ENPM 667 Final Project Report

Rishabh Biyani
UID - 114827845

May 18, 2017

Contents

List of Figures

1 Dynamic Model

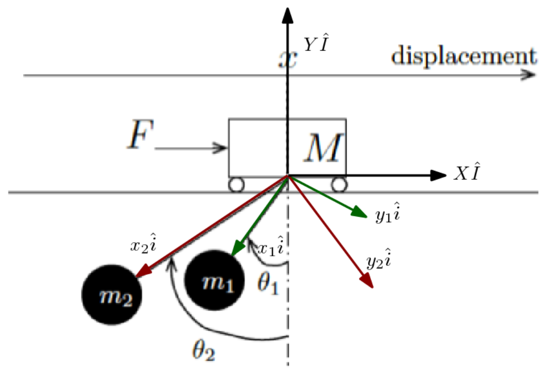


Figure 1: Diagram showing different frame assignments on the problem figure

The Euler-Lagrange equations will be used to formulate the equations of the motion. The equation for mechanical systems is written as-

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} + \frac{\partial V}{\partial q} = F \tag{1}$$

where, T and V are the kinetic and the potential energies respectively. q is a generalized co-ordinate. Here, X, θ_1 and θ_2 are the complete and independent generalized co-ordinates representing the system. The velocity of Mass M is straightforward -

$$v_M = \dot{x}\hat{I} \tag{2}$$

For the velocities of mass m_1 and m_2. Consider their respective body attached frames. The velocity of the mass m_1 can be expressed as

$$\begin{aligned} v_{m1} &= \frac{d}{dt}(l_1\hat{i}) \\ &= \left[\frac{\partial}{\partial t}(l_1)\right]_{\omega=0} + \vec{\dot{x}} + \vec{\dot{\theta_1}} \times \vec{l_1} \\ &= 0 + \vec{\dot{x}} + \dot{\theta_1}\hat{k} \times [-l_1 sin(\theta_1)\hat{I} - l_1 cos(\theta_1)\hat{J}] \\ &= [\vec{\dot{x}} - l_1\dot{\theta_1}cos(\theta_1)]\hat{I} + l_1\dot{\theta_1}sin(\theta_1)\hat{J} \end{aligned} \tag{3}$$

On similar lines the velocity of m_2 can be evaluated as -

$$v_{m2} = [\dot{x} - l_2\dot{\theta_2}cos(\theta_2)]\hat{I} + [l_2\dot{\theta_2}sin(\theta_2)]\hat{J} \tag{4}$$

Now, the total Potential and kinetic Energy of the system can be written as

$$V = m_1 l_1 g\,(1 - \cos(\theta_1(t))) + m_2 l_2 g\,(1 - \cos(\theta_2(t))) \tag{5}$$

$$\begin{aligned} T &= \frac{1}{2}m_1 v_{m1}^2 + \frac{1}{2}m_2 v_{m2}^2 + \frac{1}{2}M\dot{x}^2 \\ &= \frac{1}{2}m_1\left((l_1\theta_1'(t)\sin(\theta_1(t)))^2 + (x'(t) - l_1\theta_1'(t)\cos(\theta_1(t)))^2\right) \\ &+ \frac{1}{2}m_2\left((l_2\theta_2'(t)\sin(\theta_2(t)))^2 + (x'(t) - l_2\theta_2'(t)\cos(\theta_2(t)))^2\right) + \frac{1}{2}Mx'(t)^2 \end{aligned} \tag{6}$$

Now, equation 1 will be evaluated for each generalized co-ordinates

- For the generalized co-ordinate $\mathbf{x}$

$$
\left.
\begin{aligned}
F &= \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x}}\right) - \frac{\partial T}{\partial x} + \frac{\partial V}{\partial x} \\
&= m_1 l_1 \left(\theta_1'(t)^2 \sin\left(\theta_1(t)\right) - \theta_1''(t)\cos\left(\theta_1(t)\right)\right) + m_2 l_2 \left(\theta_2'(t)^2 \sin\left(\theta_2(t)\right) - \theta_2''(t)\cos\left(\theta_2(t)\right)\right) \\
&\quad + \left(m_1 + m_2 + M\right) x''(t) \\
\Rightarrow x''(t) &= \frac{F + l_1 m_1 \left(\theta_1''(t)\cos\left(\theta_1(t)\right) - \theta_1'(t)^2 \sin\left(\theta_1(t)\right)\right) + l_2 m_2 \left(\theta_2''(t)\cos\left(\theta_2(t)\right) - \theta_2'(t)^2 \sin\left(\theta_2(t)\right)\right)}{m_1 + m_2 + M}
\end{aligned}
\right\}
\tag{7}
$$

  with, $\frac{\partial T}{\partial x} = \frac{\partial V}{\partial x} = 0$

- For the generalized co-ordinate $\boldsymbol{\theta_1}$

$$
\begin{aligned}
0 &= \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta_1}}\right) - \frac{\partial T}{\partial \theta_1} + \frac{\partial V}{\partial \theta_1} \\
&= m_1 l_1 \left(g \sin\left(\theta_1(t)\right) - \cos\left(\theta_1(t)\right) x''(t) + l_1 \theta_1''(t)\right) \\
&= \left(g \sin\left(\theta_1(t)\right) - \cos\left(\theta_1(t)\right) x''(t) + l_1 \theta_1''(t)\right) \\
\Rightarrow \theta_1''(t) &= \frac{\cos\left(\theta_1(t)\right) x''(t) - g \sin\left(\theta_1(t)\right)}{l_1}
\end{aligned}
\tag{8}
$$

- Similarly, for generalized co-ordinate $\boldsymbol{\theta_2}$

$$
\begin{aligned}
0 &= \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta_2}}\right) - \frac{\partial T}{\partial \theta_2} + \frac{\partial V}{\partial \theta_2} \\
&= m_2 l_2 \left(g \sin\left(\theta_2(t)\right) - \cos\left(\theta_2(t)\right) x''(t) + l_2 \theta_2''(t)\right) \\
\Rightarrow \theta_2''(t) &= \frac{\cos\left(\theta_2(t)\right) x''(t) - g \sin\left(\theta_2(t)\right)}{l_2}
\end{aligned}
\tag{9}
$$

Thus, substituting (9) and (8) in (7) we have

$$
x''(t) = \frac{F - m_1 g S\left(\theta_1(t)\right) C\left(\theta_1(t)\right) - m_2 g S\left(\theta_2(t)\right) C\left(\theta_2(t)\right) - m_1 l_1 \theta_1'(t)^2 S\left(\theta_1(t)\right) - m_2 l_2 \theta_2'(t)^2 S\left(\theta_2(t)\right)}{m_1 S^2\left(\theta_1(t)\right) + m_2 S^2\left(\theta_2(t)\right) + M}
\tag{10}
$$

where, sin $\Rightarrow S$ and cos $\Rightarrow C$. Hence, equations (8) (9) and (10) collectively represent the dynamics of the given system

1.1 State Space Form

A. For the current system the state vector will be a 6 $\times$ 1 vector. A will be a 6 $\times$ 6 matrix, B will be a 6 $\times$ 1 matrix and U is a 1 $\times$ 1 scalar.This looks like as follows:-

$$
\left\{
\begin{aligned}
x_1 &= x \\
x_2 &= \dot{x_1} = \dot{x} \\
x_3 &= \theta_1 \\
x_4 &= \dot{x_2} = \dot{\theta}_1 \\
x_5 &= \theta_2 \\
x_6 &= \dot{x_5} = \dot{\theta}_2
\end{aligned}
\right.
\tag{11}
$$

Let, $X = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6]^T$. Thus, the non-linear state-space equation can be expressed as

$$\dot{X} = f(X,U) = \begin{pmatrix} x'(t) \\ K \\ \theta_1'(t) \\ \frac{\cos(\theta_1(t))K - g\sin(\theta_1(t))}{l_1} \\ \theta_2'(t) \\ \frac{\cos(\theta_2(t))K - g\sin(\theta_2(t))}{l_2} \end{pmatrix} \tag{12}$$

where $K = X''(t)$ as expressed in equation 10.

B. Now, the above non-linear equations can be linearized around the equilibrium point $x = 0$ $\theta_1 = 0$ $\theta_2 = 0$ using the first order terms of the Taylor expansion. At the equilibrium position the state vector $X = X_0 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$ and the control input $U = U_0 = 0$. The expansion is written as

$$\dot{X} = f(X_0, U_0) + \delta X \times \left[\frac{\partial f}{\partial X}\right]_{X=X_0, U=U_0} + \delta U \times \left[\frac{\partial f}{\partial U}\right]_{X=X_0, U=U_0} \tag{13}$$

where, $f(X_0, U_0) = 0$, since this is an equilibrium point. The $\frac{\partial f}{\partial X}$ and $\frac{\partial f}{\partial U}$ are the constant A and B matrices respectively of the linearized system. These derivatives are infact Jacobians because the f is a multi-valued function in X. The calculation was carried out using Mathematica and evaluated at X_0 and U_0. The script required in the calculation is located at Appendix B.8. Only the final result after substitution will be presented here. Thus,

$$\left[\frac{\partial f}{\partial X}\right]_{X=X_0, U=U_0} = A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{gm_1}{M} & 0 & -\frac{gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{g(M+m_1)}{Ml_1} & 0 & -\frac{gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{gm_1}{Ml_2} & 0 & -\frac{g(M+m_2)}{Ml_2} & 0 \end{pmatrix} \tag{14}$$

$$\left[\frac{\partial f}{\partial U}\right]_{X=X_0, U=U_0} = B = \begin{pmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{pmatrix} \tag{15}$$

And the single control input $U = F$. This equations come out to be the same even if we use the small angle approximation where, $\sin(\theta) \approx \theta$ and $\cos(\theta) \approx 1$. Thus, the linearized state space equations can be written as

$$\begin{cases} \dot{X} = AX + BU \\ \dot{X} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{gm_1}{M} & 0 & -\frac{gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{g(M+m_1)}{Ml_1} & 0 & -\frac{gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{gm_1}{Ml_2} & 0 & -\frac{g(M+m_2)}{Ml_2} & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{pmatrix} U \end{cases} \tag{16}$$

If all the three state variable viz. X, θ_1 and θ_2 are measured the Output Vector Y will be a 3×1 vector and it will be of the form

$$\begin{aligned} Y &= CX \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} \end{aligned} \tag{17}$$

2 Controls

C. Conditions for **Controllability** of the linearized system can be calculated now. The controllability matrix for this system is huge 6×6 matrix some components are written as:-

$$\begin{aligned} C &= [B \quad AB \quad A^2B \quad A^3B \quad A^4B \quad A^5B] \\ &= \begin{pmatrix} 0 & \frac{1}{M} & 0 & -\frac{g(l_2m_1+l_1m_2)}{l_1l_2M^2} & \dots\dots\dots \\ \frac{1}{M} & 0 & -\frac{g(l_2m_1+l_1m_2)}{l_1l_2M^2} & 0 & \dots\dots\dots \\ 0 & \frac{1}{Ml_1} & 0 & -\frac{g(l_2M+l_2m_1+l_1m_2)}{l_1^2l_2M^2} & \dots\dots\dots \\ \frac{1}{Ml_1} & 0 & -\frac{g(l_2M+l_2m_1+l_1m_2)}{l_1^2l_2M^2} & 0 & \dots\dots\dots \\ 0 & \frac{1}{Ml_2} & 0 & -\frac{g(l_1M+l_2m_1+l_1m_2)}{l_1l_2^2M^2} & \dots\dots\dots \\ \frac{1}{Ml_2} & 0 & -\frac{g(l_1M+l_2m_1+l_1m_2)}{l_1l_2^2M^2} & 0 & \dots\dots\dots \end{pmatrix} \end{aligned}$$

The system will be controllable if it has 6 linearly dependent rows or columns or in other words the Matrix is of full rank and is invertible with determinant as non-zero. Hence, evaluating the determinant of this Matrix. We have,

$$\begin{aligned} Det(C) &= -(g^6l_1^2 - 2g^6l_1l_2 + g^6l_2^2)/(M^6l_1^6l_2^6) \\ &= -(g^6(l_1 - l_2)^2)/(Ml_1l_2)^6 \end{aligned}$$

Determinant is independent of the masses m_1 and m_2. This shows that determinant goes to zero either when $g = 0$ or more practically $l_1 = l_2$. Thus, the system is controllable in all situations except when $\mathbf{l_1 = l_2}$

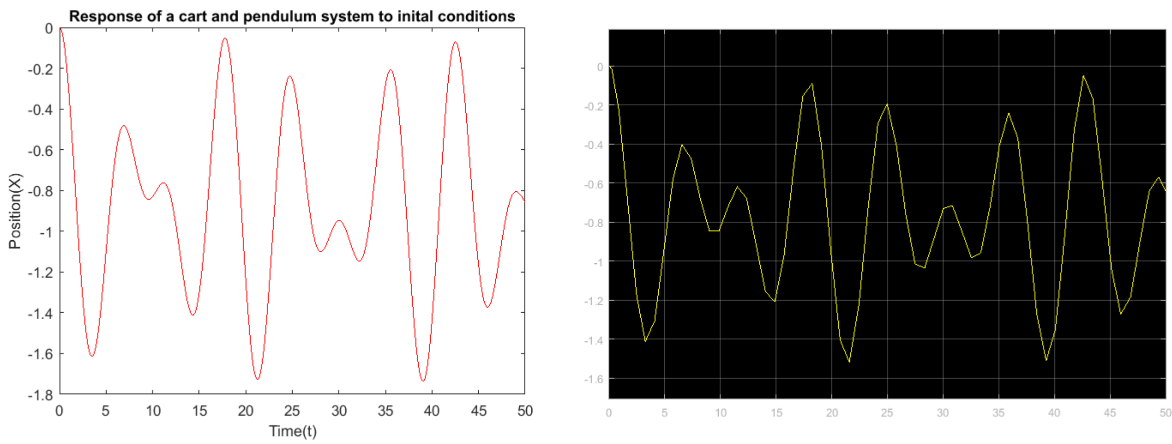
D. Substituting $M = 1000$, $m_1 = m_2 = 100$ and $l_1 = 20, l_2 = 20$ and $g = 9.81$ in the controllability Matrix. We have,

$$C_1 = 0.001 \times \begin{pmatrix} 0 & 1 & 0 & -0.1472 & 0 & 0.1419 \\ 1 & 0 & -0.1472 & 0 & 0.1419 & 0 \\ 0 & 0.05 & 0 & -0.0319 & 0 & 0.0227 \\ 0.05 & 0 & -0.0319 & 0 & 0.0227 & 0 \\ 0 & 0.1 & 0 & -0.1128 & 0 & 0.1249 \\ 0.1 & 0 & -0.1128 & 0 & 0.1249 & 0 \end{pmatrix}$$

In MATLAB, $rank(C_1) = 6$. This shows that controllability matrix is full rank hence the system is controllable.

2.1 LQR controller

LQR is an optimal control strategy wherein we minimize a quadratic cost function based on our weights on the errors in state and the required control effort. LQR control is a type of **full-state feedback** controller quite similar to the pole-placement technique, just that it is optimal. What full-state feedback



(a) Linearized System.

(b) Non-linear System.

Figure 2: Open-Loop Response of the Position of the Cart

means is that we feed the entire state vector to achieve the desired response. This technique is typically applied in systems involving multiple outputs and we desire to exercise control on all these outputs such as the current system which has three outputs. We will assume in the formulation of this controller that the state vector is known completely and we are able to measure accurately the position(x), angles of the pendulum (θ_1) and (θ_2) respectively. The LQR cost function λ and the corresponding gain K are given as:

$$\begin{cases} \lambda = \int_0^\infty (x^T Q x + u^T R u) dt \\ K = -R^{-1} B^T P \end{cases} \tag{18}$$

where, P is the solution to the Riccati equation

$$A^T P + P^T A - PBR^{-1}B^T P = 0$$

the MATLAB function **lqr** essentially encapsulates all these calculations. It calls the **care** method internally which solves the Riccati equation numerically. So, the task is to choose the Positive definite weighting Matrices Q and R which are the arguments to **lqr**. With Full-state Feedback, the control input becomes $u = -KX$. Thus, the closed-loop state space form becomes

$$\begin{aligned} \dot{X} &= AX + B(-KX) \\ &= (A - BK)X \end{aligned} \tag{19}$$

Thus, $A_c = A - BK$, $B_c = 0$

2.1.1 Simulations

Throughout this project, simulation on a linear system was carried out using a MATLAB script. Whereas, non-linear simulations of the system are performed in Simulink. The simulink diagram of the non-linear system is given in the Section B.1. Before implementing the controller, the system was simulated in Open Loop(both the linearized and non-linear system) to check the response under given initial conditions. The code to simulate in Open Loop is located in the section A.2. Initial conditions chosen are : $\mathbf{X_0} = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{15} \quad \mathbf{0} \quad \mathbf{20} \quad \mathbf{0}]$. Where, the angles are in degrees. Care has been taken to chose the angles close to the equilibrium position and it should not exceed more than 25 degrees. Figure 2a and 2b shows the response of the position of the cart(x) from these initial conditions. Clearly, a controller will be useful to get a desired response where the position of the cart and the two pendulums stabilizes. For LQR controller as mentioned in the last section we need to choose the Q and the R

matrices. Usually, a good guideline is to choose $R = 1$ and $Q = C^T C$ [1] and then later adjust the non-zero values of the Q to get a desired response. This technique works because we are only interested in the relative values of Q and R. Thus, after some iterations following weights were chosen.

$$Q(1,1) = 90000000$$
$$Q(3,3) = 80000000000$$
$$Q(5,5) = 70000000000;$$

And the resulting gains are $K = 100000 \times [0.0949 \quad 0.2879 \quad 2.7949 \quad 0.5221 \quad 1.3727 \quad -2.2776]$. The MATLAB script for calculations of lqr on a linear system is located at A.3 and with the non-linear Simulink model in B.2.

Comments: The simulations plots observed on both the linear and non-linear system are located in the figure 3. The response of the system is very good considering that the position of the cart stabilizes again within 15 seconds and the oscillations of the pendulum get contained below 2 degrees within 10 seconds. Also, the control input is at the max of the magnitude 120000 N which is not unreasonable considering that M is about 1000 Kg. The control input might seem to be zero after first few seconds but it is around 300 N when zoomed in those regions. Because of the overall range of values these seem to look negligible.

The Eigen values of the closed loop can be calculated from Eig(($\mathbf{A} - \mathbf{BK}$)) and are as follows:

$$-3.9073 + 4.0258i$$
$$-3.9073 - 4.0258i$$
$$-0.2712 + 0.3949i$$
$$-0.2712 - 0.3949i$$
$$-0.1336 + 0.7837i$$
$$-0.1336 - 0.7837i$$

Since, all the eigen values of the closed loop linearized system have a negative real part **Lyapunov Indirect method** guarantees local stability.

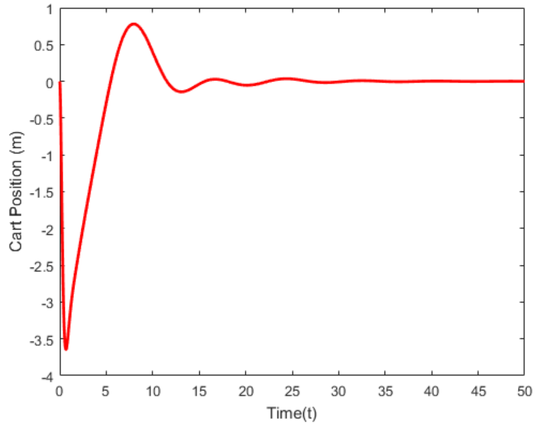
2.2 State Estimation

The analysis in the first part of the project was based on the assumption that all the state variables are known accurately. However, this may not be the case always and it depends on the chosen output vector whether we will be able to recover the state vector completely. There are two sub-categories in State Estimation. They can be described briefly as follows:

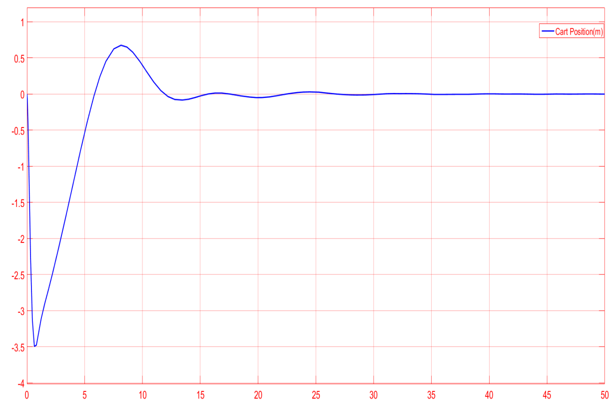
- **State Observer**: This is the case where we assume that that our sensor model is perfect and the chosen output vector represent the actual state faithfully. This is where we talk about the **Luenberger Observer** as we will see in the section E and F of the problem.
- **Kalman Filtering**: Here we get a bit more practical and say that our sensor models are not perfect and we model the sensor noise using an independent and identically distributed Gaussian distribution(White noise). This aspect is dealt with in some detail when a **LQG** based controller is designed in Section G.

2.2.1 Observability

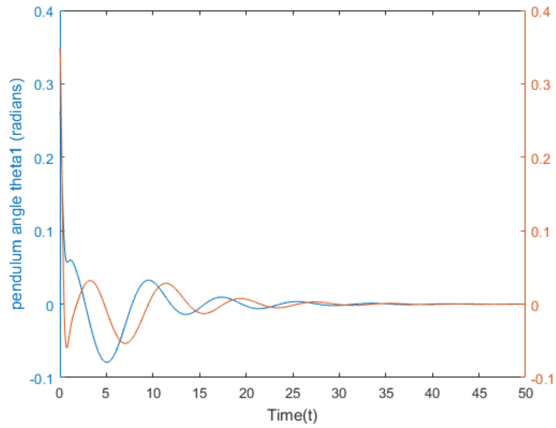
Observability is an important notion for both the state estimation cases described in the last section. For the case of 'State Observer' it gives a confirmation whether the chosen output vector will be able to reproduce the state vector. For the other case of Kalman Filtering, if the observability criterion is satisfied it can be shown that the P Matrix converges to a constant value and can be solved from the Algebraic Riccati equation, essentially giving us a constant value of Kalman gain for a LTI system.



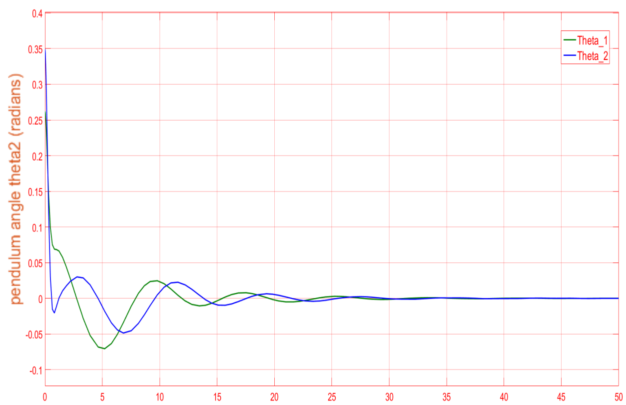
(a) Linearized System.



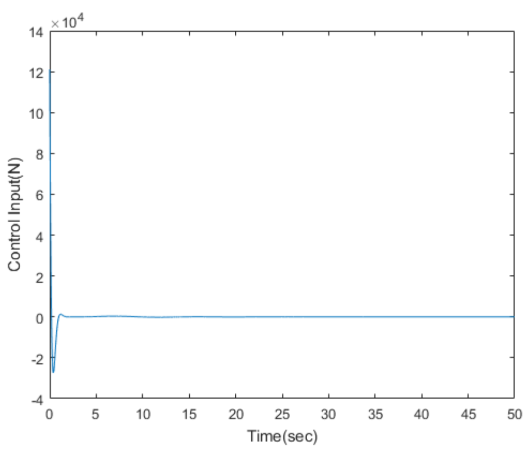
(b) Non-linear System.



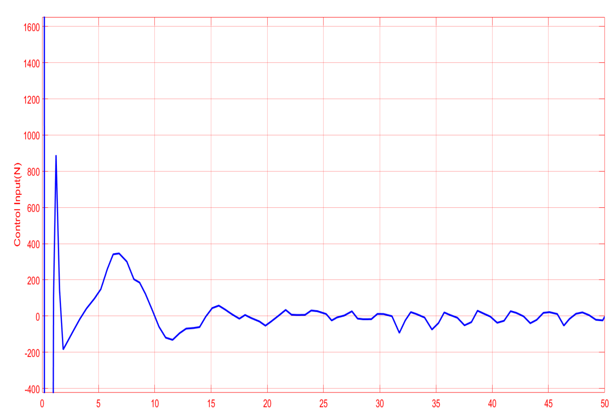
(c) Linearized System.



(d) Non-linear System.



(e) Linearized System.



(f) Non-linear System.

Figure 3: Closed-Loop Response of the system with a LQR controller to non-zero initial conditions

E. A system is said to be Observable if the Observability Matrix is full-rank. For this system the Observability Matrix is given by:

$$O = \begin{pmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \end{pmatrix} \tag{20}$$

Thus, this matrix will be evaluated for each given choice of output vectors.

Choice 1: $\mathbf{y(t)} = \mathbf{x(t)}$. Thus, $C = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$. And the Observability Matrix is.

$$O_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.981 & 0 & -0.981 & 0 \\ 0 & 0 & 0 & -0.9810 & 0 & -0.9810 \\ 0 & 0 & 0.6255 & 0 & 1.1067 & 0 \\ 0 & 0 & 0 & 0.6255 & 0 & 1.1067 \end{pmatrix}$$

Since, $rank(O_1) = 6$. The system is observable for this choice of the output vector.

Choice 2: $\mathbf{y(t)} = (\boldsymbol{\theta_1}\mathbf{(t)}, \boldsymbol{\theta_2}\mathbf{(t)})$. Thus,

$$C = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$O_2 = \begin{pmatrix} 0 & 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \\ 0 & 0 & -0.5395 & 0 & -0.0491 & 0 \\ 0 & 0 & -0.0981 & 0 & -1.0791 & 0 \\ 0 & 0 & 0 & -0.5395 & 0 & -0.0491 \\ 0 & 0 & 0 & -0.0981 & 0 & -1.0791 \\ 0 & 0 & 0.2959 & 0 & 0.0794 & 0 \\ 0 & 0 & 0.1588 & 0 & 1.1693 & 0 \\ 0 & 0 & 0 & 0.2959 & 0 & 0.0794 \\ 0 & 0 & 0 & 0.1588 & 0 & 1.1693 \end{pmatrix}$$

Since, $rank(O_2) = 4 < 6$ and clearly there are two columns with zero elements. The matrix is not full rank and the system is not observable for this choice of output vector

Choice 3: $\mathbf{y(t)} = (\mathbf{x(t)}, \boldsymbol{\theta_2}\mathbf{(t)})$. Thus, we have

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$O_3 = \begin{pmatrix} 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \\ 0 & 0 & -0.9810 & 0 & -0.9810 & 0 \\ 0 & 0 & -0.0981 & 0 & -1.0791 & 0 \\ 0 & 0 & 0 & -0.9810 & 0 & -0.9810 \\ 0 & 0 & 0 & -0.0981 & 0 & -1.0791 \\ 0 & 0 & 0.6255 & 0 & 1.1067 & 0 \\ 0 & 0 & 0.1588 & 0 & 1.1693 & 0 \\ 0 & 0 & 0 & 0.6255 & 0 & 1.1067 \\ 0 & 0 & 0 & 0.1588 & 0 & 1.1693 \end{pmatrix}$$

Since, $rank(O_3) = 6$, we have the system to be observable for this choice of output vector.

Choice 4: $\mathbf{y(t)} = (\mathbf{x(t)}, \theta_\mathbf{1}(\mathbf{t}), \theta_\mathbf{2}(\mathbf{t}))$. Thus, we have

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$O_4 = \begin{pmatrix} 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \\ 0 & 0 & -0.9810 & 0 & -0.9810 & 0 \\ 0 & 0 & -0.5395 & 0 & -0.0491 & 0 \\ 0 & 0 & -0.0981 & 0 & -1.0791 & 0 \\ 0 & 0 & 0 & -0.9810 & 0 & -0.9810 \\ 0 & 0 & 0 & -0.5395 & 0 & -0.0491 \\ 0 & 0 & 0 & -0.0981 & 0 & -1.0791 \\ 0 & 0 & 0.6255 & 0 & 1.1067 & 0 \\ 0 & 0 & 0.2959 & 0 & 0.0794 & 0 \\ 0 & 0 & 0.1588 & 0 & 1.1693 & 0 \\ 0 & 0 & 0 & 0.6255 & 0 & 1.1067 \\ 0 & 0 & 0 & 0.2959 & 0 & 0.0794 \\ 0 & 0 & 0 & 0.1588 & 0 & 1.1693 \end{pmatrix}$$

Since, $rank(O_4) = 6$. We have the system as observable for this choice of output vector.

Thus, Choice 1,3 and 4 yield a output vector for which the system is observable.

2.2.2 Luenberger Observer

F. The Luenberger Observer is described as:

$$\begin{aligned} \dot{\hat{X}} &= A\hat{X} + BU + L(Y - \hat{Y}) \\ &= A\hat{X} + BU + LC(X - \hat{X}) \end{aligned} \tag{21}$$

It, essentially corrects the state estimate based on the true output values. The task is to choose suitable vaue for L. It is required that the error between the estimated state and true state should exponential go down to zero. Thus, $(X_e = X - \hat{X}) \to 0$. Looking at the error dynamics we have,

$$\begin{aligned} \dot{X} - \dot{\hat{X}} &= AX + BU - A\hat{X} - BU - LC(X - \hat{X}) \\ &= A(X - \hat{X}) - LC(X - \hat{X}) \\ &= (A - LC)(X - \hat{X}) \\ \Rightarrow \dot{X}_e &= (A - LC)X_e \end{aligned}$$

Thus, for stability, eigen values of $(A^T - C^T L^T)$ should have a negative real part. Thus, we can obtain the values for L using the Pole-Placement technique. Usually, it is required that we get an estimate of the state quickly and thus poles are placed further apart from the imaginary axis. But, there is a limit to how far we can go from imaginary axis. For a more practical case, where sensor noise is going to be present, the noise will amplify its magnitude and give a bad estimate. It was observed in simulations when poles were placed further apart the linearized system was able to converge to the actual state position quickly. But inital error in estimate was very large($>$4 radians). And also, when this was used on the non-linear system the error did not reduce and was always more than 11-15 degrees. After going back and forth between linear and non-linear implementation of the system the following poles gave a suitable response for each of the output vector.

For implementing the Luenberger Observer on the non-linear system following modifications were made to the Original Luenberger Observer

$$\dot{\hat{X}} = f(\hat{X}, U) + LC(X - \hat{X}) \tag{22}$$

Where, $f(\hat{X}, U)$ is the original non-linear state space form as given by the equation 12. Thus, the variation in $\hat{X}$ was predicted using the Simulink model in B.1.

OutputVector	Poles
$x(t)$	$[-0.1 \quad -0.2 \quad -2.2 \quad -1.9 \quad -2.1 \quad -1.6]$
$(x(t), \theta_2(t))$	$[-0.5 \quad -1 \quad -1.5 \quad -2 \quad -2.5 \quad -3]$
$(x(t), \theta_1(t), \theta_2(t))$	$[-2 \quad -3 \quad -4 \quad -5 \quad -6 \quad -7]$

2.2.3 Simulations

The algorithm used for implementing the Luenberger Observer is as follows:

$$\begin{aligned}
& Y \rightarrow lsim(sys, u, t, X_0) \\
& \hat{X} \rightarrow X_1 \\
& X_{est} \rightarrow \hat{X} \\
& k \rightarrow 2, dt \rightarrow 0.01 \\
for \quad & dt : dt : t \\
& d\hat{X} \rightarrow A\hat{X} + Bu(k) + L(Y(k) - C\hat{X}) \\
& \hat{X} \rightarrow \hat{X} + dt \times d\hat{X} \\
& X_{est} \rightarrow [X_{est}; \hat{X}] \\
& k \rightarrow k + 1 \\
& end
\end{aligned}$$

where, the system is simulated first and that response is stored in Y. The initial estimate $\hat{X}$ can take some value other than the actual value X_0. $u(k)$ and $Y(k)$ are the values of the control input and the actual state measured at that timestep respectively. This logic is simulated using a MATLAB function found at A.4. Also, a Simulink Model of the Cart and pendulum Estimator is created which can be found in section B.3. A step input of 100 N is chosen to obtain a reasonable displacement of the cart postion. Figure 4, 5 and 6 present the response of the observer on a Linearized system with the three different output vectors. Similarly, Figure 7, 8 and 9 present the response, when this observer is applied to the non-linear system.

Comments: As can be seen from the simulation plots the position is tracked very accurately. Figure 7b, 8b and 9b show that the position error max is about 3 cm. When, the observer is implemented on the linear system we see that the observer converges rapidly to the actual value for all the three quantities of interest. But, for the non-linear case the scenario is slightly different. In general it can be seen that the estimation error grows for a output vector which is not being observed directly. Like, as seen from the figure 7d and 7f for the case of all the three output vectors, the error in angle is always less than 1 degree. While, for the case of error in angle θ_1 which is not observed directly, it can be seen in figure 8d that the error can be more than 5 degrees. Same observation can be made for θ_1 and θ_2 in the case when only $x(t)$ is observed. Also, it was observed that error in the angles grew when the initial

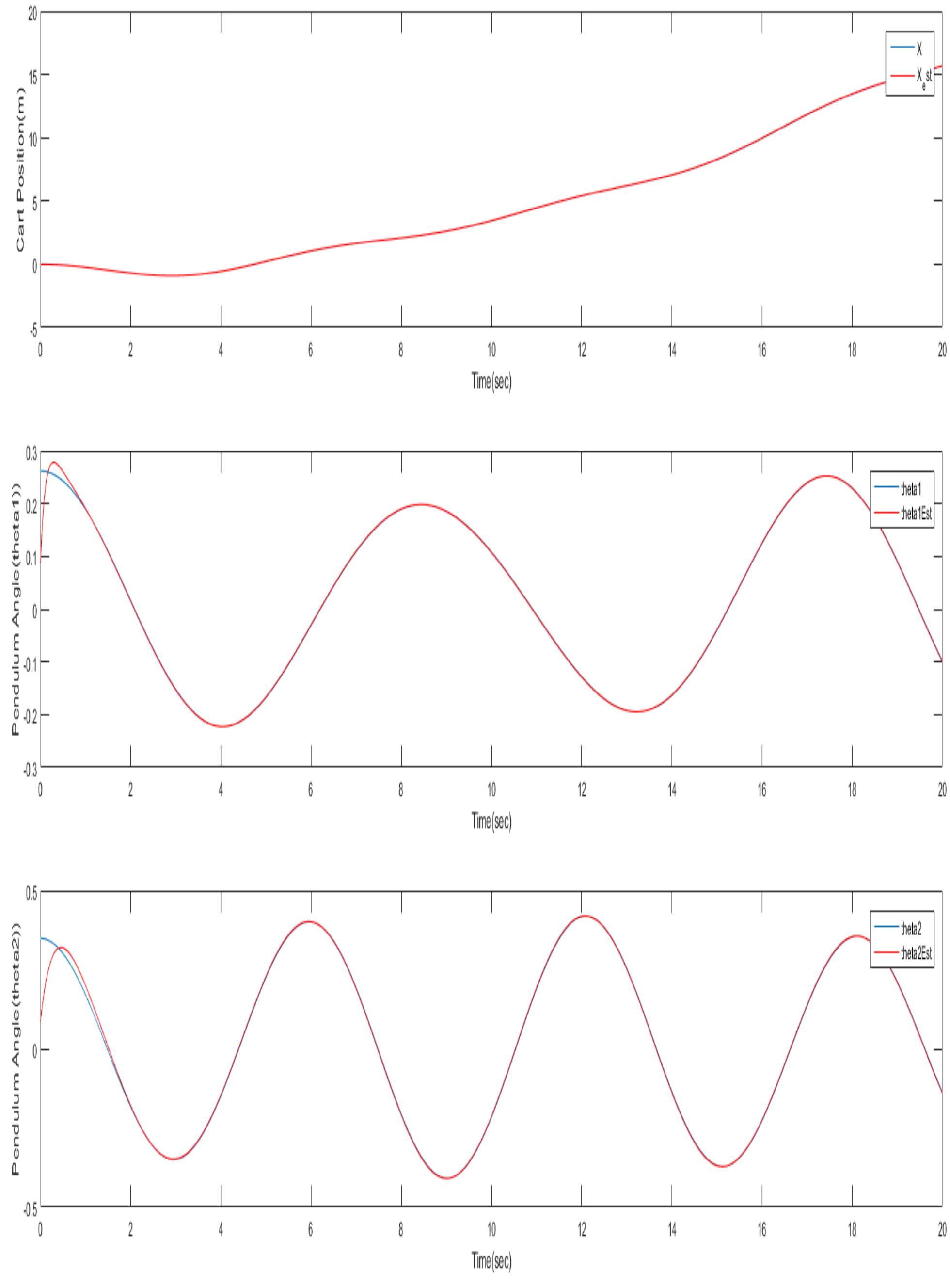


Figure 4: Response to step input of the actual and Estimated State on a linearized System with output vector $y(t) = (x(t), \theta_1(t),\theta_2(t))$, $X_0 = [0.2, 0, 15\frac{\pi}{180}, 0, 20\frac{\pi}{180}, 0]$ and $X_{est} = [0, 0, 0, 0, 0, 0]$

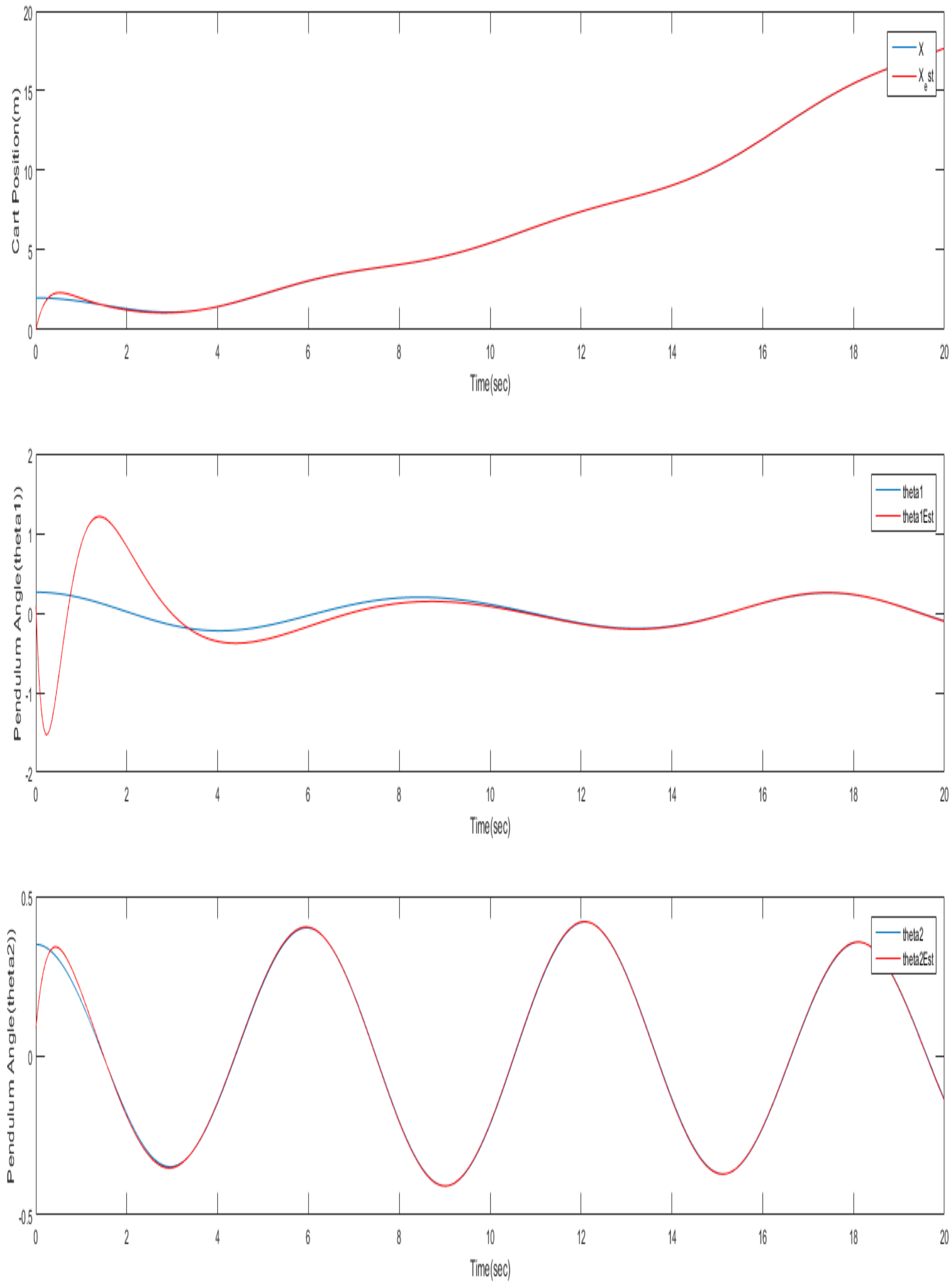


Figure 5: Response to step input of the actual and Estimated State on a linearized System with output vector $y(t) = (x(t),\, \theta_2(t))$, $X_0 = [0.2, 0, 5\frac{\pi}{180}, 0, 5\frac{\pi}{180}, 0]$ and $X_{est} = [0, 0, 5\frac{\pi}{180}, 0, 5\frac{\pi}{180}, 0]$

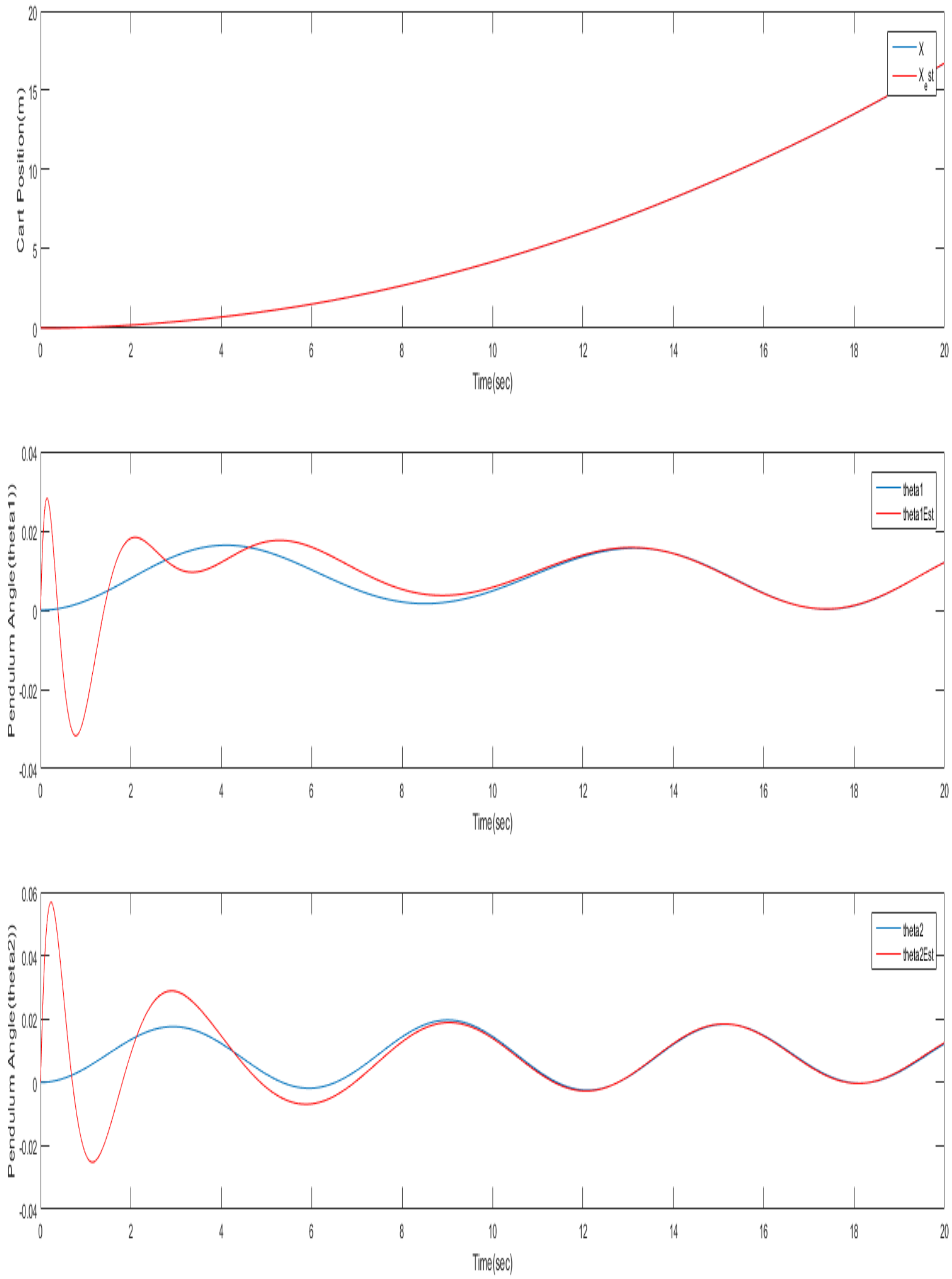
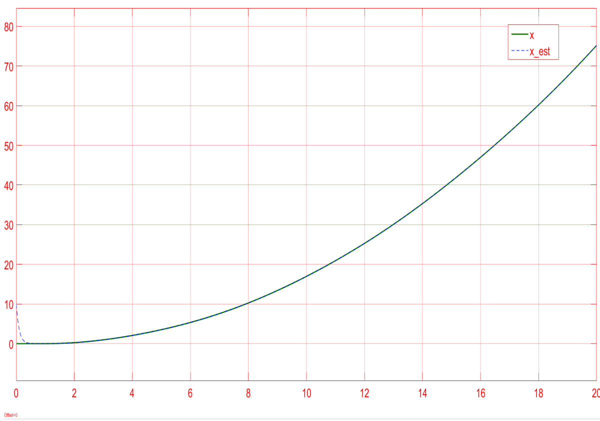
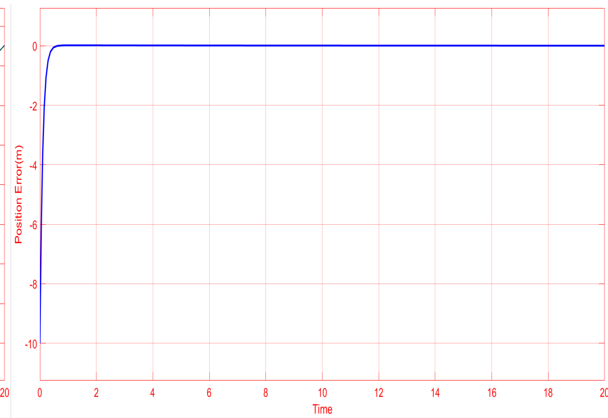


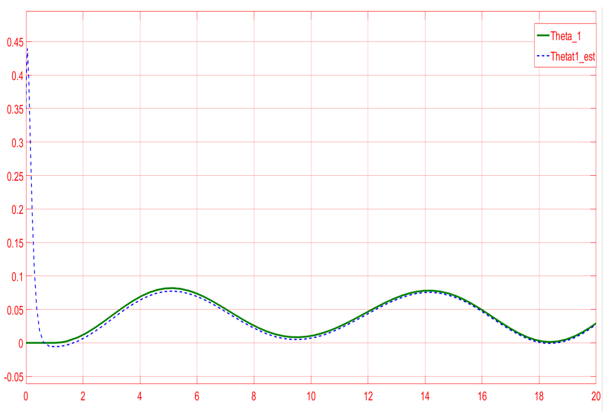
Figure 6: Response to step input of the actual and Estimated State on a linearized System with output vector $y(t) = (x(t),\, \theta_2(t)),\ X_0 = [0.2, 0, 5\frac{\pi}{180}, 0, 5\frac{\pi}{180}, 0]$ and $X_{est} = [0, 0, 5\frac{\pi}{180}, 0, 5\frac{\pi}{180}, 0])$



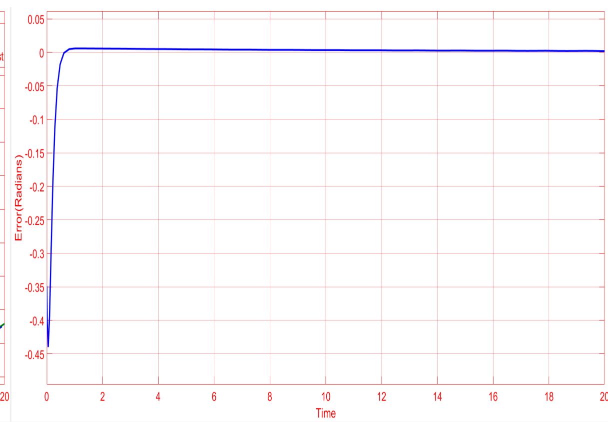
(a) Cart Postion(x)



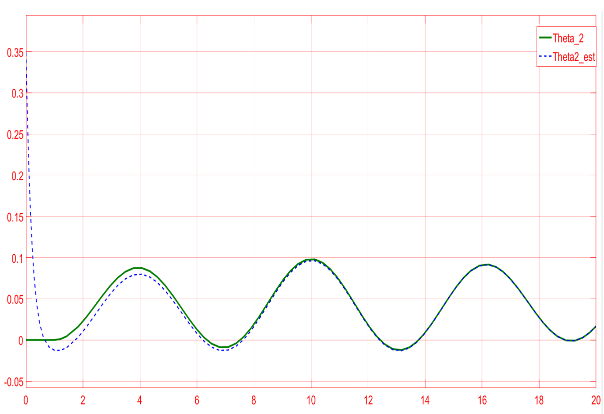
(b) Cart Postion Error(x - $\hat{x}$)



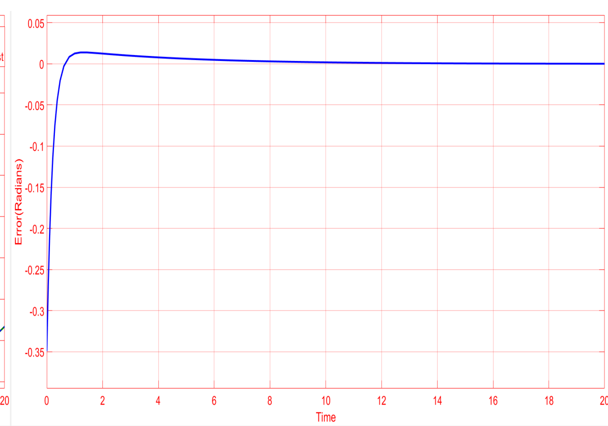
(c) Pendulum Angle (θ_1)



(d) Pendulum Angle Error ($\theta_1 - \hat{\theta_1}$)

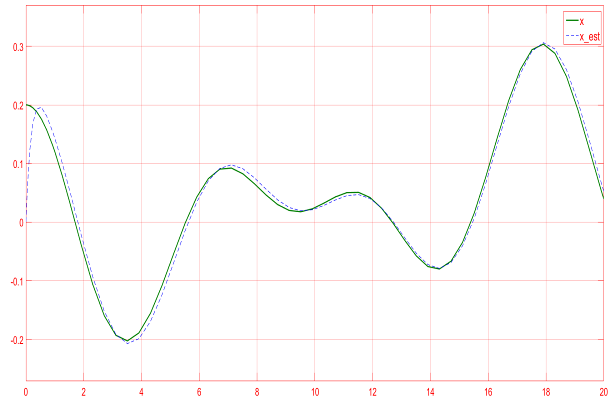


(e) Pendulum Angle (θ_2)

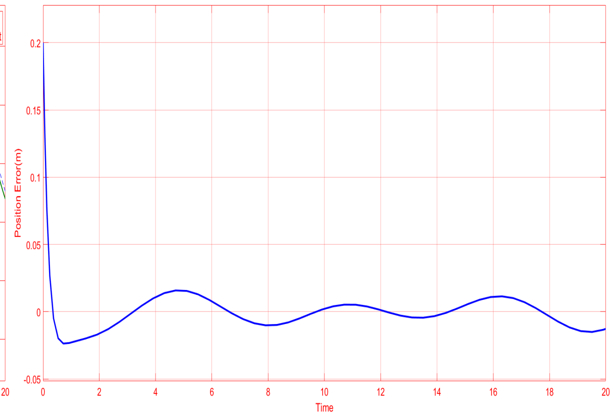


(f) Pendulum Angle Error ($\theta_2 - \hat{\theta_2}$)

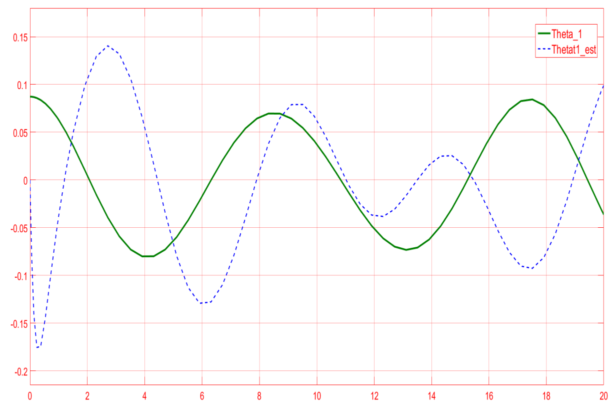
Figure 7: Response of Non-Linear System with $y(t) = (x(t), \theta_1(t), \theta_2(t))$, $X_0 = [0, 0, 0, 0, 0, 0]$ and $X_{est} = [0.2, 0, 15\frac{\pi}{180}, 0, 20\frac{\pi}{180}, 0]$



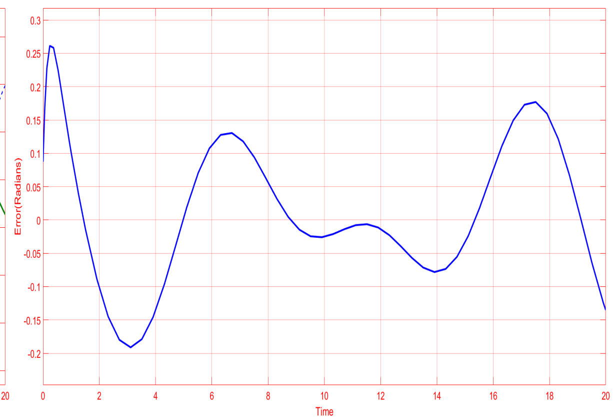
(a) Cart Postion(x)



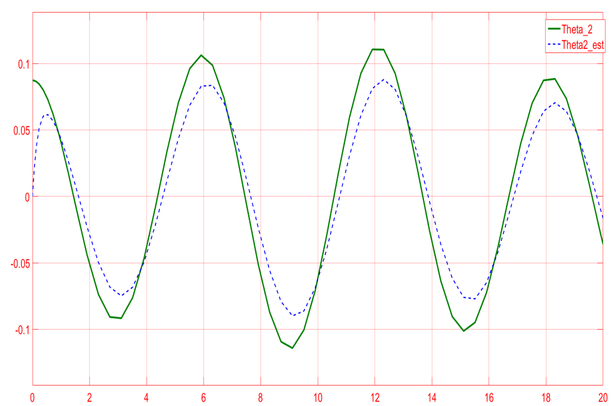
(b) Cart Postion Error(x - $\hat{x}$)



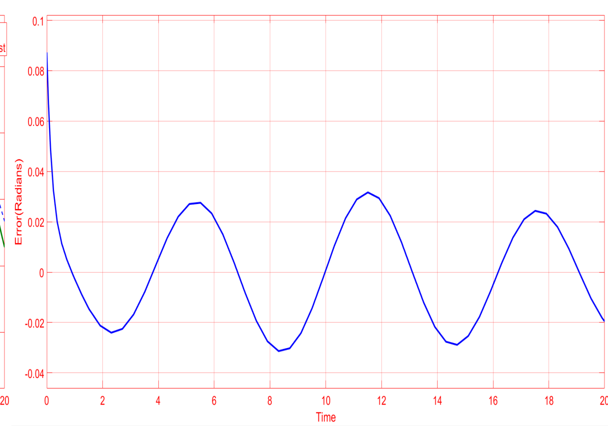
(c) Pendulum Angle (θ_1)



(d) Pendulum Angle Error ($\theta_1 - \hat{\theta_1}$)

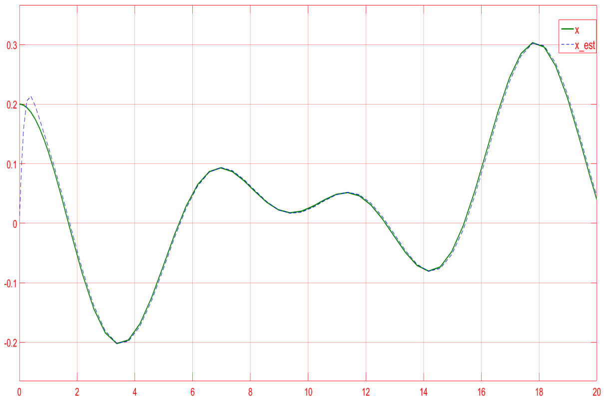


(e) Pendulum Angle (θ_2)

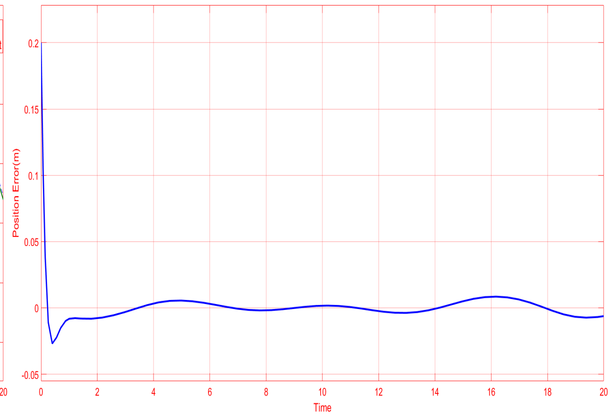


(f) Pendulum Angle Error ($\theta_2 - \hat{\theta_2}$)

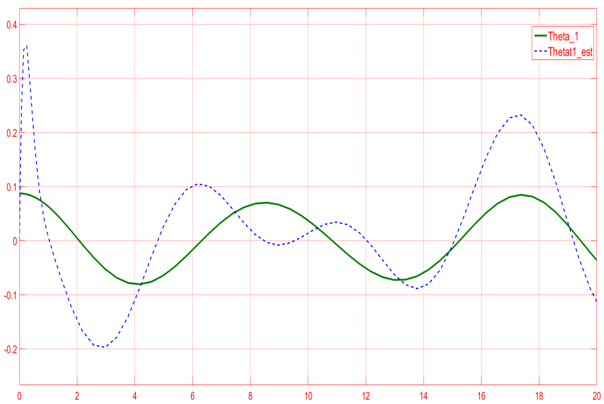
Figure 8: Response of Non-Linear System with $y(t) = (x(t), \theta_2(t))$,$X_0 = [0.2, 0, 5\frac{\pi}{180}, 0, 5\frac{\pi}{180}, 0]$ and $X_{est} = [0, 0, 5\frac{\pi}{180}, 0, 5\frac{\pi}{180}, 0]$



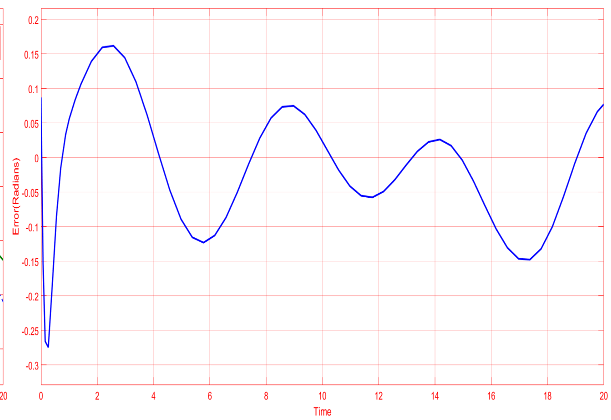
(a) Cart Postion(x)



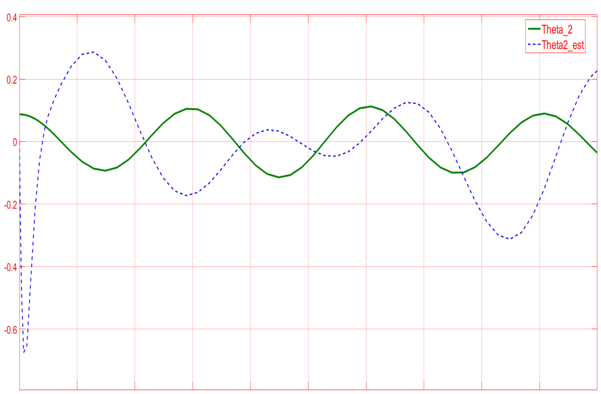
(b) Cart Postion Error(x - $\hat{x}$)



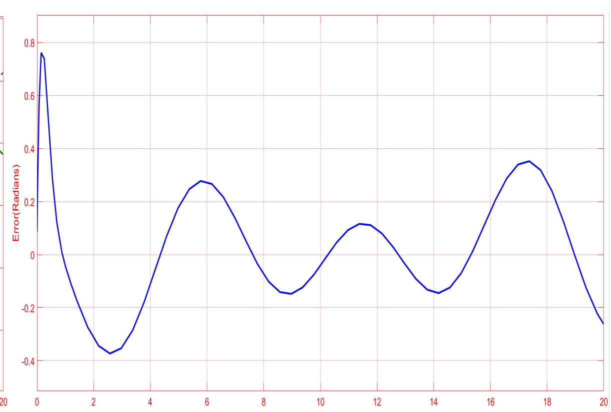
(c) Pendulum Angle (θ_1)



(d) Pendulum Angle Error ($\theta_1 - \hat{\theta_1}$)



(e) Pendulum Angle (θ_2)



(f) Pendulum Angle Error ($\theta_2 - \hat{\theta_2}$)

Figure 9: Response of Non-Linear System with $y(t) = x(t)$,$X_0 = [0.2, 0, 5\frac{\pi}{180}, 0, 5\frac{\pi}{180}, 0]$ and $X_{est} = [0, 0, 5\frac{\pi}{180}, 0, 5\frac{\pi}{180}, 0]$

estimation error was large. In conclusion, when all the three outputs are observed the response is the best and the error is minimum. The author believes that a better performance for the non-linear cases might be extracted from the other two cases by either better tuning or using some other **non-linear** observer.

2.3 LQG Output Feedback Controller

G. Linear Quadratic Gaussian is a control technique that combines the Optimal controller and the optimal Estimator(Kalman-Bucy). In other words, the gain K is chosen from the LQR technique and gain L is chosen from Kalman-Bucy Filter. The smallest output vector which is observable i.e $x(t)$ is used as the choice of output vector. A ToolBox Function 'Kalman' is used to obtain the optimal value of the Observer Gain L. With Kalman we assume that the sensors are affected by noise. Thus, in this case the position sensor is affected by noise. The state space form for this system will be written as

$$\begin{cases} \dot{X} = AX + BU + Bw \\ Y = CX + v \end{cases} \tag{23}$$

Where, the U is a known input and w and v are the known white Process and Measurement noises respectively. Since, these are Gaussian with zero mean, the **covariance of measurement noise** is $E(ww^T) = \mathbf{R}$ and the **covariance of Process Noise** is $E(vv^T) = \mathbf{Q}$. Since, we are observing only one output and have only one input, these covariances are going to have a single element. Hence, we assume that $\mathbf{Q} = \mathbf{0.2}$ and $\mathbf{R} = \mathbf{0.01}$. When operating with our output feedback controller, when we close the loop the control input $U = -K\hat{X}$. Therefore, writing the closed loop equations

$$\begin{aligned} \dot{X} &= AX + B(-K\hat{X}) + Bw \\ \dot{\hat{X}} &= A\hat{X} + B(-K\hat{X}) + LC(X - \hat{X}) \\ &= LCX + (A - BK - LC)\hat{X} \end{aligned}$$

$$\Rightarrow \begin{pmatrix} \dot{X} \\ \dot{\hat{X}} \end{pmatrix} = \begin{pmatrix} A & BK \\ LC & A - BK - LC \end{pmatrix} \begin{pmatrix} X \\ \hat{X} \end{pmatrix} + Bw$$

where, L is the optimal observer gain and K is the optimal control gain. Expressing in terms of $X_e = X - \hat{X}$ i.e substituting for $\hat{X} = X - X_e$. We have,

$$\begin{pmatrix} \dot{X} \\ \dot{X}_e \end{pmatrix} = \begin{pmatrix} A - BK & BK \\ 0 & A - LC \end{pmatrix} \begin{pmatrix} X \\ \hat{X} \end{pmatrix} \begin{pmatrix} X \\ X_e \end{pmatrix} + Bw \tag{24}$$

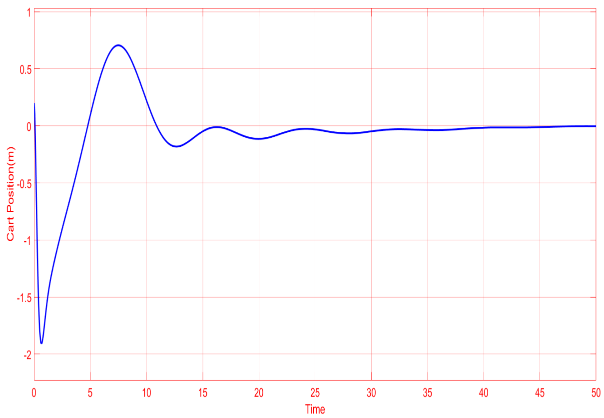
Since, the Matrix is upper diagonal the eigen values can be calculates separately for each of the diagonal elements. In other words, both of these components can be designed independently and the system will still be stable. This is known as the **Separation principle**.

2.3.1 Simulations

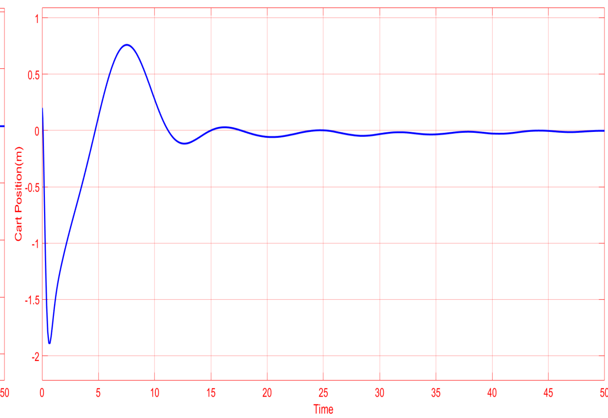
Because simulating a LQG controller requires that we perturb the system with noise, both simulations on a linearized System and Non-Linear system are performed more efficiently using Simulink. The Linear model is located at B.4 and the non-linear model is located at B.5. Figure 10 shows the plots that are obtained after the simulating from initial conditions with an LQG controller on the linearized and the non-linear system.

2.4 Reference Tracking

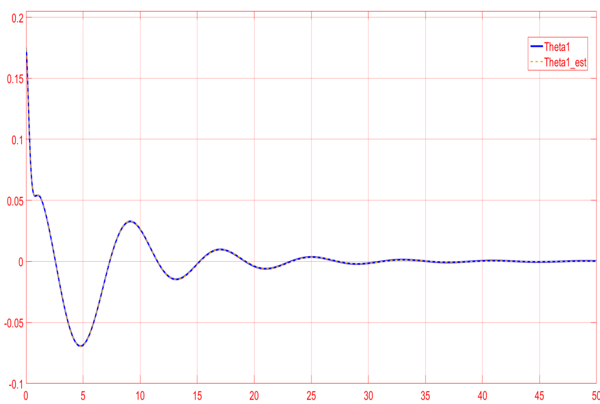
It is required that the Cart should reach a desired position x_d. Let, $r = x_d$ and for time being, discuss the case where the state vector is known. With, full state feedback the control input takes



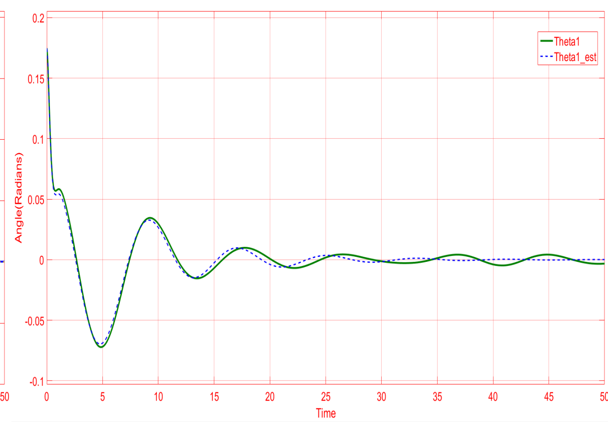
(a) Linear system Cart Postion(x)



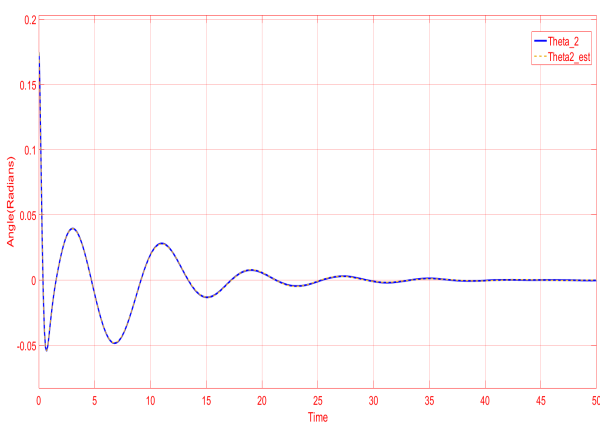
(b) Non-Linear system Cart Postion(x)



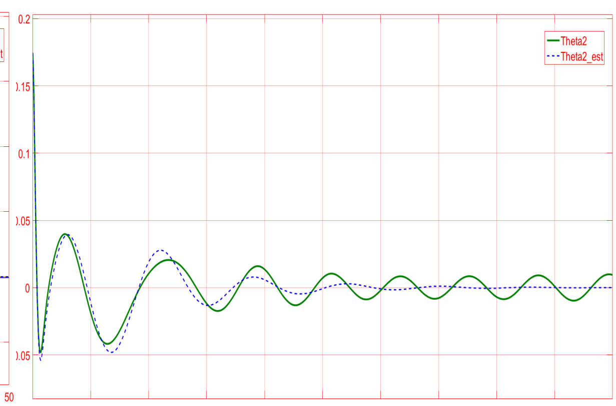
(c) Linear system Pendulum Angle (θ_1)



(d) Non-Linear system Pendulum Angle (θ_1)



(e) Linear system Pendulum Angle (θ_2)



(f) Non-Linear system Pendulum Angle (θ_2)

Figure 10: Response of Linear and Non-Linear System with LQG controller to initial conditions with $X_0 = [0.2, 0, 10\frac{\pi}{180}, 0, 10\frac{\pi}{180}, 0]$ and $X_{est} = [0, 0, 10\frac{\pi}{180}, 0, 10\frac{\pi}{180}, 0]$

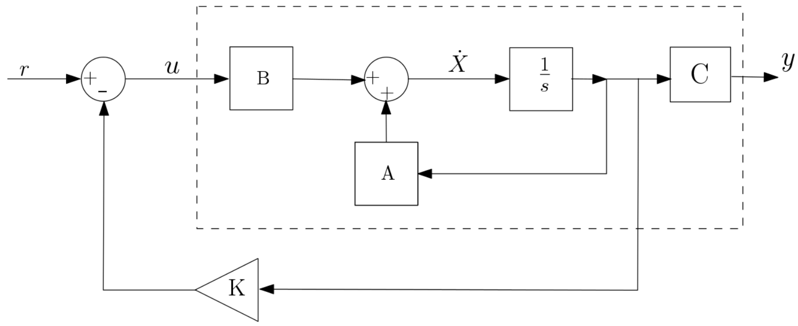


Figure 11: System Schematic with reference tracking for Position of the Cart

the form $u = r - KX$. This is illustrated in the figure 11. Writing the closed Loop equations for this we have:

$$\begin{aligned}\dot{X} &= AX + B(r - KX)\\ &= (A - BK)X + Br\end{aligned} \tag{25}$$

Thus, for the closed-Loop System the new input $u = r$ and $A_c = (A - BK)$, $B_c = B$. But, in our actual system(with Output Feedback Controller) what we have is infact a State Estimate and Process and Measurement perturbed with noise as discussed in the Equation 23. Thus, the actual system looks like as in Fig 12, and the state space equations are rewritten as

$$\begin{aligned}\dot{X} &= (A - BK)\hat{X} + Br + Bw\\ Y &= C\hat{X} + v\end{aligned} \tag{26}$$

When this system is simulated(MATLAB script is located at A.5) giving the known input for the

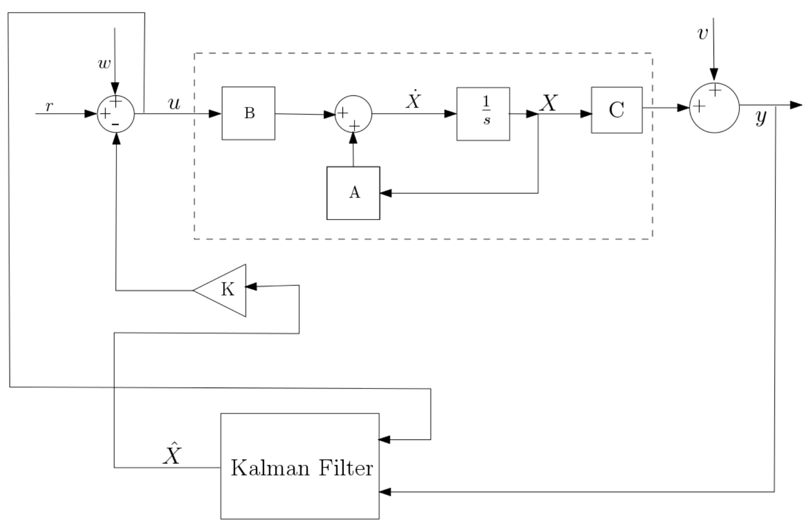


Figure 12: System Schematic with Kalman Filtering for Position Control of the Cart

position there is always a steady state error as illustrated in Figure 13. There are two solutions to resolve this problem.

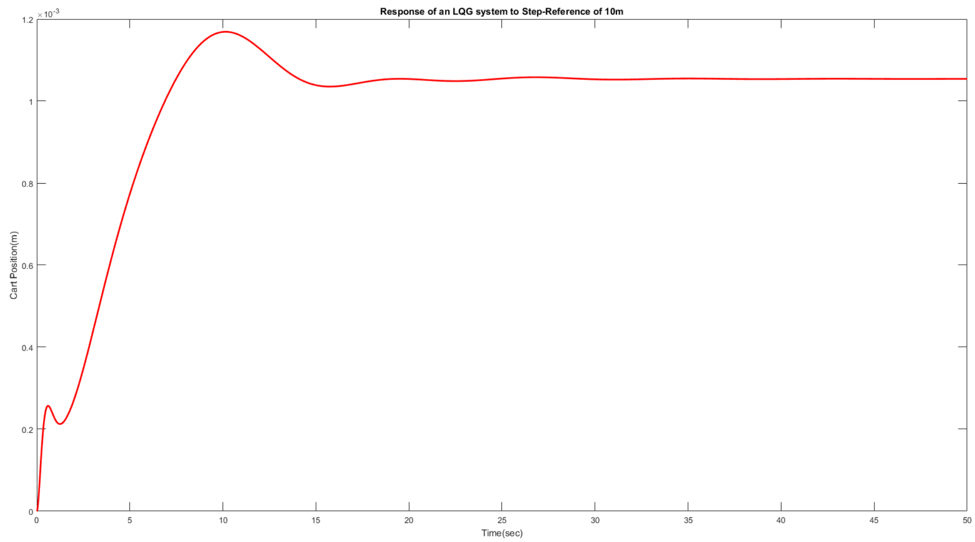


Figure 13: The inital LQG controller is unable to take the cart to a desired position of 10m and settles at a steady state at far less value in position

- **Precompensator:** Since, we are comparing a reference in position to the entire state vector there is an expected difference in the control signal and thus the steady state performance is not as expected. This is compensated for by premultipling the reference with a suitable gain. The gain is calculated using a MATLAB function 'rscale.m' which is available at [1]. The schematic looks like as in the Figure 14. As will be seen in simulations in the next section this strategy is very effective and attains the reference position of the cart. It is able to reject the disturbances from the noise and also other constant disturbances of relatively less magnitude. It is robust enough to adjust for different values of the reference input for the position. However, for greater magnitude of disturbances integral term is observed to be a more robust solution.
- **Integral Term:** The trick is to augment the steady state error in the cart position as a state variable with an expectation that this error term will also exponentially go down to zero. Let, $x_e = x - r$ and $X_I = \int (x_e) dt$. Rewriting the new State-Space Form, which is a complete version including this time with an additional input of external disturbances U_d

$$\begin{cases} \begin{pmatrix} \dot{X} \\ x_e \end{pmatrix} = \begin{pmatrix} A & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ X_I \end{pmatrix} + \begin{pmatrix} 0 \\ -1 \end{pmatrix} r + \begin{pmatrix} B \\ 0 \end{pmatrix} U + \begin{pmatrix} B \\ 0 \end{pmatrix} w + \begin{pmatrix} B \\ 0 \end{pmatrix} U_d \\ \dot{X}_a = A_I X_a + B_I U + B_I w + B_I U_d + \begin{pmatrix} 0 \\ -1 \end{pmatrix} r \\ Y = (C \quad 0) \begin{pmatrix} X \\ X_I \end{pmatrix} + v \end{cases} \tag{27}$$

Thus, as it can be seen, augmenting the state vector(X_a) with an integral term introduces an additional input to system in the form of reference signal r. Now, for the tracking problem

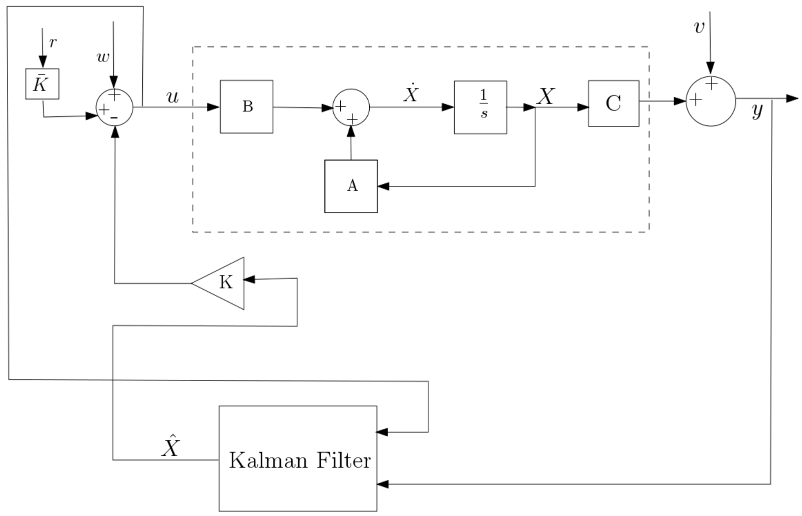


Figure 14: System Schematic with Precompensator for Position Control of the Cart

taking the control input $U = r - K\hat{X}_a$.Where, $\hat{X}_a = \begin{pmatrix} \hat{X} \\ \hat{x} - r \end{pmatrix}$. Hence, the equations become

$$\begin{aligned}
\dot{X'} &= A_I X_a + B_I(r - K\hat{X}_a) + B_I w + B_I U_d + \begin{pmatrix} 0 \\ -1 \end{pmatrix} r \\
&= A_I X_a - B_I K\hat{X}_a + \left[B_I + \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right] r + B_I w + B_I U_d \\
&= A_I X - B_I K\hat{X}_a + \begin{pmatrix} B \\ -1 \end{pmatrix} r + B_I w + B_I U_d
\end{aligned}$$

A point to notice here is that $\hat{X}_a$ can be computed normally by just computing the original state estimation vector $\hat{X}$. Thus, even with the augmented state vector the Kalman State Estimator will not be affected. Now, using the **Separation principle** we will assume that $\hat{X}_a$ is known and is equal to X_a and design our LQR controller based on this X_a to get a suitable response. Also, for the sake of simplicity in tuning the gains the noise term $B_I w$ will be neglected(which are going be negligible when compared to the large disturbance input). Thus, rewriting the closed-loop state space equations for this system with the augmented state vector X_a we have

$$\begin{cases} \dot{X}_a = (A_I - B_I K)X_a + \begin{pmatrix} B \\ -1 \end{pmatrix} r + B_I U_d \\ Y = (C \quad 0) \begin{pmatrix} X \\ X_I \end{pmatrix} + v \end{cases} \tag{28}$$

Thus, for the closed loop system $A_c = A_I - B_I K$. Let, $B_2 = \begin{pmatrix} B \\ -1 \end{pmatrix}$. Thus, There are two inputs to this system. One, is the disturbance force U_d and other is the reference input r. The control Matrix corresponding to these are B_I and B_2 respectively. Also, $C_c = (C \quad 0)$

2.4.1 Simulations

– **Using Precompensator:** The precompensating gain is added to the reference signal and

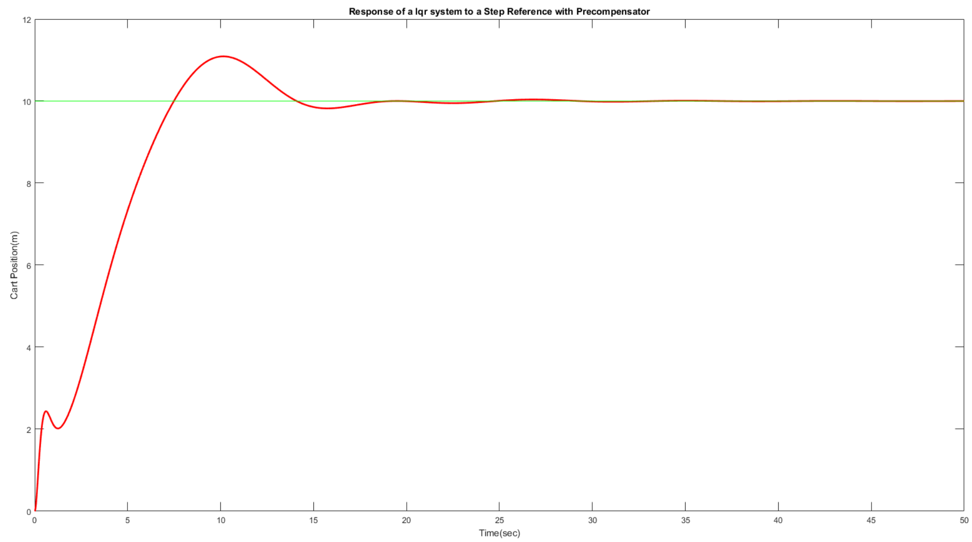


Figure 15: Plot showcasing that adding a precompensator achieves the desired tracking. However, when magnitude of external disturbances is more than $2000N$ we start getting steady state error again

the linearized system is simulated. A MATLAB script and a Simulink model to do this is found at A.6 and B.5 respectively. Figure 15 shows the attained behaviour as a result of this component.

– **Integral Action:** For the Linearized System the MATLAB function that implements the proposed augmented closed loop state space form of equation 28 is Located at A.7. Now, for tuning the weights, again we have $Q = C' \times C$. There is one more value in the Q Matrix which needs tuning. This is $Q(7, 7)$. The three other values are same as the one used in the section 2.1. Thus, values of the chosen weights are

$$
\begin{aligned}
Q(1,1) &= 90000000 \\
Q(3,3) &= 80000000000 \\
Q(5,5) &= 70000000000 \\
\mathbf{Q(7,7)} &= \mathbf{10000000}
\end{aligned}
$$

while keeping the value of $R = 1$. The response seen in the figure 16 is satisfying considering the time taken to reach the reference and a very less overshoot.

2.5 Final Design

The Integral Action described with reference to a linear system is applied with a LQG output feedback controller on the original non-linear system as is illustrated in B.6. As shown previously, the kalman state estimator is not affected with augmented state vector and hence, implementing a integral component to non-linear system is as straightforward as adding a extra input signal equal to $-Kx_e = K(r - \hat{x})$. The gain required some tuning and was not same as the one used on the linear system. Figure 17 shows the simulation plots. The Cart successfully attains the desired reference of 100m and the pendulum angles at this position are very near to zero degrees(< 0.5. Figure 17 clearly depicts the quality of the controller, where the overshoot in the pendulum angles is always **less** than about 17 degrees. Also, the cart attains the position within 25 seconds.

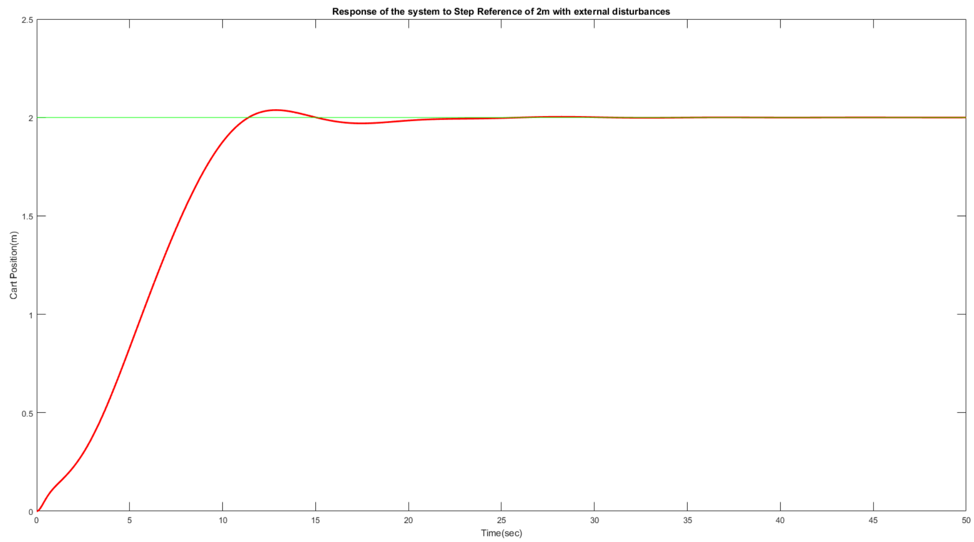
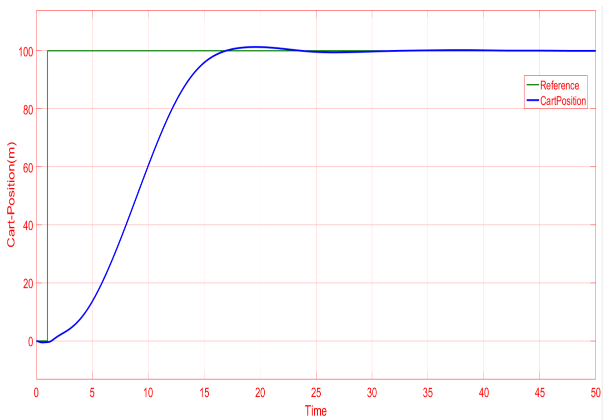


Figure 16: Plot showcasing the augmented state vector using integral action achieves the desired tracking. It is capable of rejecting a magnitude of external disturbances even larger than $2000N$

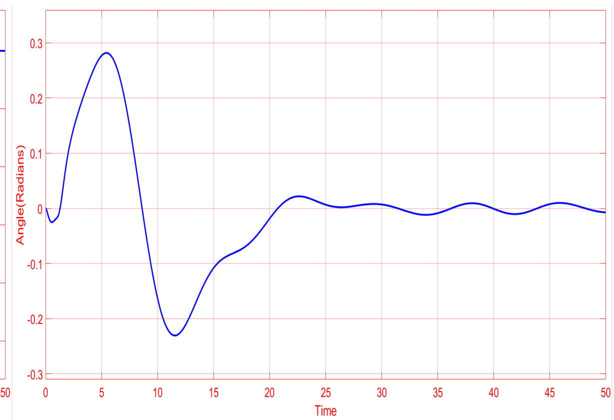
As a fun component the complete output-feedback controller is applied to a simscape model to visualize the system in action. The simscape model is located in B.7 and a snapshot of the pendulum system in action is shown in the figure 18. The plots obtained from the simscape model and the non-linear model created in Simulink were in agreement. This shows that using the Simscape Workbench can be a good technique to verify the validity of the equation of the motion.

3 Conclusion

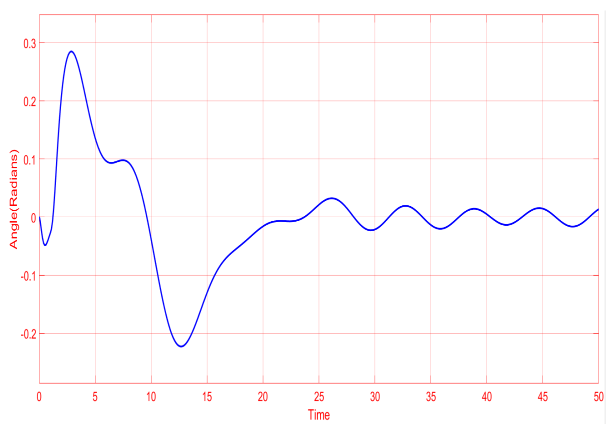
In this project the entire process of deriving the dynamic model and developing a controller to achieve a desired response was obtained. Motivation behind Linearizing the dynamic model was clearly palpable, wherei n techniques like Controllability, Observability, LQR, LQG are only possible to implement on a linear system. Control Toolbox is very handy in MATLAB and various functions were made familiar as a result of this project. To be able to augment the state vector to achieve steady state error cancellation and think of obtaining an extra input was a learning experience. Final Design was especially satisfying as it took some effort tuning and obtaining a desired response. Overall, this project was treated as a guest and the author acted as a good host:)



(a) Cart Postion(x)



(b) Pendulum Angle (θ_1)



(c) Pendulum Angle (θ_2)

Figure 17: Response of the non-linear Cart and Two Pendulum system to a constant reference in position(x) = 100m using a LQG controller supplemented with Integral Action.

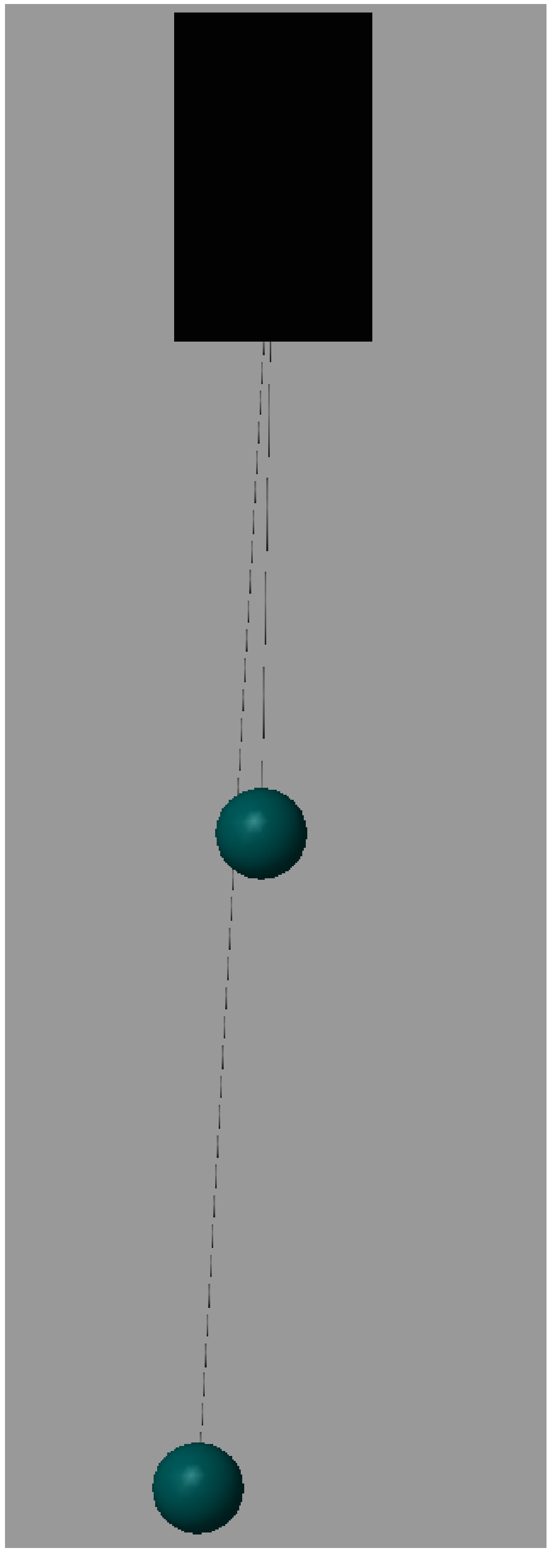


Figure 18: A snap showing the created simscape model in action during the simulation

A MATLAB code

A.1 Utility Function used at the start of each file

```matlab
function [A,B,C,D] = getParams()
%Utility Function to fetch the Parameters of the problem
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.81;

A = [0 1 0 0 0 0;0 0 -m1*g/M 0 -m2*g/M 0;0 0 0 1 0 0;0 0 -(M + m1)*g/(M*l1) 0 -m2*g/(M*l1) 0;0 0 0 0 0 1;0 0 -m1*g/(M*l2) 0 -(M + m2)*g/(M*l2) 0];
B = [0;1/M;0;1/(M*l1);0;1/(M*l2)];
C = [1 0 0 0 0 0;0 0 1 0 0 0;0 0 0 0 1 0];
D = 0;
end
```

A.2 Simulate Open Loop Response

```matlab
function simulateOL()
%Simulate response of the cart and the pendulum system in Open Loop
[A,B,C,D] = getParams();
states = {'x','x_dot','theta1','theta1_dot','theta2','theta2_dot'};
inputs = {'F'};
outputs = {'x','theta1','theta2'};

sys_ol = ss(A, B, C, D, 'statename',states,'inputname',inputs,'outputname',outputs);

%Response to some initial conditions and zero-input
x0 = [0,0,20*pi/180,0,20*pi/180,0];
t = 0:0.01:50;
F = zeros(size(t));
%Simulate using lsim, Can use 'initial' command too
[Y,~,~] = lsim(sys_ol,F,t,x0);

% plot(t,Y(:,2),'g');
% hold on;
% plot(t,Y(:,3),'b');
% figure
plot(t,Y(:,1),'r');
xlabel('Time(t)')
ylabel('Position(X)')
title('Response of a cart and pendulum system to inital conditions')
end
```

A.3 Simulate the response of LQR controller

```matlab
function simulateLQR()
[A,B,C,D] = getParams();

Q = C' * C;
Q(1,1) = 90000000;
Q(3,3) = 80000000000;
Q(5,5) = 70000000000;

R = 1;
[K,~,~] = lqr(A,B,Q,R);

states = {'x','x_dot','theta1','theta1_dot','theta2','theta2_dot'};
inputs = {'F'};
outputs = {'x','theta1','theta2'};

sys_cl = ss(A - B * K, zeros(size(B)), C, D, 'statename',states,'inputname',inputs,'outputname',outputs);

%Check the eigen values of the closed loop system
%eig(A - B * K)

x0 = [0,0,15*pi/180,0,20*pi/180,0];
t = 0:0.01:50;
F = zeros(size(t));
[Y,~,X] = lsim(sys_cl,F,t,x0);

%Calculate the control input as a function of time
u = zeros(size(t));
for i = 1:size(X,1)
   u(i) = K * (X(i,1:6))';
end

%Plots
[AX,~,~] = plotyy(t,Y(:,2),t,Y(:,3),'plot');
set(get(AX(1),'Ylabel'),'String','pendulum angle theta1 (radians)')
set(get(AX(2),'Ylabel'),'String','pendulum angle theta2 (radians)')
xlabel('Time(t)');
figure
plot(t,Y(:,1),'r', 'linewidth',2)
xlabel('Time(t)')
ylabel('Cart Position (m)')
figure
plot(t,u)
xlabel('Time(sec)')
ylabel('Control Input(N)');
end
```

A.4 Simulate the response of Luenberger Observer

```
function simulateLuenberger()
[A,B,~,D] = getParams();

%When only x(t) is the chosen output vector
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C1 = [1 0 0 0 0 0];
C = C1;
%P1 = [-2 -3 -4 -5 -6 -7];
P1 = [-1 -1.2 -1.4 -1.6 -1.8 -2];
P = P1;
outputs = {'x'};

%When x(t) and theta2(t) are the chosen output vectors
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% C3 = [1 0 0 0 0 0;0 0 0 0 1 0];
% C = C3;
% P3 = [-0.5 -1 -1.5 -2 -2.5 -3];
% P = P3;
% outputs = {'x','theta2'};

%when x(t), theta1(t) and theta2(t) are the Output Vectors
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% C4 = [1 0 0 0 0 0;0 0 1 0 0 0;0 0 0 0 1 0];
% C = C4;
% P4 = [-2 -3 -4 -5 -6 -7];
% P = P4;
% outputs = {'x','theta1','theta2'};

X0 = [0;0;0;0;0;0];
%Xhat = [0;0;0;0;0;0];
Xhat = [0.02;0;0;0;0;0];

L = place(A',C',P)';

states = {'x','x_dot','theta1','theta1_dot','theta2','theta2_dot'};
inputs = {'F'};

sys_ol = ss(A, B, C, D, 'statename',states,'inputname',inputs,'outputname',outputs);

%Step Response to an input of 100 N
t = 0:0.01:20;
u = 100 * ones(size(t));
[Y,~,X] = lsim(sys_ol,u,t,X0);

X_est = Xhat';
k = 2;
for n = 0.01:0.01:20
    dXhat = A * Xhat + B .* u(k) + L * (Y(k,:)' - C*Xhat);
    Xhat = Xhat + 0.01.*dXhat;
    X_est = [X_est;Xhat'];
    k = k + 1;
end
```

```
figure
subplot(3,1,1), plot(t,X(:,1)),hold on,plot(t,X_est(:,1),'r')
xlabel('Time(sec)'),ylabel('Cart Position(m)'),legend('X','X_est')
subplot(3,1,2), plot(t,X(:,3)),hold on,plot(t,X_est(:,3),'r')
xlabel('Time(sec)'),ylabel('Pendulum Angle(theta1))'),legend('theta1','theta1Est')
subplot(3,1,3), plot(t,X(:,5)),hold on,plot(t,X_est(:,5),'r')
xlabel('Time(sec)'),ylabel('Pendulum Angle(theta2))'),legend('theta2','theta2Est')
end
```

A.5 Illustrate Steady state Error in Reference Tracking

```
function simulateReferenceLQG()
%This illustrates steady state error with a LQG controller

[A,B,~,D] = getParams();
C = [1 0 0 0 0 0];

Q = C' * C;

Q(1,1) = 90000000;
Q(3,3) = 80000000000;
Q(5,5) = 70000000000;

R = 1;
[K,~,~] = lqr(A,B,Q,R);

%Get the Kalman State Estimator
sys_1 = ss(A,[B B],C,[zeros(1,1) zeros(1,1)]);

Rn = 10^-2 * eye(1);
Qn = 0.2;

sensors = [1];
known = [1];
[~,L,~] = kalman(sys_1,Qn,Rn,[],sensors,known);

states = {'x','x_dot','theta1','theta1_dot','theta2','theta2_dot','e_1','e_2','e_3','e_4','e_5','e_6'};
inputs = {'F'};
outputs = {'x'};

Ac = [A-B*K B*K;zeros(size(A)) A-L*C];
Bc = [B;zeros(size(B))];
Cc = [C zeros(size(C))];
sys_cl = ss(Ac,Bc,Cc,D, 'statename',states,'inputname',inputs,'outputname',outputs);

init_pos = [0,0,0];
x0 = [init_pos(1);0;init_pos(2);0;init_pos(3);0;init_pos(1);0;0.001*init_pos(2);0;0.001*init_pos(3);0];
t = 0:0.01:50;
F = 10*ones(size(t));
[Y,~,~] = lsim(sys_cl,F,t,x0);
figure
```

```
plot(t,Y(:,1),'r','linewidth',2);
ylabel('Cart Position(m)')
xlabel('Time(sec)')
title('Response of an LQG system to Step-Reference of 10m')
end
```

A.6 Illustrate action of precompensator to cancel the steady-state error

```
function simulateReferenceLQR()
%Use this to illustrate the Precompensator
[A,B,~,D] = getParams();

C = [1 0 0 0 0 0];

Q = C' * C;
Q(1,1) = 90000000;
Q(3,3) = 80000000000;
Q(5,5) = 70000000000;

R = 1;
[K,~,~] = lqr(A,B,Q,R);

states = {'x','x_dot','theta1','theta1_dot','theta2','theta2_dot'};
inputs = {'F'};
outputs = {'x'};

%Use the Precompensator to magnify the input. Illustrated using B*Nbar
sys_ss = ss(A,B,C,0);
Nbar = rscale(sys_ss,K);
sys_cl = ss(A - B * K, B*Nbar, C, D, 'statename',states,'inputname',inputs,'outputname',
    outputs);

%If stability is to be checked
%eig(A - B * K)

x0 = [0,0,0,0,0,0];
t = 0:0.01:50;
F = 10* ones(size(t));
[Y,~,~] = lsim(sys_cl,F,t,x0);

plot(t,Y(:,1),'r', 'linewidth', 2);
ylabel('Cart Position(m)')
xlabel('Time(sec)')

hold on
plot(t,F,'g')

title('Response of a lqr system to a Step Reference with Precompensator')
end
```

A.7 Illustrate the action of Integration to cancel the steady-state error

```matlab
function simulateReferenceLQRIntegral()
%Use this to illustrate the Integral Action to resolve the steady State
%Error
[A,B,~,D] = getParams();
C = [1 0 0 0 0 0];

%Prepare the Augmented Matrices
A_I = [A zeros(size(A,1),1);eye(1,size(A,2)) 0];
B_I = [B;zeros(size(B,2))];
B_2 = B_I;
B_2(end) = -1;
C_I = [C 0];

%After augmenting the state vector the system may or may not be
%controllable. This check is to confirm that the system is controllable
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%rank(ctrb(A_I,B_I))

%Setup the Weights and get the optimal gain
Q = C_I' * C_I;
Q(1,1) = 90000000;
Q(3,3) = 80000000000;
Q(5,5) = 70000000000;
Q(7,7) = 10000000;

R = 1;
[K,~,~] = lqr(A_I,B_I,Q,R);

states = {'x','x_dot','theta1','theta1_dot','theta2','theta2_dot','x_int'};
inputs = {'F','disturb'};
outputs = {'x'};

%Precompensation based on the original plant model
%Evaluate K2 First based on just the original plant model
% sys_ss = ss(A,B,C,zeros(size(C,1),size(B,2)));
% Nbar = rscale(sys_ss,K2)
%Apply this as B*Nbar. This cannot reject constant disturbances

%Closed Loop Plant with the Integral action
sys_cl = ss(A_I - B_I * K, [B_2 B_I], C_I, D, 'statename',states,'inputname',inputs,'outputname',outputs);

%Initial State
%X0 = [0,0,0,0,0,0,0];
r = 2;
Fd = 1000;
t = 0:0.01:50;
F = r * ones(size(t,2),1);
disturb = Fd*ones(size(t,2),1);
[Y,~,~] = lsim(sys_cl,[F disturb],t);

plot(t,Y(:,1),'r','linewidth',2);
```

```
xlabel('Time(sec)')
ylabel('Cart Position(m)')
hold on
plot(t,r*ones(size(t)),'g')

title('Response of the system to Step Reference of 2m with external disturbances')
end
```

B Simulink Models

B.1 Cart And Pendulum Non-Linear Model

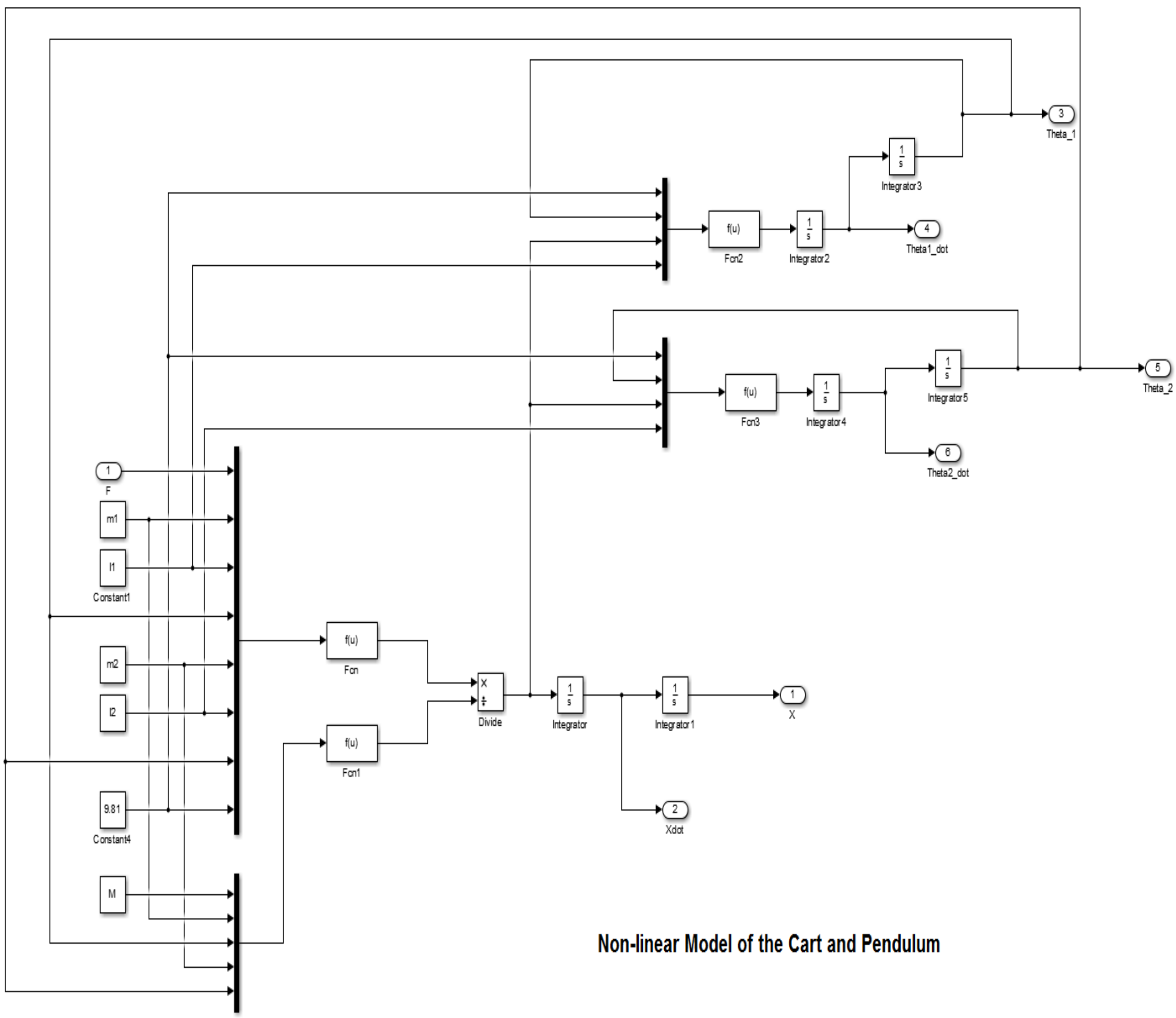


Figure 19: Non-Linear Model of the Cart and Pendulum system

B.2 LQR control on the non-linear model

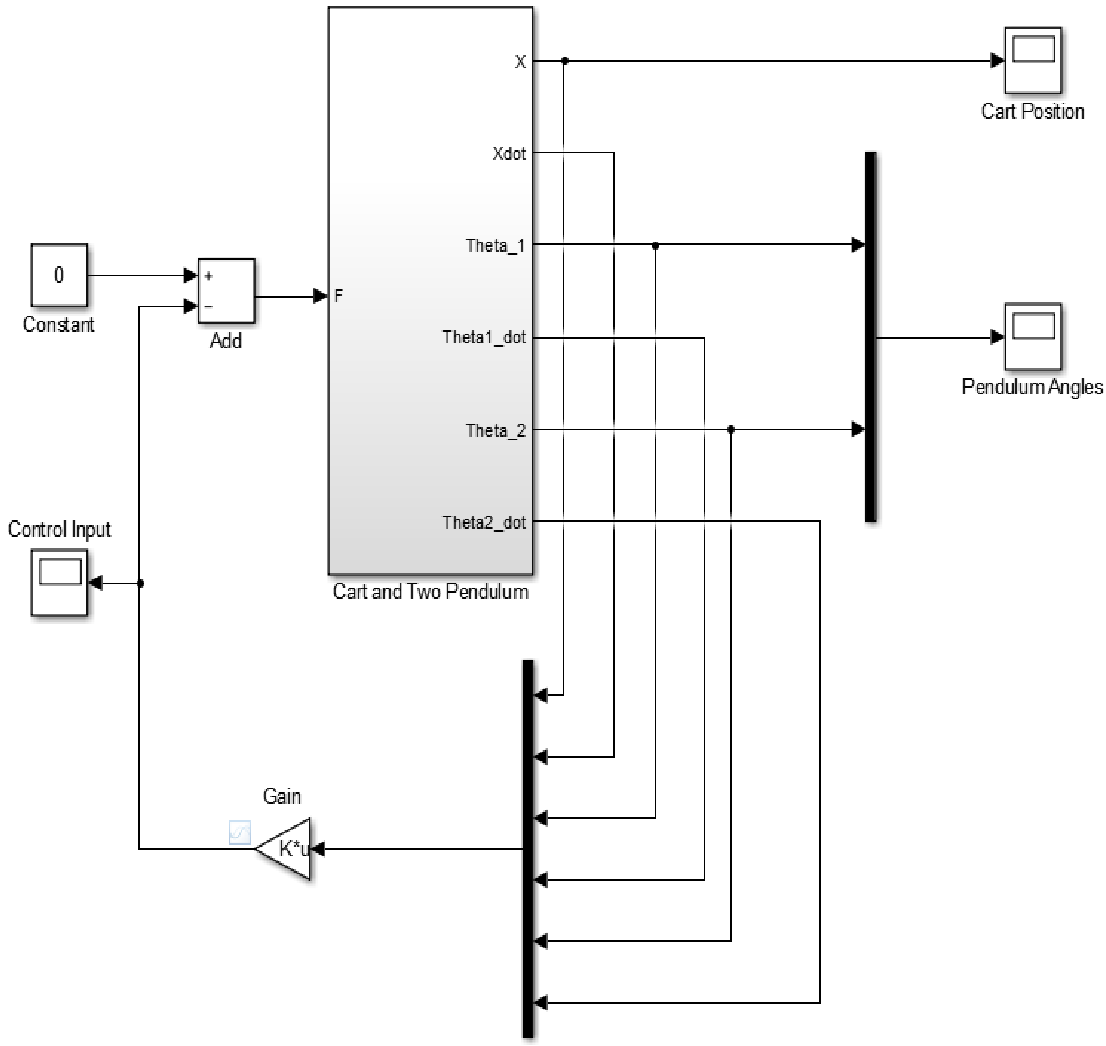


Figure 20: LQR controller

B.3 Luenberger Observer on the non-linear model

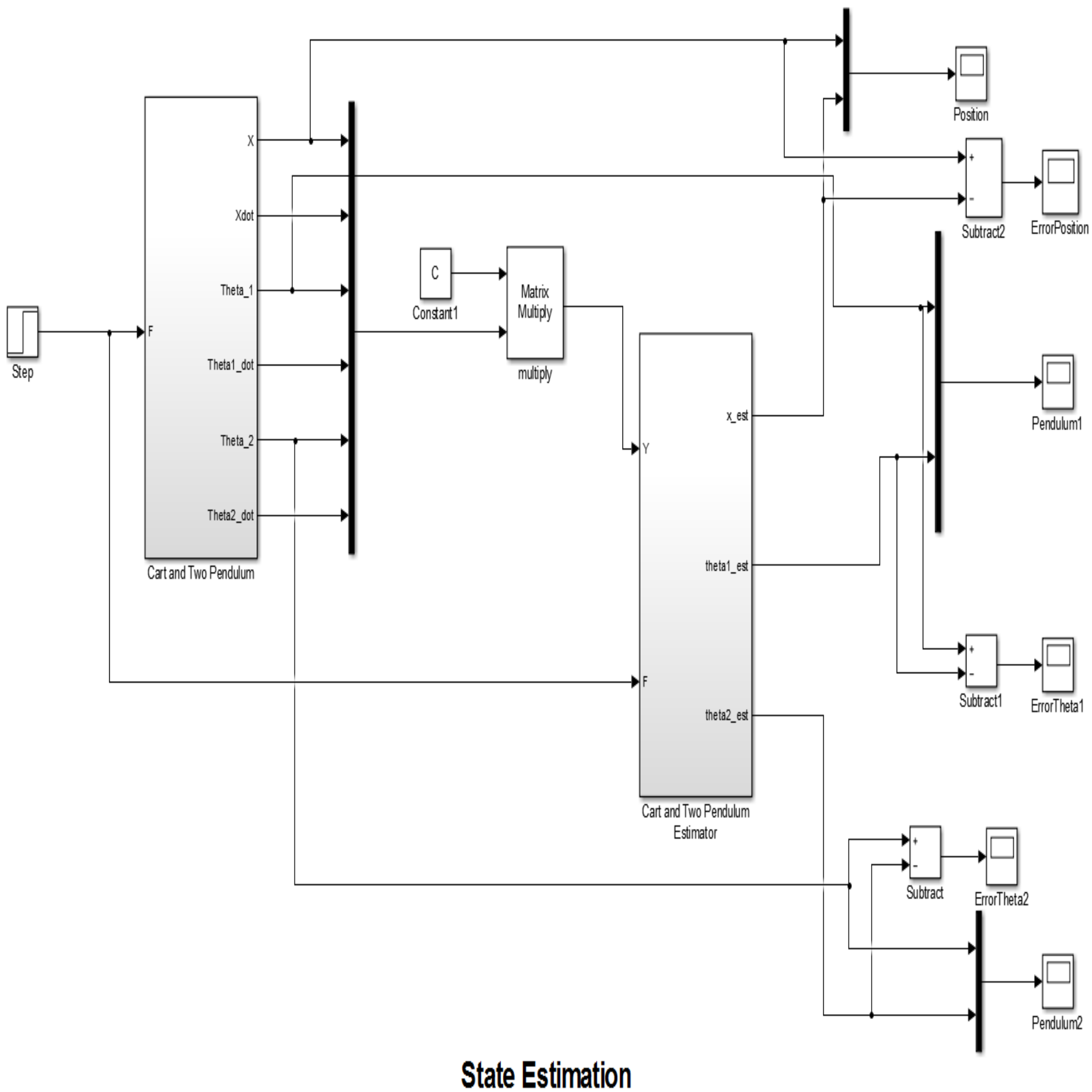


Figure 21: Complete State Estimation Model

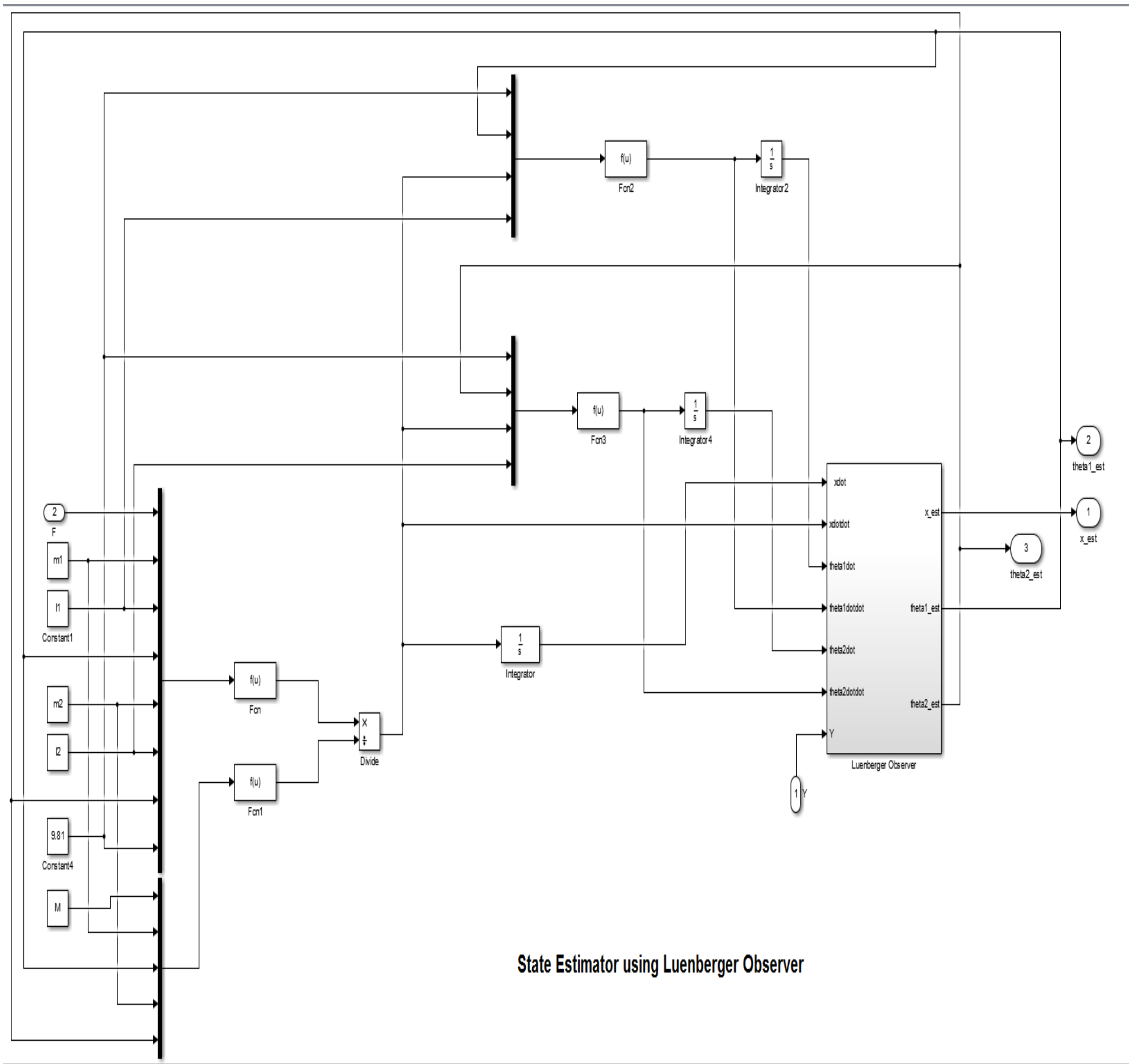


Figure 22: State Estimation Model

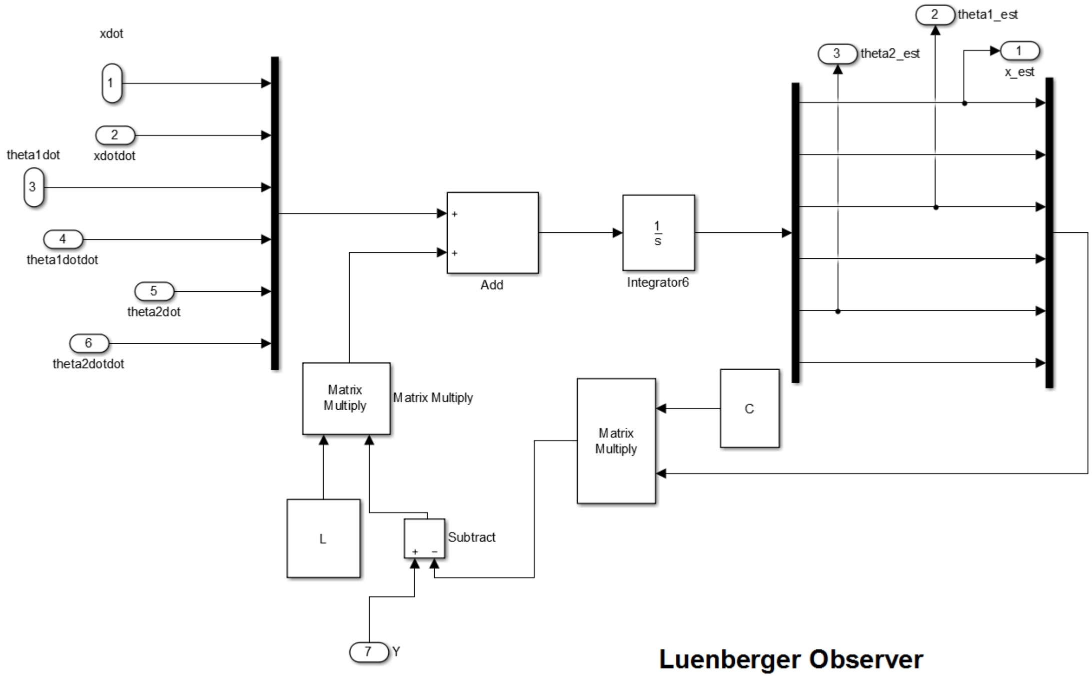


Figure 23: LuenbergerObserver

B.4 LQG controller on the linear model

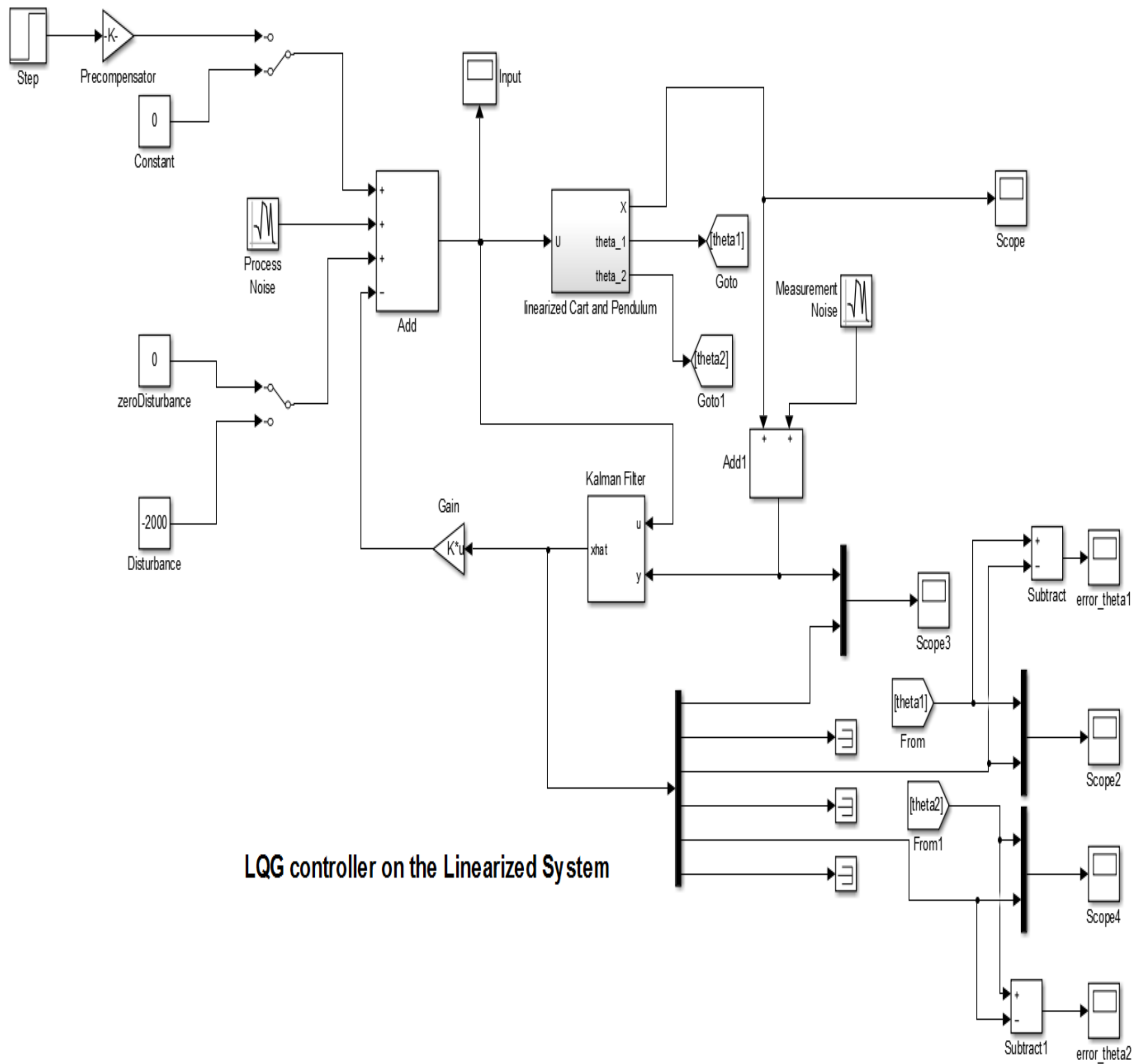


Figure 24: Linear Model of the Cart and pendulum

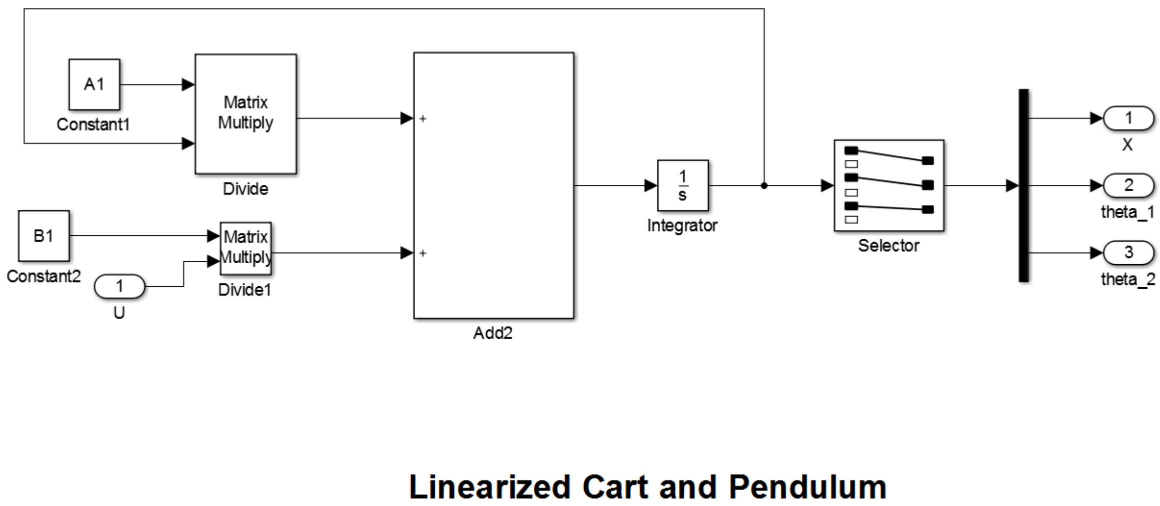


Figure 25: Linear Model of the Cart and pendulum

B.5 LQG controller on the Non-Linear model

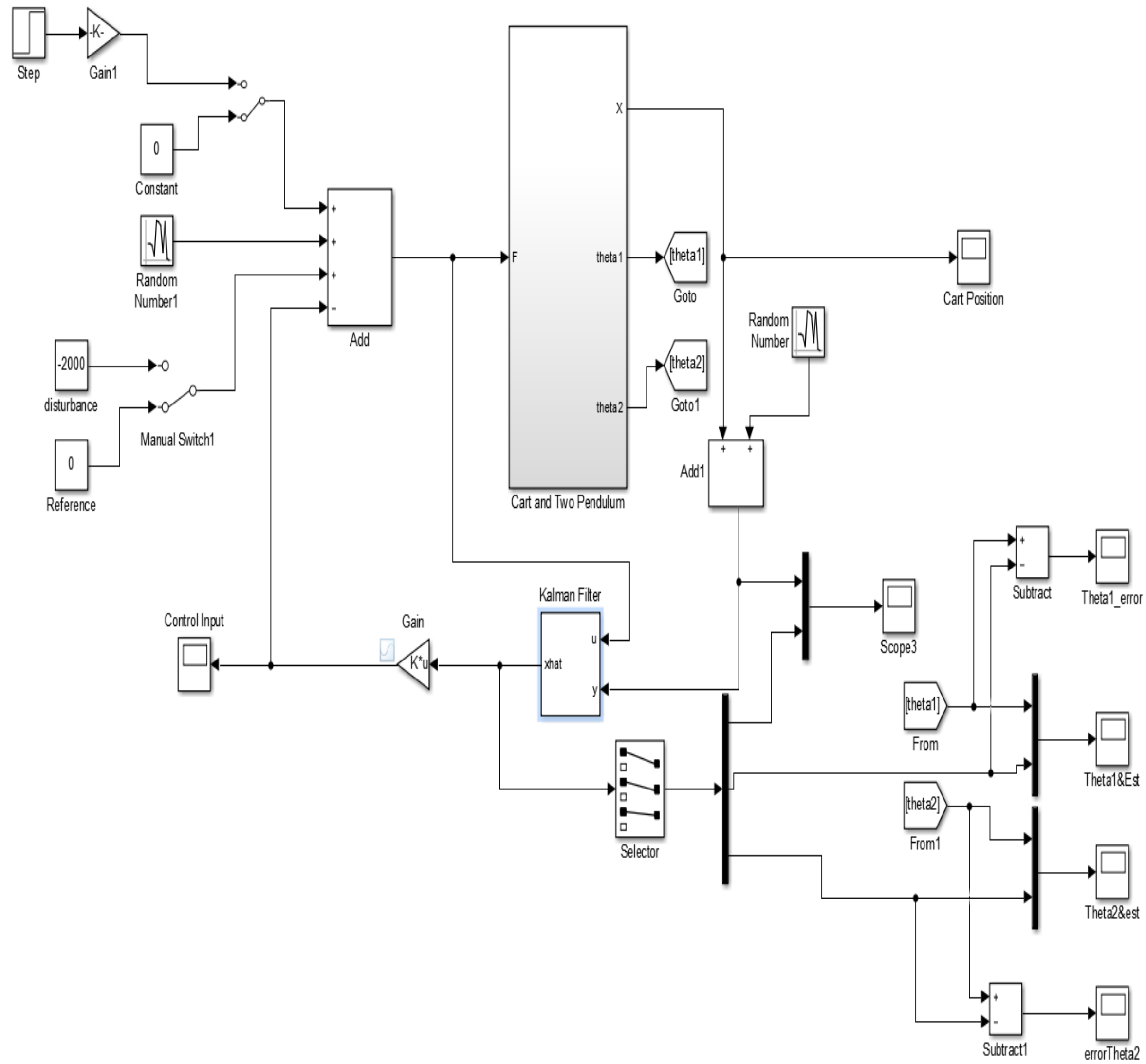


Figure 26: Non-linear implementation of the LQG controller

B.6 Final Design

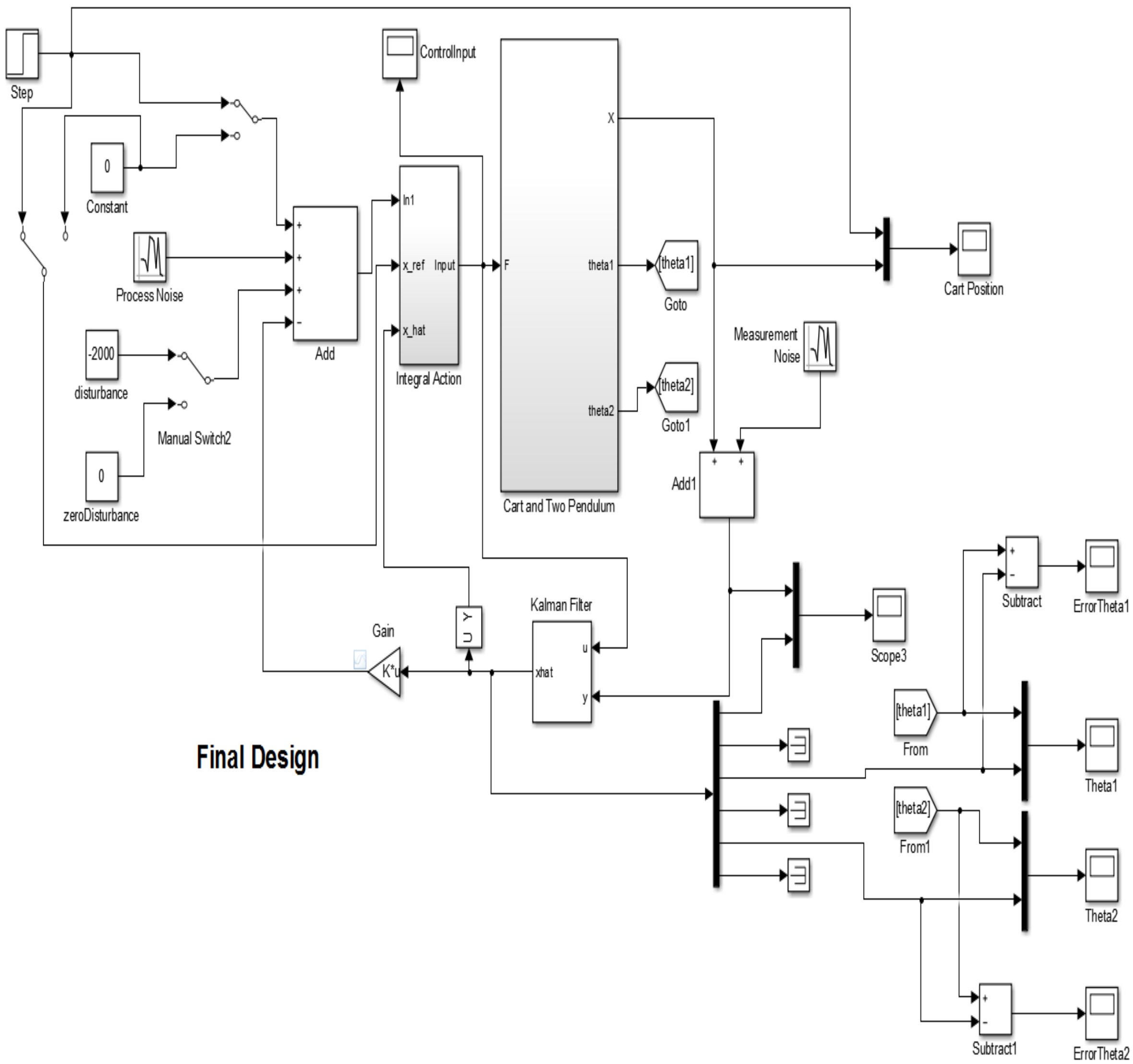


Figure 27: LQG controller with Integral Action applied to the non-linear system

B.7 Simscape Model of the Cart and two Pendulum system

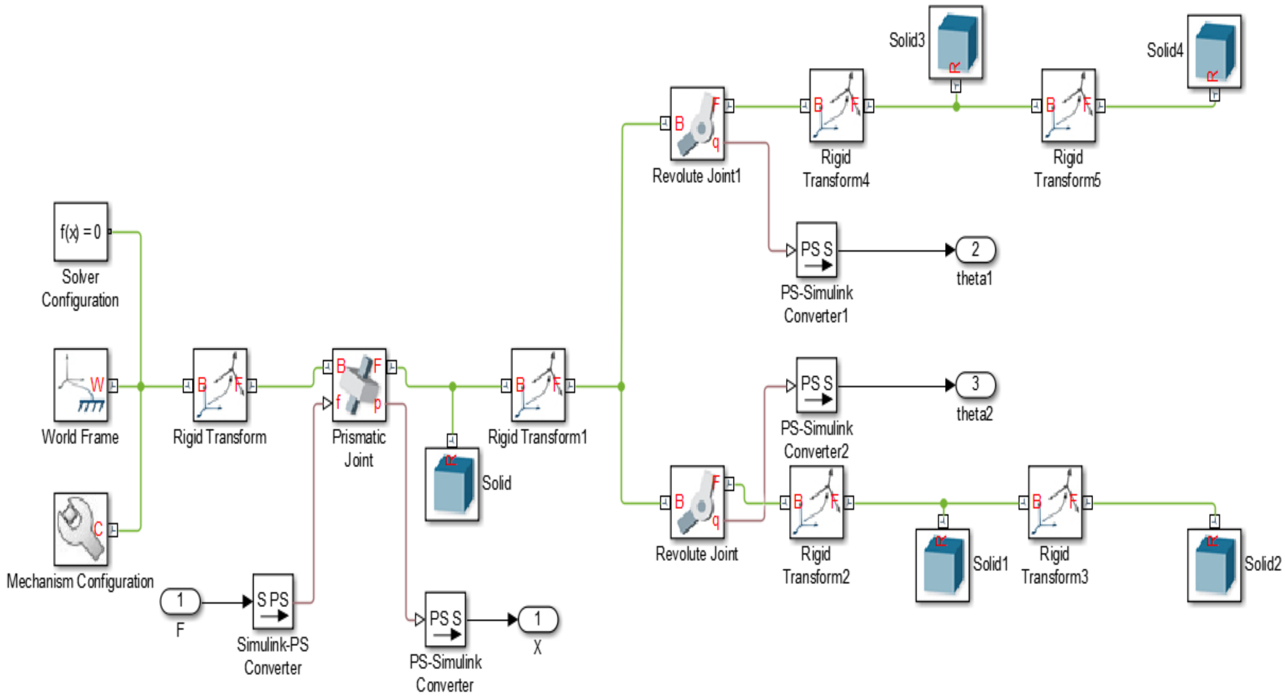


Figure 28: Simscape Model of the Cart and Two Pendulum

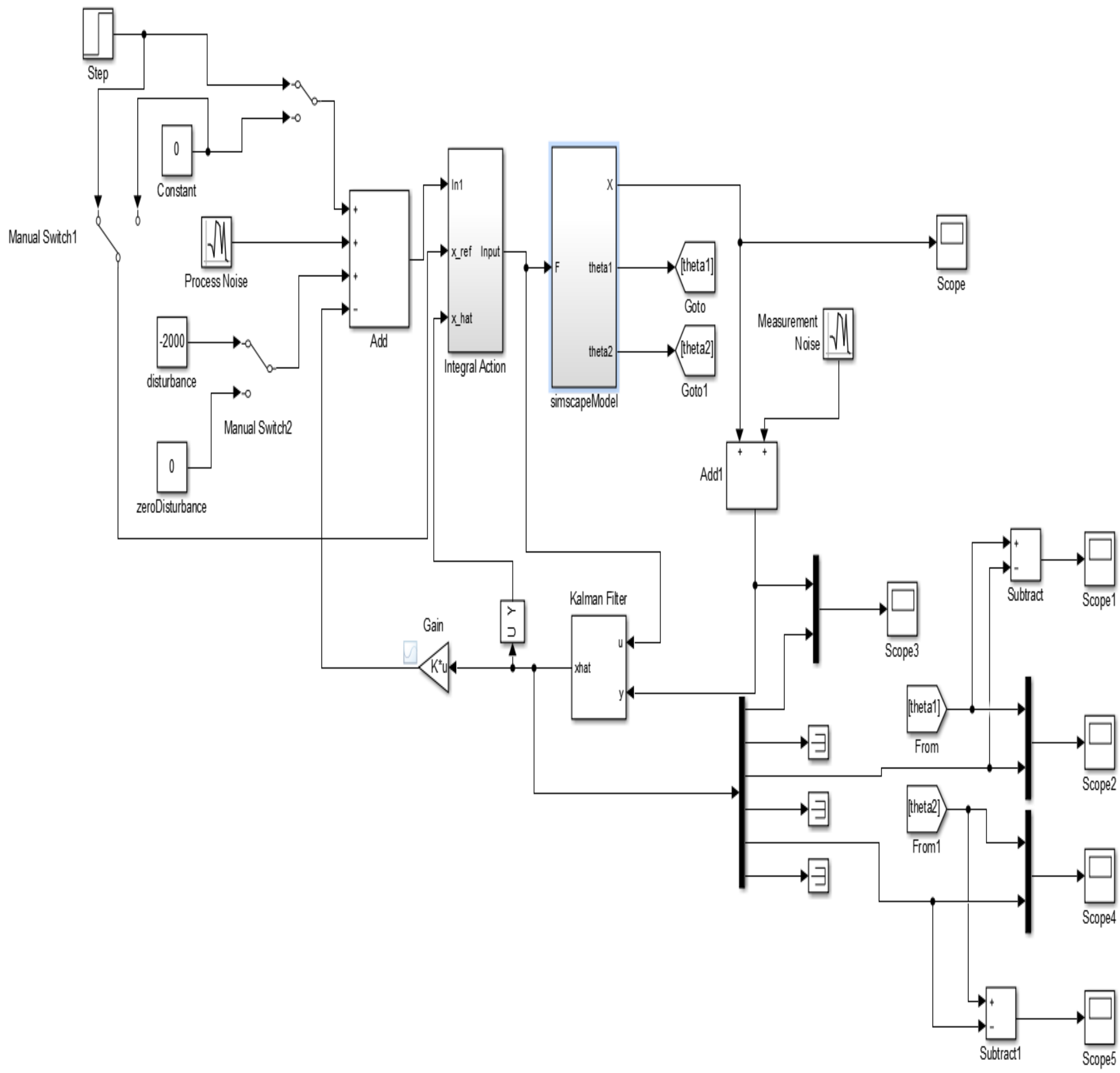


Figure 29: Final Simulation with the Simscape Model

Script to evaluate the constant A and B matrices of AX + BU

Evaluate the Kinetic and Potential Energy

```
T = M * (x'[t])^2 / 2 +
   m1 * ((x'[t] - l1 * θ1'[t] * Cos[θ1[t]])^2 + (l1 * θ1'[t] * Sin[θ1[t]])^2) / 2 +
   m2 * ((x'[t] - l2 * θ2'[t] * Cos[θ2[t]])^2 + (l2 * θ2'[t] * Sin[θ2[t]])^2) / 2;
V = m1 * g * l1 * (1 - Cos[θ1[t]]) + m2 * g * l2 * (1 - Cos[θ2[t]]);
```

Crank the Lagrangian Equation for each generalized Co-ordinate

```
firstTerm = Simplify[D[D[T, x'[t]], t] - D[T, x[t]] + D[V, x[t]]];
secondTerm = Simplify[D[D[T, θ1'[t]], t] - D[T, θ1[t]] + D[V, θ1[t]]];
thirdTerm = Simplify[D[D[T, θ2'[t]], t] - D[T, θ2[t]] + D[V, θ2[t]]];
FullSimplify[Solve[firstTerm == F, x''[t]]]
FullSimplify[Solve[secondTerm == 0, θ1''[t]]]
FullSimplify[Solve[thirdTerm == 0, θ2''[t]]]
```

$$\left\{\left\{x''[t] \to \frac{1}{M + m_1 + m_2}\left(F + l_1 m_1 \left(-\mathrm{Sin}[\theta_1[t]]\, \theta_1'[t]^2 + \mathrm{Cos}[\theta_1[t]]\, \theta_1''[t]\right) + l_2 m_2 \left(-\mathrm{Sin}[\theta_2[t]]\, \theta_2'[t]^2 + \mathrm{Cos}[\theta_2[t]]\, \theta_2''[t]\right)\right)\right\}\right\}$$

$$\left\{\left\{\theta_1''[t] \to \frac{-g\,\mathrm{Sin}[\theta_1[t]] + \mathrm{Cos}[\theta_1[t]]\, x''[t]}{l_1}\right\}\right\}$$

$$\left\{\left\{\theta_2''[t] \to \frac{-g\,\mathrm{Sin}[\theta_2[t]] + \mathrm{Cos}[\theta_2[t]]\, x''[t]}{l_2}\right\}\right\}$$

Evaluate the Non-Linear State Space function f(X,U)

```
K = (-(m₁ * Sin[θ₁[t]] * Cos[θ₁[t]] + m₂ * Sin[θ₂[t]] * Cos[θ₂[t]]) * g -
      m₁ * l₁ * (θ₁'[t])^2 * Sin[θ₁[t]] - m₂ * l₂ * (θ₂'[t])^2 * Sin[θ₂[t]] + F) /
    (M + m₁ * Sin[θ₁[t]]^2 + m₂ * Sin[θ₂[t]]^2);
f = {{x'[t]}, {K}, {θ₁'[t]}, {(-g * Sin[θ₁[t]] + Cos[θ₁[t]] * K) / l₁},
    {θ₂'[t]}, {(-g * Sin[θ₂[t]] + Cos[θ₂[t]] * K) / l₂}};
X = {{x[t]}, {x'[t]}, {θ₁[t]}, {θ₁'[t]}, {θ₂[t]}, {θ₂'[t]}};
```

Evaluate the Jacobian at the equilibrium position X = 0

```
A = D[f, {Transpose[X]}];
B = D[f, F];
A_c =
 Simplify[A /. {x[t] → 0, x'[t] → 0, θ₁[t] → 0, θ₁'[t] → 0, θ₂[t] → 0 , θ₂'[t] → 0}]
B_f = Simplify[
  B /. {x[t] → x, x'[t] → xdot, θ₁[t] → 0, θ₁'[t] → 0, θ₂[t] → 0 , θ₂'[t] → 0}]
```

$$\left\{\{\{\{0,1,0,0,0,0\}\}\},\ \left\{\left\{\left\{0,0,-\frac{g\,m_1}{M},0,-\frac{g\,m_2}{M},0\right\}\right\}\right\},\ \{\{\{0,0,0,1,0,0\}\}\},\ \left\{\left\{\left\{0,0,-\frac{g\,(M+m_1)}{M\,l_1},0,-\frac{g\,m_2}{M\,l_1},0\right\}\right\}\right\},\ \{\{\{0,0,0,0,0,1\}\}\},\ \left\{\left\{\left\{0,0,-\frac{g\,m_1}{M\,l_2},0,-\frac{g\,(M+m_2)}{M\,l_2},0\right\}\right\}\right\}\right\}$$

$$\left\{\{0\},\left\{\frac{1}{M}\right\},\{0\},\left\{\frac{1}{M\,l_1}\right\},\{0\},\left\{\frac{1}{M\,l_2}\right\}\right\}$$

References

[1] Control Tutorials
`http://http://ctms.engin.umich.edu/CTMS/`