PROJECT REPORT

ON

CONTROLLER DESIGN AND ANALYSIS OF DOUBLE PENDULUM CRANE

Srujan Panuganti
116302319

Vamshi Kumar Bogoju
116301404

December, 2018

Contents

Chapter 1

Problem Statement

ENPM 667 Final Project

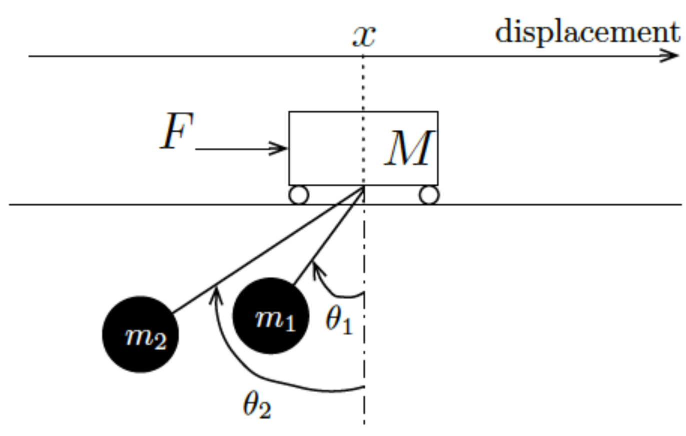
University of Maryland, College Park

Due Date: December 17 before Final Exam

INSTRUCTIONS

- Please include all relevant calculations, simulation plots, simulation schemes and algorithms as part of your solutions. Please submit your solutions in a single .pdf file.
- Online submission is only allowed for students in the online and remote sections of the course.
- Students in Sections 0101 and 0201 need to submit a hard copy in class and also need to email a copy of their solutions to the TAs.
- This project is worth 20% of the course grade.
- Two students are allowed (not required) to work as a group on this project. If you decide to work as a group please submit one technical report with both of your names clearly written at the top.

First Component (100 points): Consider a crane that moves along an one-dimensional track. It behaves as a frictionless cart with mass M actuated by an external force F that constitutes the input of the system. There are two loads suspended from cables attached to the crane. The loads have mass m_1 and m_2, and the lengths of the cables are l_1 and l_2, respectively. The following figure depicts the crane and associated variables used throughout this project.



A) (25 points) Obtain the equations of motion for the system and the corresponding nonlinear state-space representation.

B) (25 points) Obtain the linearized system around the equilibrium point specified by $x = 0$ and $\theta_1 = \theta_2 = 0$. Write the state-space representation of the linearized system.

C) (25 points) Obtain conditions on M, m_1, m_2, l_1, l_2 for which the linearized system is controllable.

D) (25 points) Choose $M = 1000Kg$, $m_1 = m_2 = 100Kg$, $l_1 = 20m$ and $l_2 = 10m$. Check that the system is controllable and obtain an LQR controller. Simulate the resulting response to initial conditions when the controller is applied to the linearized system and also to the original nonlinear system. Adjust the parameters of the LQR cost until you obtain a suitable response. Use Lyapunov's indirect method to certify stability (locally or globally) of the closed-loop system.

Second Component (100 points): Consider the parameters selected in C) above.

E) Suppose that you can select the following output vectors: $x(t)$, $(\theta_1(t), \theta_2(t))$, $(x(t), \theta_2(t))$ or $(x(t), \theta_1(t), \theta_2(t))$. Determine for which output vectors the linearized system is observable.

F) Obtain your "best" Luenberger observer for each one of the output vectors for which the system is observable and simulate its response to initial conditions and unit step input. The simulation should be done for the observer applied to both the linearized system and the original nonlinear system.

G) Design an output feedback controller for your choice of the "smallest" output vector. Use the LQG method and apply the resulting output feedback controller to the original nonlinear system. Obtain your best design and illustrate its performance in simulation. How would you reconfigure your controller to asymptotically track a constant reference on x ? Will your design reject constant force disturbances applied on the cart ?

Chapter 2

System Model

2.1 Equations of motion for the system

Kinetic energy for the mass M,

$$T_1 = \frac{M}{2}\dot{x}^2, \tag{2.1}$$

Kinetic energy for mass m_1,

$$T_2 = \frac{m_1}{2}\left[\dot{x} - l_1\dot{\theta_1(t)}\cos\theta_1(t)\right]^2 + \frac{m_1}{2}\left[l_1\dot{\theta_2(t)}\sin\theta_2(t)\right]^2, \tag{2.2}$$

Kinetic energy for mass m_2,

$$T_3 = \frac{m_2}{2}\left[\dot{x} - l_2\dot{\theta_2(t)}\cos\theta_2(t)\right]^2 + \frac{m_2}{2}\left[l_2\dot{\theta_2(t)}\sin\theta_2(t)\right]^2 \tag{2.3}$$

Therefore, total Kinetic energy is given by,

$$\begin{aligned} T = T_1 + T_2 + T_3 &= \frac{M}{2}\dot{x}^2 + \frac{m_1}{2}\left[\dot{x} - l_1\dot{\theta_1(t)}\cos\theta_1(t)\right]^2 + \frac{m_1}{2}\left[l_1\dot{\theta_2(t)}\sin\theta_2(t)\right]^2 \\ &+ \frac{m_2}{2}\left[\dot{x} - l_2\dot{\theta_2(t)}\cos\theta_2(t)\right]^2 + \frac{m_2}{2}\left[l_2\dot{\theta_2(t)}\sin\theta_2(t)\right]^2 \end{aligned} \tag{2.4}$$

Potential Energy for the mass $M = 0$
Potential Energy for the mass $m1 = -m_1 g l_1 \cos\theta_1$
Potential Energy for the mass $m2 = -m_2 g l_2 \cos\theta_2$
Total potential energy is given be,

$$P = -m_1 g l_1 \cos\theta_1 - m_2 g l_2 \cos\theta_2 \tag{2.5}$$

2.2 Lagrangian Function

Now the Langrange is given by

$$\begin{aligned}
\mathscr{L} &= T - P && (2.6)\\
&= \frac{M}{2}\dot{x}^2 + \frac{m_1}{2}\left[\dot{x} - l_1\dot{\theta_1}(t)\cos\theta_1(t)\right]^2 + \frac{m_1}{2}\left[l_1\dot{\theta_2}(t)\sin\theta_2(t)\right]^2 \\
&+ \frac{m_2}{2}\left[\dot{x} - l_2\dot{\theta_2}(t)\cos\theta_2(t)\right]^2 + \frac{m_2}{2}\left[l_2\dot{\theta_2}(t)\sin\theta_2(t)\right]^2 + m_1 g l_1\cos\theta_1 + m_2 g l_2\cos\theta_2 && (2.7)
\end{aligned}$$

The Euler - Lagrangian equations are given as

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial\mathscr{L}}{\partial\dot{x}(t)}\right) - \frac{\partial\mathscr{L}}{\partial x(t)} &= F, && (2.8)\\
\frac{d}{dt}\left(\frac{\partial\mathscr{L}}{\partial\dot{\theta_1}(t)}\right) - \frac{\partial\mathscr{L}}{\partial\theta_1(t)} &= 0, && (2.9)\\
\frac{d}{dt}\left(\frac{\partial\mathscr{L}}{\partial\dot{\theta_2}(t)}\right) - \frac{\partial\mathscr{L}}{\partial\theta_2(t)} &= 0 && (2.10)
\end{aligned}$$

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial\mathscr{L}}{\partial\dot{x}(t)}\right) - \frac{\partial\mathscr{L}}{\partial x(t)} = F &= M\ddot{x(t)} + \frac{m_1}{2}\left[2\ddot{x(t)} - 2l_1\ddot{\theta_1}(t)\cos\theta_1(t) + 2l_1\dot{\theta_1}(t)\sin\theta_1(t)\dot{\theta_1}(t)\right]\\
&+ \frac{m_2}{2}\left[2\ddot{x(t)} - 2l_2\ddot{\theta_2}(t)\cos\theta_2(t) + 2l_2\dot{\theta_2}(t)\sin\theta_2(t)\dot{\theta_2}(t)\right] && (2.11)
\end{aligned}$$

The above equation can then be re-written as follows,

$$\ddot{x(t)} = \frac{F + m_1 l_1\left(\cos\theta_1(t)\ddot{\theta_1}(t) - \dot{\theta_1}(t)^2\sin\theta_1(t)\right) + m_2 l_2\left(\cos\theta_2(t)\ddot{\theta_2}(t) - \dot{\theta_2}(t)^2\sin\theta_2(t)\right)}{M + m_1 + m_2} \quad (2.12)$$

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial\mathscr{L}}{\partial\dot{\theta_1}(t)}\right) - \frac{\partial\mathscr{L}}{\partial\theta_1(t)} = 0 &= m_1 l_1^2\ddot{\theta_1}(t) - m_1 l_1\cos\theta_1(t)\ddot{x(t)} + m_1 l_1\dot{x(t)}\sin\theta_1(t)\dot{\theta_1}(t)\\
&+ m_1 l_1^2\dot{\theta_1}(t)^2\cos\theta_1(t)\sin\theta_1(t) - m_1 l_1\dot{\theta_1}(t)\sin\theta_1(t)\dot{x(t)}\\
&- m_1 l_1^2\dot{\theta_1}(t)^2\cos\theta_1(t)\sin\theta_1(t) + m_1 g l_1\sin\theta_1(t) && (2.13)
\end{aligned}$$

after canceling out few terms and re-writing the equation we obtain,

$$\ddot{\theta_1}(t) = \frac{\cos\theta_1(t)\ddot{x(t)} - g\sin\theta_1(t)}{l_1}, \quad (2.14)$$

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial\mathscr{L}}{\partial\dot{\theta_2}(t)}\right) - \frac{\partial\mathscr{L}}{\partial\theta_2(t)} = 0 &= m_2 l_2^2\ddot{\theta_2}(t) - m_2 l_2\cos\theta_2(t)\ddot{x(t)} + m_2 l_2\dot{x(t)}\sin\theta_2(t)\dot{\theta_2}(t)\\
&+ m_2 l_2^2\dot{\theta_2}(t)^2\cos\theta_2(t)\sin\theta_2(t) - m_2 l_2\dot{\theta_2}(t)\sin\theta_2(t)\dot{x(t)}\\
&- m_2 l_2^2\dot{\theta_2}(t)^2\cos\theta_2(t)\sin\theta_2(t) + m_2 g l_2\sin\theta_2(t) && (2.15)
\end{aligned}$$

after canceling out few terms and re-writing the equation we obtain,

$$\ddot{\theta_2}(t) = \frac{\cos\theta_2(t)\ddot{x(t)} - g\sin\theta_2(t)}{l_2}, \quad (2.16)$$

2.3 Representation in statespace

The state vector can be written as

$$\dot{X(t)} = \begin{bmatrix} \dot{x(t)} \\ \ddot{x(t)} \\ \dot{\theta_1(t)} \\ \ddot{\theta_1(t)} \\ \dot{\theta_2(t)} \\ \ddot{\theta_2(t)} \end{bmatrix} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \end{bmatrix} \; while \quad X = \begin{bmatrix} x(t) \\ \dot{x(t)} \\ \theta_1(t) \\ \dot{\theta_1(t)} \\ \theta_2(t) \\ \dot{\theta_2(t)} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}, \; Now \;\; Let \dot{X(t)} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} \tag{2.17}$$

from above we can write

$$f_1 = \dot{x_1} = x_2 \tag{2.18}$$

$$f_3 = \dot{x_3} = x_4 \tag{2.19}$$

$$f_5 = \dot{x_5} = x_6 \tag{2.20}$$

$$\tag{2.21}$$

By substituting equations 2.14 and 2.16 in the equation 2.12 and substituting the sate variables we obtain,

$$f_2 = \ddot{x(t)} = \dot{x_2} = \frac{F - m_1 g \cos x_3 \sin x_3 - m_1 l_1 x_4^2 \sin x_3 - m_2 g \cos x_5 \sin x_5 - m_2 l_2 x_6^2 \sin x_5}{M + m_1 \sin x_3{}^2 + m_2 \sin x_5{}^2} \tag{2.22}$$

we can also write the below equations by substituting the state variables in the equation 2.14 and 2.16,

$$f_4 \;=\; \dot{x_4} = \frac{\dot{x_2} \cos x_3 - g \sin x_3}{l_1} \tag{2.23}$$

$$f_6 \;=\; \dot{x_6} = \frac{\dot{x_2} \cos x_5 - g \sin x_5}{l_2} \tag{2.24}$$

$$\tag{2.25}$$

Chapter 3

Linearization of the non linear system

Now we need to linearize the above equations in order to represent the state space.
The A_f matrix is obtained by taking the gradient of $X\dot(t)$ with respect to the state variables

$$A_f = \nabla F|_{(0,0)} \tag{3.1}$$

$$A_f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} & \frac{\partial f_1}{\partial x_5} & \frac{\partial f_1}{\partial x_6} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} & \frac{\partial f_2}{\partial x_5} & \frac{\partial f_2}{\partial x_6} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} & \frac{\partial f_3}{\partial x_5} & \frac{\partial f_3}{\partial x_6} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} & \frac{\partial f_4}{\partial x_5} & \frac{\partial f_4}{\partial x_6} \\ \frac{\partial f_5}{\partial x_1} & \frac{\partial f_5}{\partial x_2} & \frac{\partial f_5}{\partial x_3} & \frac{\partial f_5}{\partial x_4} & \frac{\partial f_5}{\partial x_5} & \frac{\partial f_5}{\partial x_6} \\ \frac{\partial f_6}{\partial x_1} & \frac{\partial f_6}{\partial x_2} & \frac{\partial f_6}{\partial x_3} & \frac{\partial f_6}{\partial x_4} & \frac{\partial f_6}{\partial x_5} & \frac{\partial f_6}{\partial x_6} \end{bmatrix}_{(0,0)} \tag{3.2}$$

The matrix B_f is given by performing the gradient of the State vector with respect to the input F

$$B_f = \begin{bmatrix} \frac{\partial f1}{\partial F} \\ \frac{\partial f2}{\partial F} \\ \frac{\partial f3}{\partial F} \\ \frac{\partial f4}{\partial F} \\ \frac{\partial f5}{\partial F} \\ \frac{\partial f6}{\partial F} \end{bmatrix} \tag{3.3}$$

computing the above partial derivatives we obtain,

$$A_f = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{gm_1}{M} & 0 & -\frac{gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{g(\frac{m_1}{M}+1)}{l_1} & 0 & \frac{-m_2 g}{l_1 M} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-m_1 g}{l_2 M} & 0 & -\frac{g(\frac{m_2}{M}+1)}{l_2} & 0 \end{bmatrix}, B_f = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{bmatrix}$$

Let

3.1 Controllability check

$$R = \begin{bmatrix} B_f & A_f B_f & A_f^2 B_f & A_f^3 B_f & A_f^4 B_f & A_f^5 B_f \end{bmatrix}$$

then,

$$\begin{aligned}
R &= \begin{bmatrix} 0 & \frac{1}{M} & 0 & \sigma_6 & 0 & \sigma_3 \\ \frac{1}{M} & 0 & \sigma_6 & 0 & \sigma_3 & 0 \\ 0 & \frac{1}{Ml_1} & 0 & \sigma_4 & 0 & \sigma_1 \\ \frac{1}{Ml_1} & 0 & \sigma_4 & 0 & \sigma_1 & 0 \\ 0 & \frac{1}{Ml_2} & 0 & \sigma_5 & 0 & \sigma_2 \\ \frac{1}{Ml_2} & 0 & \sigma_5 & 0 & \sigma_2 & 0 \end{bmatrix} \\
&where \\
\sigma_1 &= \frac{\frac{g^2 m_2 (M+m_1)}{M^2} + \frac{g^2 l_1 m_2 (M+m_2)}{M^2 l_2}}{Ml_2} + \frac{\frac{g^2 (M+m_1)^2}{M^2 {l_1}^2} + \frac{g^2 l_1 l_2 m_1 m_2}{M^2}}{Ml_1} \\
\sigma_2 &= \frac{\frac{g^2 m_1 (M+m_2)}{M^2} + \frac{g^2 l_2 m_1 (M+m_1)}{M^2 l_1}}{Ml_1} + \frac{\frac{g^2 (M+m_2)^2}{M^2 {l_2}^2} + \frac{g^2 l_1 l_2 m_1 m_2}{M^2}}{Ml_2} \\
\sigma_3 &= \frac{g^2 \left({l_1}^3 l_2 m_1 m_2 + {l_1}^2 {m_2}^2 + M {l_1}^2 m_2 + l_1 {l_2}^3 m_1 m_2 + {l_2}^2 {m_1}^2 + M {l_2}^2 m_1\right)}{M^3 {l_1}^2 {l_2}^2} \\
\sigma_4 &= -\frac{g\left(m_2 {l_1}^3 + M l_2 + l_2 m_1\right)}{M^2 {l_1}^2 l_2} \\
\sigma_5 &= -\frac{g\left(m_1 {l_2}^3 + M l_1 + l_1 m_2\right)}{M^2 l_1 {l_2}^2} \\
\sigma_6 &= -\frac{g(l_1 m_2 + l_2 m_1)}{M^2 l_1 l_2}
\end{aligned}$$

$rank[R] = 6$ computed using MATLAB, see Appendix. The rank of R in 6, Hence the system is controllable for the given values if M, m_1, m_2, l_1, l_2

3.2 Conditions for Controllability

To check the controllability of matrix R, the rank is calculated. if the rank of R is full rank, i.e., Determinant of R is not equal to zero.
For finding the conditions for controllability, we find the determinatant of the R Matrix. After simplifying the determinant, we obtain

$$det = -\frac{g^6(l_1 - l_2)^2}{(Ml_1 l_2)^6} \tag{3.4}$$

when l_1 not equal to l_2 then the determinant not equal to zero, thus the R matrix is controllable.

3.3 Design of Linear Quadratic Regulator Controller

Using the given values for $M = 1000Kg, m_1 = 100Kg, m_2 = 100Kg, l_1 = 20m, l_2 = 10m$ and assuming the $g = 9.81m/sec$, we obtain the A_f and B_f matrices as shown in the appendix
The LQR Controller is designed in MATLAB for both the linear and non linear system for different Initial conditions, refer Appendix

Chapter 4

Design of Observers and Linear Quadratic Gaussian

4.1 Check for observability

The observebility check for the output vectors are performed and has been concluded that $(\theta_1(t), \theta_2(t))$ is not observeble. all other output vectorss are observable
See MATLAB code for the same in Appendix.

4.2 Design of Luenberger Observer

The Luenberger observer for the above mentioned output vectors has been designed and applied to the original nonlinear system also. See the MATLAB code in the appendix for the same.

4.3 Design of Linear Quadratic Gaussian

The Linear Quadratic Gaussian for the linearized system is obtained. See the MATLAB code for the same

Chapter 5

Appendix

```
%syms M m1 m2 l1 l2 g
```

```
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
```

A_f and B_f matrices

```
A_f =[0 1 0 0 0 0;0 0 (-m1*g)/M 0 (-m2*g)/M 0;0 0 0 1 0 0;0 0 -(g/l1)*((m1+M)/M) 0 (-m2*g)/M*l1
```

```
A_f = 6×6
         0    1.0000         0         0         0         0
         0         0   -0.9810         0   -0.9810         0
         0         0         0    1.0000         0         0
         0         0   -0.5396         0  -19.6200         0
         0         0         0         0         0    1.0000
         0         0   -9.8100         0   -1.0791         0
```

```
B_f =[0;1/M;0;1/(M*l1);0;1/(M*l2)]
```

B_f = 6×1

10^{-3} ×

```
         0
    1.0000
         0
    0.0500
         0
    0.1000
```

```
C = [1,0,0,0,0,0]
```

```
C = 1×6
     1     0     0     0     0     0
```

```
D = [0]
```

```
D = 0
```

```
eigs(A_f);

plant = ss(A_f,B_f,C,D)
```

```
plant =

  A =
             x1        x2        x3        x4        x5        x6
   x1         0         1         0         0         0         0
   x2         0         0    -0.981         0    -0.981         0
```

```
  x3        0        0        0        1        0        0
  x4        0        0  -0.5396        0   -19.62        0
  x5        0        0        0        0        0        1
  x6        0        0    -9.81        0   -1.079        0

 B =
          u1
  x1       0
  x2   0.001
  x3       0
  x4   5e-05
  x5       0
  x6  0.0001

 C =
      x1  x2  x3  x4  x5  x6
  y1   1   0   0   0   0   0

 D =
      u1
  y1   0

Continuous-time state-space model.
```

```
Plant_poles = pole(plant)
```

```
Plant_poles = 6×1 complex
   0.0000 + 0.0000i
   0.0000 + 0.0000i
  -3.6148 + 0.0000i
  -0.0000 + 3.8322i
  -0.0000 - 3.8322i
   3.6148 + 0.0000i
```

Controllability Matrix and rank condition

```
Co = ctrb(plant)
```

```
Co = 6×6
         0    0.0010         0   -0.0001         0    0.0025
    0.0010         0   -0.0001         0    0.0025         0
         0    0.0001         0   -0.0020         0    0.0128
    0.0001         0   -0.0020         0    0.0128         0
         0    0.0001         0   -0.0006         0    0.0202
    0.0001         0   -0.0006         0    0.0202         0
```

```
Ob = obsv(plant)
```

```
Ob = 6×6
    1.0000         0         0         0         0         0
         0    1.0000         0         0         0         0
         0         0   -0.9810         0   -0.9810         0
         0         0         0   -0.9810         0   -0.9810
         0         0   10.1529         0   20.3058         0
         0         0         0   10.1529         0   20.3058
```

```
rank_Co = rank(Co)
```

```
rank_Co = 6
```

```
rank_Ob = rank(Ob)
```

```
rank_Ob = 6
```

Position of poles

```
rlocus(plant)
```

Initial conditions

```
I = [0;0;0;0;0;0];
```

LQR Controller Design

```
Q = [1200000000000 10 0 0 0 0;10 2 0 0 0 0;0 0 11000000000000 0 0 0 ; 0 0 0 0.1 0 0;0 0 0 0 140
R = 10;
eig_Q = eigs(Q);
N = [1;2;3;1;5;2];

Positive_Semi_Definite = [Q N;N' R] %this matrix must be positive-semi definite
```

```
Positive_Semi_Definite = 7×7
10^13 ×
    0.1200    0.0000         0         0         0         0    0.0000
    0.0000    0.0000         0         0         0         0    0.0000
         0         0    1.1000         0         0         0    0.0000
         0         0         0    0.0000         0         0    0.0000
         0         0         0         0    1.4000         0    0.0000
         0         0         0         0         0    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
```

```
[K2,S2,e2] = lqr(A_f+eye(size(A_f)),B_f,Q,R,N)
```

```
K2 = 1×6
10^7 ×
   -0.0763   -0.0589   -2.8052   -0.6839    3.0355    0.9735
S2 = 6×6
10^15 ×
    0.0023    0.0011    0.0360    0.0081   -0.0442   -0.0150
    0.0011    0.0006    0.0226    0.0048   -0.0245   -0.0089
    0.0360    0.0226    0.8753    0.1910   -0.9032   -0.3241
    0.0081    0.0048    0.1910    0.0429   -0.2012   -0.0703
   -0.0442   -0.0245   -0.9032   -0.2012    1.0050    0.3488
   -0.0150   -0.0089   -0.3241   -0.0703    0.3488    0.1250
e2 = 6×1 complex
 -13.5859 +13.6057i
```

```
 -13.5859 -13.6057i
  -4.5397 + 0.0000i
  -1.1921 + 3.6421i
  -1.1921 - 3.6421i
  -2.6311 + 0.0000i
```

Root Locus with LQR Controller

```
%rlocus(Plant_LQR)
```

```
A_cl = A_f-(B_f*K2);
Plant_LQR = ss(A_cl,B_f,C,D)
```

```
Plant_LQR =
 
  A = 
              x1          x2          x3          x4          x5          x6
   x1          0           1           0           0           0           0
   x2      762.8       588.8   2.805e+04        6839  -3.036e+04       -9735
   x3          0           0           0           1           0           0
   x4      38.14       29.44        1402         342       -1537      -486.7
   x5          0           0           0           0           0           1
   x6      76.28       58.88        2795       683.9       -3037      -973.5
 
  B = 
           u1
   x1       0
   x2   0.001
   x3       0
   x4   5e-05
   x5       0
   x6  0.0001
 
  C = 
       x1  x2  x3  x4  x5  x6
   y1   1   0   0   0   0   0
 
  D = 
       u1
   y1   0
 
Continuous-time state-space model.
```

```
step(Plant_LQR)
```

Luenberger Observer Design

```
%Desired Location of Poles

Poles_L_Obs = [-28;-30;-10;-6;-2;-4];
```

```
L = place(A_f',C',Poles_L_Obs)'
```

```
L = 6×1
10^5 ×
    0.0008
    0.0228
   -0.9791
   -3.8919
    0.6901
    2.2299
```

```
Luen_SYS = ss(A_f-L*C,B_f,C,D)
```

```
Luen_SYS =

  A =
              x1          x2          x3          x4          x5          x6
   x1        -80           1           0           0           0           0
   x2      -2278           0      -0.981           0      -0.981           0
   x3  9.791e+04           0           0           1           0           0
   x4  3.892e+05           0     -0.5396           0      -19.62           0
   x5 -6.901e+04           0           0           0           0           1
   x6  -2.23e+05           0       -9.81           0      -1.079           0

  B =
           u1
   x1       0
   x2   0.001
   x3       0
   x4   5e-05
   x5       0
   x6  0.0001

  C =
       x1  x2  x3  x4  x5  x6
   y1   1   0   0   0   0   0

  D =
       u1
   y1   0

Continuous-time state-space model.
```

```
step(Luen_SYS)
```

Check for obervability

```
C_Lo = ctrb(A_f',C')
```

```
C_Lo = 6×6
    1.0000         0         0         0         0         0
         0    1.0000         0         0         0         0
         0         0   -0.9810         0   10.1529         0
         0         0         0   -0.9810         0   10.1529
```

```
         0         0   -0.9810         0   20.3058         0
         0         0         0   -0.9810         0   20.3058
```

```
rank(C_Lo)
```

```
ans = 6
```

```
Ob_Lo = obsv(A_f,C)
```

```
Ob_Lo = 6×6
    1.0000         0         0         0         0         0
         0    1.0000         0         0         0         0
         0         0   -0.9810         0   -0.9810         0
         0         0         0   -0.9810         0   -0.9810
         0         0   10.1529         0   20.3058         0
         0         0         0   10.1529         0   20.3058
```

```
rank(Ob_Lo)
```

```
ans = 6
```

LQG Controller Design

```
At = [ A_f-B_f*K2              B_f*K2
       zeros(size(A_f))    A_f-L*C ]
```

```
At = 12×12
10^5 ×
         0    0.0000         0         0         0         0         0         0 ···
    0.0076    0.0059    0.2805    0.0684   -0.3036   -0.0973   -0.0076   -0.0059
         0         0         0    0.0000         0         0         0         0
    0.0004    0.0003    0.0140    0.0034   -0.0154   -0.0049   -0.0004   -0.0003
         0         0         0         0         0    0.0000         0         0
    0.0008    0.0006    0.0280    0.0068   -0.0304   -0.0097   -0.0008   -0.0006
         0         0         0         0         0         0   -0.0008    0.0000
         0         0         0         0         0         0   -0.0228         0
         0         0         0         0         0         0    0.9791         0
         0         0         0         0         0         0    3.8919         0
    ⋮
```

```
Bt = [    B_f
       zeros(size(B_f)) ]
```

```
Bt = 12×1
10^-3 ×
         0
    1.0000
         0
    0.0500
         0
    0.1000
         0
         0
```

```
        0
        0
     ⋮
```

```
Ct = [ C    zeros(size(C)) ]
```

```
Ct = 1×12
     1     0     0     0     0     0     0     0     0     0     0     0
```

LQG Step Response

```
step(ss(At,Bt,Ct,0))
```

```

```

For the outputs X(t) and theta_2(t) The observer is given as

```
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
```

A_f and B_f matrices

```
A_f =[0 1 0 0 0 0;0 0 (-m1*g)/M 0 (-m2*g)/M 0;0 0 0 1 0 0;0 0 -(g/l1)*((m1+M)/M) 0 (-m2*g)/M*l1
```

```
A_f = 6×6
         0    1.0000         0         0         0         0
         0         0   -0.9810         0   -0.9810         0
         0         0         0    1.0000         0         0
         0         0   -0.5396         0  -19.6200         0
         0         0         0         0         0    1.0000
         0         0   -9.8100         0   -1.0791         0
```

```
B_f =[0;1/M;0;1/(M*l1);0;1/(M*l2)]
```

B_f = 6×1

10^{-3} ×

```
         0
    1.0000
         0
    0.0500
         0
    0.1000
```

```
eigs(A_f)
```

```
ans = 6×1 complex
  -0.0000 + 3.8322i
  -0.0000 - 3.8322i
  -3.6148 + 0.0000i
   3.6148 + 0.0000i
   0.0000 + 0.0000i
   0.0000 + 0.0000i
```

```
C = [1 0 0 0 0 0;0 0 0 0 1 0;]
```

```
C = 2×6
     1     0     0     0     0     0
     0     0     0     0     1     0
```

```
D = [0;0]
```

```
D = 2×1
     0
     0
```

```
plant = ss(A_f,B_f,C,D);
Plant_poles = pole(plant);
```

Controllability Matrix and rank condition

```
Co = ctrb(plant);
rank_Co = rank(Co);
```

Initial conditions

```
I = [0;1;1;0;1;0]
```

```
I = 6×1
     0
     1
     1
     0
     1
     0
```

Luenberger Observer Design

```
%Luen_SYS = ss(A_f-L*C,B_f,C,D)
%step(Luen_SYS)
```

Luenberger Observer Design

Controllability Check - Rank condition of the observability matrix satisfies. Hence Observable

```
C_Lo = ctrb(A_f',C')
```

```
C_Lo = 6×12
    1.0000         0         0         0         0         0         0         0 ···
         0         0    1.0000         0         0         0         0         0
         0         0         0         0   -0.9810   -9.8100         0         0
         0         0         0         0         0         0   -0.9810   -9.8100
         0    1.0000         0         0   -0.9810   -1.0791         0         0
         0         0         0    1.0000         0         0   -0.9810   -1.0791
```

```
rak_ctrb1 = rank(C_Lo)
```

```
rak_ctrb1 = 6
```

```
Ob_Lo = obsv(A_f,C)
```

```
Ob_Lo = 12×6
    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   -9.8100         0   -1.0791         0
         0         0         0   -0.9810         0   -0.9810
         0         0         0   -9.8100         0   -1.0791
         0         0   10.1529         0   20.3058         0
         0         0   15.8790         0  193.6367         0
      ⋮
```

```
rank_Obsv1 = rank(Ob_Lo)
```

```
rank_Obsv1 = 6
```

```
pole(Plant_LQR)
```

```
ans = 6×1 complex
 -14.5859 +13.6057i
 -14.5859 -13.6057i
  -2.1921 + 3.6421i
  -2.1921 - 3.6421i
  -5.5397 + 0.0000i
  -3.6311 + 0.0000i
```

```
Poles_L_Obs = [-28;-30;-10;-6;-2;-4]
```

```
Poles_L_Obs = 6×1
   -28
   -30
   -10
    -6
    -2
    -4
```

```
L = place(A_f',C',Poles_L_Obs)'
```

```
L = 6×2
   34.2743    0.8932
  174.7915   36.8658
  -37.6817 -208.5662
  -47.2669 -228.2706
    1.2047   45.7257
   56.7906  537.4500
```

```
Luen_SYS = ss(A_f-L*C,B_f,C,D)
```

```
Luen_SYS =

  A =
             x1        x2        x3        x4        x5        x6
   x1    -34.27         1         0         0   -0.8932         0
   x2    -174.8         0    -0.981         0    -37.85         0
   x3     37.68         0         0         1     208.6         0
```

```
  x4    47.27        0  -0.5396        0    208.7        0
  x5   -1.205        0        0        0   -45.73        1
  x6   -56.79        0    -9.81        0   -538.5        0

B =
         u1
  x1      0
  x2  0.001
  x3      0
  x4  5e-05
  x5      0
  x6 0.0001

C =
      x1  x2  x3  x4  x5  x6
  y1   1   0   0   0   0   0
  y2   0   0   0   0   1   0

D =
      u1
  y1   0
  y2   0

Continuous-time state-space model.
```

```
step(Luen_SYS)
```

```
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
```

A_f and B_f matrices

```
A_f =[0 1 0 0 0 0;0 0 (-m1*g)/M 0 (-m2*g)/M 0;0 0 0 1 0 0;0 0 -(g/l1)*((m1+M)/M) 0 (-m2*g)/M*l1
```

```
A_f = 6×6
         0    1.0000         0         0         0         0
         0         0   -0.9810         0   -0.9810         0
         0         0         0    1.0000         0         0
         0         0   -0.5396         0  -19.6200         0
         0         0         0         0         0    1.0000
         0         0   -9.8100         0   -1.0791         0
```

```
B_f =[0;1/M;0;1/(M*l1);0;1/(M*l2)]
```

```
B_f = 6×1
10^-3 ×
         0
    1.0000
         0
    0.0500
         0
    0.1000
```

```
eigs(A_f)
```

```
ans = 6×1 complex
  -0.0000 + 3.8322i
  -0.0000 - 3.8322i
  -3.6148 + 0.0000i
   3.6148 + 0.0000i
   0.0000 + 0.0000i
   0.0000 + 0.0000i
```

```
C = [1 0 0 0 0 0;0 0 1 0 0 0;0 0 0 0 1 0;]
```

```
C = 3×6
     1     0     0     0     0     0
     0     0     1     0     0     0
     0     0     0     0     1     0
```

```
D = [0;0;0]
```

```
D = 3×1
     0
     0
     0
```

```
plant = ss(A_f,B_f,C,D);
Plant_poles = pole(plant);
```

Controllability Matrix and rank condition

```
Co = ctrb(plant);
Ob = obsv(plant);
rank_Co = rank(Co);
eig_Co = eigs(Co);
rank_Ob = rank(Ob);
%eig_Ob = eigs(Ob);
```

Initial conditions

```
I = [0;1;1;0;1;0];
```

Luenberger Observer Design

Observerbility check for x(t),theta_1(t), and theta_2(t)

```
C_Lo = ctrb(A_f',C')
```

```
C_Lo = 6×18
    1.0000         0         0         0         0         0         0         0 ···
         0         0         0    1.0000         0         0         0         0
         0    1.0000         0         0         0         0   -0.9810   -0.5396
         0         0         0         0    1.0000         0         0         0
         0         0    1.0000         0         0         0   -0.9810  -19.6200
         0         0         0         0         0    1.0000         0         0
```

```
rank(C_Lo)
```

```
ans = 6
```

```
Ob_Lo = obsv(A_f,C)
```

```
Ob_Lo = 18×6
    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   -0.5396         0  -19.6200         0
         0         0   -9.8100         0   -1.0791         0
         0         0         0   -0.9810         0   -0.9810
        ⋮
```

```
rank(Ob_Lo)
```

```
ans = 6
```

```
pole(Plant_LQR)
```

```
ans = 6×1 complex
 -14.5859 +13.6057i
 -14.5859 -13.6057i
  -2.1921 + 3.6421i
  -2.1921 - 3.6421i
  -5.5397 + 0.0000i
  -3.6311 + 0.0000i
```

```
Poles_L_Obs = [-28;-30;-10;-6;-2;-4]
```

```
Poles_L_Obs = 6×1
   -28
   -30
   -10
    -6
    -2
    -4
```

```
L = place(A_f',C',Poles_L_Obs)'
```

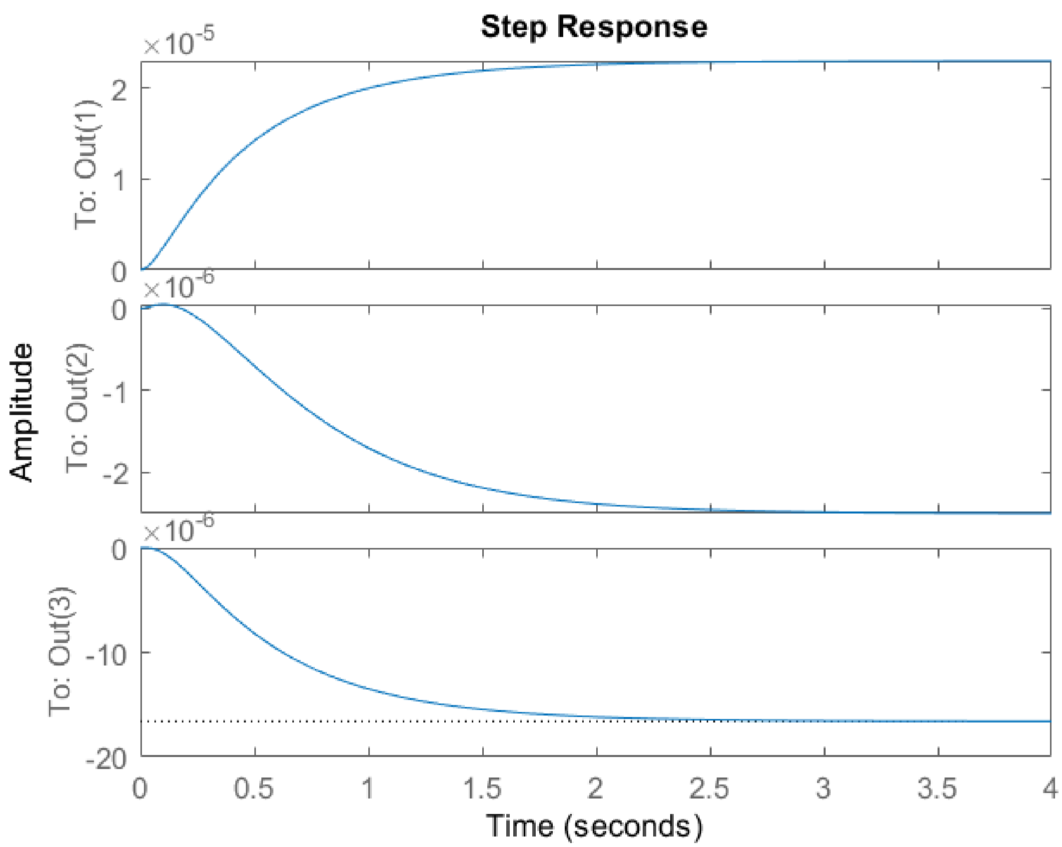
```
L = 6×3
   24.6012    3.0047   10.7179
   87.8210    8.7493   59.2856
    0.8441   31.8371   -2.7304
   -7.4693  119.8831  -51.1957
   11.4406   -5.0483   23.5617
   69.4268  -57.9228   96.6033
```

```
Luen_SYS = ss(A_f-L*C,B_f,C,D)
```

```
Luen_SYS =
 
  A = 
             x1        x2        x3        x4        x5        x6
   x1     -24.6         1    -3.005         0    -10.72         0
   x2    -87.82         0     -9.73         0    -60.27         0
   x3   -0.8441         0    -31.84         1      2.73         0
   x4     7.469         0    -120.4         0     31.58         0
   x5    -11.44         0     5.048         0    -23.56         1
   x6    -69.43         0     48.11         0    -97.68         0
 
  B = 
            u1
   x1        0
   x2    0.001
   x3        0
   x4    5e-05
   x5        0
   x6   0.0001
 
  C = 
       x1  x2  x3  x4  x5  x6
   y1   1   0   0   0   0   0
   y2   0   0   1   0   0   0
```

```
  y3   0   0   0   0   1   0

 D = 
      u1
  y1   0
  y2   0
  y3   0

Continuous-time state-space model.
```

```
step(Luen_SYS)
```




```
eigs(A_f' - C'*L')
```

```
ans = 6×1
  -30.0000
  -28.0000
  -10.0000
   -6.0000
   -4.0000
   -2.0000
```

```
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
I = [1;1;5;0;5;0];
%{
Try the following sets of values values
---> Initial COndition: I = [0;0;0;0;0;0]
---> I = [0;0;5;0;0;0]; t = 1:0.001:10;
---> I = [1;0;5;0;5;0]; t = 1:0.0001:10;
---> I = [1;0;5;0;5;0]; t = 1:0.001:10;
---> I = [0;0;5;0;10;0]; t = 1:0.001:10;
I = [1;1;5;0;5;0]; t = 1:0.0001:10;
%}
t = 1:0.0001:10;
[t,x] = ode45(@nlsys_solve,t,I);
plot(t,x,'linewidth',1);
title('response of the non-liear system');

%%Function Definition
function nlsys= nlsys_solve(t,x)
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;

A_f =[0 1 0 0 0 0;0 0 (-m1*g)/M 0 (-m2*g)/M 0;0 0 0 1 0 0;0 0 -(g/
l1)*((m1+M)/M) 0 (-m2*g)/M*l1 0;0 0 0 0 0 1;0 0 (-m1*g)/M*l2 0 -(g/
l2)*((m2+M)/M) 0];
B_f =[0;1/M;0;1/(M*l1);0;1/(M*l2)];
eigs(A_f);
C = [1,0,0,0,0,0];
%C = diag([1 1 1 1 1 1]);
%C1 = [1,0,0,0,0,0];
%D = [0;0;0;0;0;0];
D = [0];

Q = [1200000000 0 0 0 0 0;0 2 0 0 0 0;0 0 11000000000 0 0 0 ; 0 0 0
 0.1 0 0;0 0 0 0 14000000000 0;0 0 0 0 0 1];
R = 10;
eig_Q = eigs(Q);
N = [0;0;0;0;0;0];

[K2,S2,e2] = lqr(A_f+eye(size(A_f)),B_f,Q,R,N);

F = -K2*x;

nlsys = zeros(6,1);
```

```
nlsys(1)= x(2);

nlsys(2)= ((F) - m1*g*sind(x(3))*cosd(x(3)) - m1*l1*x(4)^2*sind(x(3))
 - m2*l2*x(6)^2*sind(x(5))  - m2*g*sind(x(5))*cosd(x(5)))/(M
+m1*(sind(x(3)))^2+ m2*(sind(x(5)))^2);

nlsys(3)= x(4);

nlsys(4)= (((F) - m1*g*sind(x(3))*cosd(x(3)) - m1*l1*x(4)^2*sind(x(3))
 - m2*l2*x(6)^2*sind(x(5))  - m2*g*sind(x(5))*cosd(x(5)))*cosd(x(3))/
(M+m1*(sind(x(3)))^2+ m2*(sind(x(5)))^2)*l1)-((g*sind(x(3)))/l1);

nlsys(5)= x(6);

nlsys(6)= (((F) - m1*g*sind(x(3))*cosd(x(3)) - m1*l1*x(4)^2*sind(x(3))
 - m2*l2*x(6)^2*sind(x(5))  - m2*g*sind(x(5))*cosd(x(5)))*cosd(x(5))/
(M+m1*(sind(x(3)))^2+ m2*(sind(x(5)))^2)*l2)-((g*sind(x(5)))/l1);

end
```

```
Warning: Failure at t=1.010437e+00.  Unable to meet integration
 tolerances
without reducing the step size below the smallest value allowed
 (3.552714e-15)
at time t.
```

Published with MATLAB® R2018a

```
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
I = [0;0;10;0;10;0];
%{
Try the following sets of values values
---> Initial COndition: I = [0;0;0;0;0;0]
---> I = [0;0;5;0;0;0]; t = 1:0.001:10;
---> I = [1;0;5;0;5;0]; t = 1:0.0001:10;
---> I = [1;0;5;0;5;0]; t = 1:0.001:10;
---> I = [0;0;5;0;10;0]; t = 1:0.001:10;
I = [1;1;5;0;5;0]; t = 1:0.0001:10;
%}
t = 1:0.0001:102;
[t,x] = ode45(@nlsys_solve,t,I);
plot(t,x,'linewidth',1);
title('response of the non-liear system');

%%Function Definition
function nlsys= nlsys_solve(~,x)
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;

A_f =[0 1 0 0 0 0;0 0 (-m1*g)/M 0 (-m2*g)/M 0;0 0 0 1 0 0;0 0 -(g/
l1)*((m1+M)/M) 0 (-m2*g)/M*l1 0;0 0 0 0 0 1;0 0 (-m1*g)/M*l2 0 -(g/
l2)*((m2+M)/M) 0];
B_f =[0;1/M;0;1/(M*l1);0;1/(M*l2)];
eigs(A_f);
C = [1,0,0,0,0,0];
%C = diag([1 1 1 1 1 1]);
%C1 = [1,0,0,0,0,0];
%D = [0;0;0;0;0;0];
D = [0];


Poles_L_Obs = [-28;-30;-10;-6;-2;-4];

L = place(A_f',C',Poles_L_Obs);

F = L*x;

nlsys = zeros(6,1);

nlsys(1)= x(2);
```

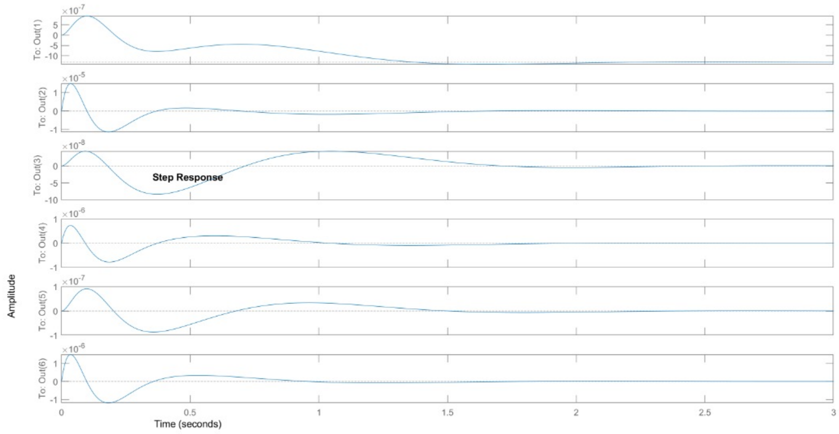
```
nlsys(2)= ((F) - m1*g*sind(x(3))*cosd(x(3)) - m1*l1*x(4)^2*sind(x(3))
 - m2*l2*x(6)^2*sind(x(5))  - m2*g*sind(x(5))*cosd(x(5)))/(M
+m1*(sind(x(3)))^2+ m2*(sind(x(5)))^2);

nlsys(3)= x(4);

nlsys(4)= (((F) - m1*g*sind(x(3))*cosd(x(3)) - m1*l1*x(4)^2*sind(x(3))
 - m2*l2*x(6)^2*sind(x(5))  - m2*g*sind(x(5))*cosd(x(5)))*cosd(x(3))/
(M+m1*(sind(x(3)))^2+ m2*(sind(x(5)))^2)*l1)-((g*sind(x(3)))/l1);

nlsys(5)= x(6);

nlsys(6)= (((F) - m1*g*sind(x(3))*cosd(x(3)) - m1*l1*x(4)^2*sind(x(3))
 - m2*l2*x(6)^2*sind(x(5))  - m2*g*sind(x(5))*cosd(x(5)))*cosd(x(5))/
(M+m1*(sind(x(3)))^2+ m2*(sind(x(5)))^2)*l2)-((g*sind(x(5)))/l1);

end
```

Published with MATLAB® R2018a

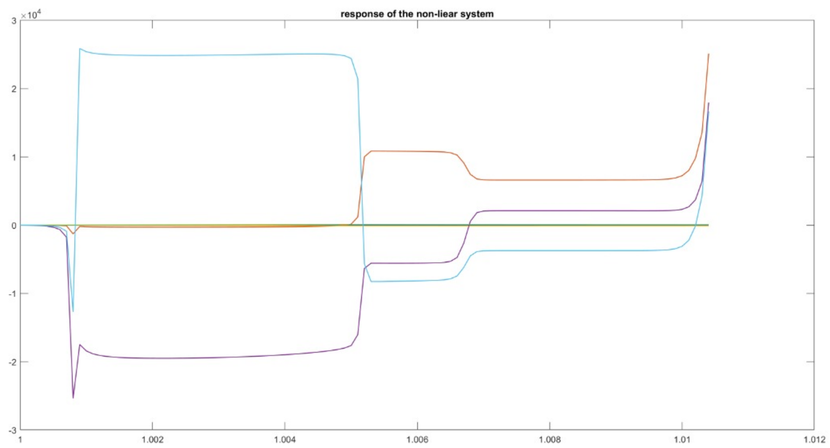
Chapter 6

Results

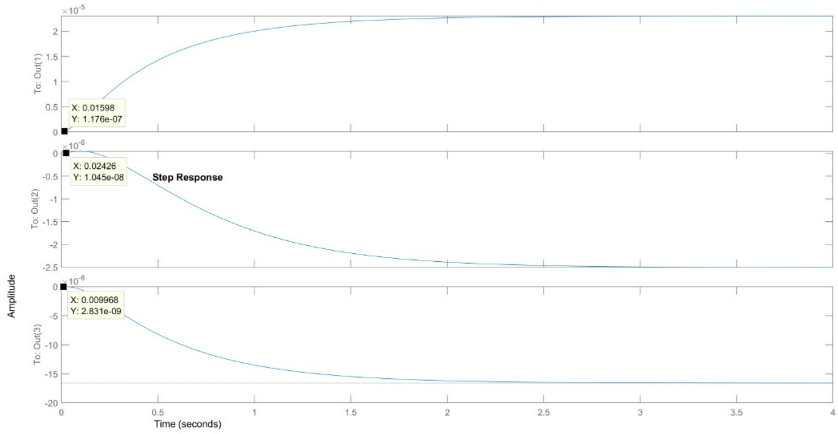
The state output for LQR Controller are obtained as

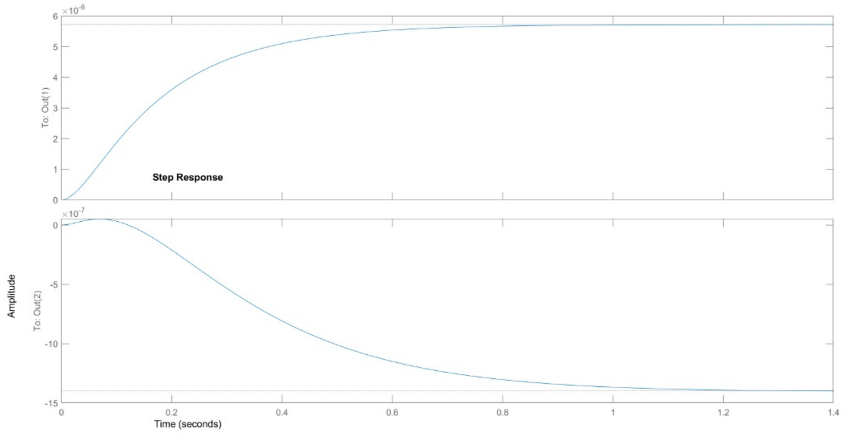


The state output for LQR Controller for nonlinear system is obtained as

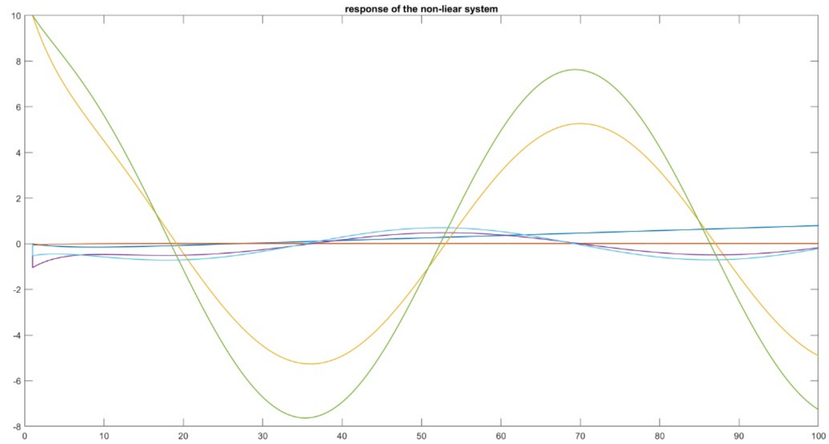


The outputs for Luenberger observer for linear system is obtained as

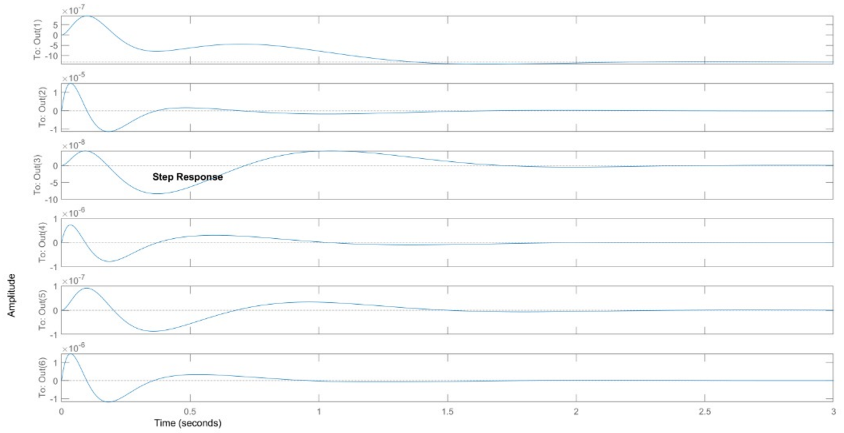




The outputs for Luenberger observer for nonlinear system is obtained as



The state estimates obtained using the LQG Controller are:



6.1 Conclusion

As per the given problem statement, The system is linearized about an equilibrium point. Design of LQR, Luenberger Observer and LQG is performed for the given system.