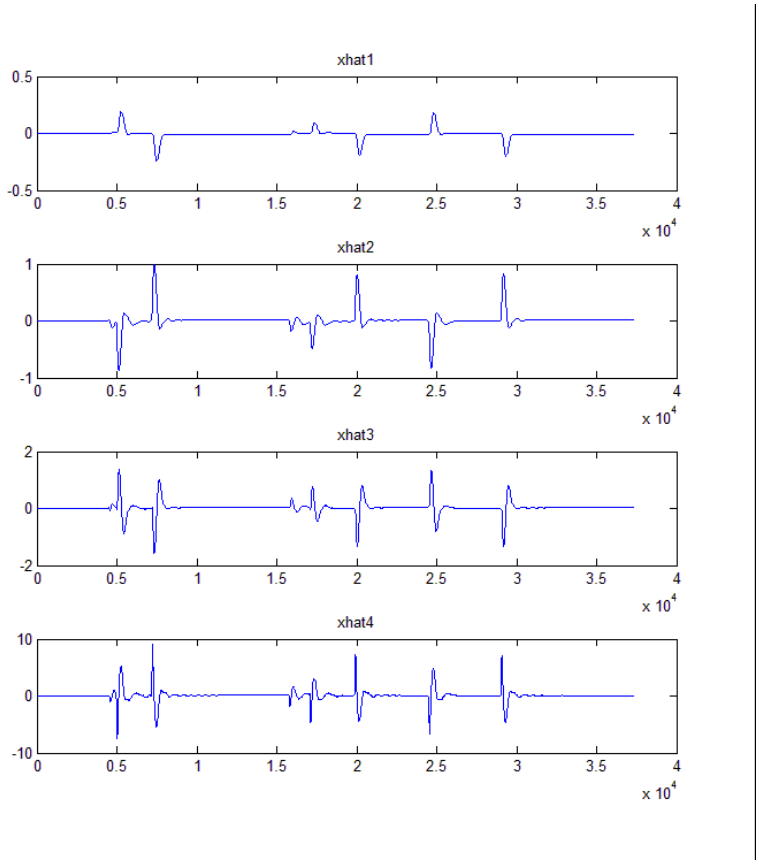


Figure 40: xhat – second equilibrium

It can be seen that the system is also able to maintain its equilibrium position at the second equilibrium. The small oscillations are due to applying an external disturbance by pushing the second link out of its operating point to test the system's ability to compensate accordingly. It was able to quickly respond to perturbations in the system and the reaction time was quick enough to maintain equilibrium and not produce unbounded results within reasonable limitations for the perturbation input.

Discussions and Conclusions

In the third and fourth phase of the project, the state-space controllers for the inverted pendulum was designed, simulated, and experimentally verified using the models obtained from the first and second phases. A linear state-feedback controller and state observer were designed based on the linearized model of the nonlinear inverted pendulum system based around two different equilibrium states. These controllers were simulated on Simulink and tested on the physical system to ensure the design produced the desired results of maintaining the equilibrium positions.

In choosing the desired poles, the pole placement plays a critical role in determining how the system will react to disturbances. If the poles are placed on the real axis, the response would appear to be critically damped. Choosing poles that are complex conjugates creates a system that appears to be underdamped, where a larger angle is able to react faster to disturbances but may result in a higher overshoot and more oscillations. The poles must lie on the negative side of the plain to ensure it is asymptotically stable and the pole placement of the controller must be within defined limits. These limits constrain the values of K from being too large so that the system does not react too forcefully with larger perturbations, since the system has limits on how much torque it can produce. Repeated poles were also avoided to ensure the stability of the system. The values of the pole placements for the observer were larger than that of the controller in order ensure that the system is able to correctly estimate the final two states quicker than it is able to apply the controller input. The fast poles ensure that the estimator error converges to zero significantly faster.

It was observed from the simulations that were conducted that the poles that were on the real axis performed better than the poles that contained complex conjugate pairs. When the imaginary component of the poles is large, the system may go into resonance and may not reach a steady state. While the complex conjugate poles may produce a transient response that reacts faster to perturbations, too much oscillations means that it may take a longer time for the oscillations to reach steady state, making the system vulnerable to changes in that period of time. The poles on the real axis were able to produce fast reaction times that were more stable and were able to produce desirable results when implemented on the physical system.

In the final implementation, the controller was tested for robustness when different external disturbances are physically applied to the system. If the controller is not robust due to improper choice of placement for the poles of the controller or observer, the system may not settle down to equilibrium or may cause the system to go beyond its operating region. However, the values for the controller and observer that were simulated where shown to be effective in controlling the physical system as well.

Contributions

Max Poddoubnyi – Abstract, Open-loop Response, State-feedback Controller, State Estimation
Sara Jamil – State-feedback Control with State Estimation, Experiments, Discussion and Conclusions