DESIGN OF FULL STATE FEEDBACK CONTROLLER FOR INVERTED PENDULUM CART SYSTEM USING STATE SPACE MODELLING METHOD

ECE 2010- Control System

Project Report

Prof: Bagubali A

Slot: F1+TF1

By:

G Vignesh 18BEC0692

Nikhil Subramanian 18BEC0711

Swapnil Mishra 18BEC0983

Acknowledgment

I would like to take this opportunity to thank my Professor, Bagubali Sir for his continued support and help in completing this project. Without his support and suggestions, it would not have been possible for me to complete this project.

This project helped me in understanding the concept of State space modelling approach for designing feedback controllers for physical systems and implementing them in MATLAB.

I would also like to thank Vellore Institute of Technology for providing me support in the form of arranging Guest lectures and Value Added Programs that help us in improving our technical knowledge and skills.

This project consumed huge amount of time and effort and would not have been possible without the help of many individuals.

ABSTRACT

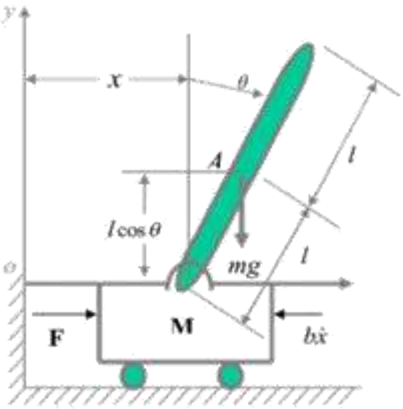
In this report an inverted pendulum is designed using state space modelling approach and a full state feedback controller is designed using pole placement and LQR (Linear Quadratic Regulation) methods. Also, tracking problem is addressed by designing a steady state error controller. And then, considering the real scenario, a reduced state observer is designed by assuming some of the state variables are measurable, and then for a worst case scenario a full order state observer is designed. Finally, the state feedback controller and the state observer are summed up to provide a pre-compensator and an overall steady state error controller is also added to that new system. All mathematical models are presented clearly and simulations together with their analysis were done using MATLAB software. For better clarity on what is happening with the control method and the system, an animation GUI is also designed.

INTRODUCTION

Inverted pendulum cart system is designed to obtain optimal control on its stability with the help of Pole Placement Controller and LQR methods. The designed controller provides better set point tracking and disturbance rejection. As a typical control system, the control of an inverted pendulum is excellent in testing and evaluating different control methods. It is highly unstable and non-linear system and it is unstable without control ,i.e. the pendulum will simply fall over if the cart isn't moved to balance it. The main objective of the control system is to balance the inverted pendulum by applying a force to the cart that the pendulum is attached to. The Inverted Pendulum Cart System is a classical benchmark control problem. Its dynamics resembles with that of many real world industrial systems of interest like missile launchers, pendulum bots, human walking, stability control of walking robots, vibration control of launching platform for shuttles etc.

Obtaining Equations:

<u>Given parameters-</u>



M – Mass of the cart = 0.5 kg

m – Mass of the pendulum = 0.5 kg

b – Friction of the cart = 0.1 N/m/sec

l – Length to pendulum center of mass = 0.3 m

I – Inertia of the pendulum = 0.006 kg*m^2

F – Force applied to the cart

X – Cart position co-ordinate

θ - Pendulum angle from vertical

- **Linearizing the non-linear equation of motion of the system around θ=0 and**
- **Finding the associated state space model of the system**

Taking the system of equations of motion of the system from above-

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

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

<u>System modeling</u>

Since the analysis (state space model) and control design techniques that I will employ in this problem apply only to linear systems, these equations need to be linearized. Specifically, I will linearize the equations about the vertically upward equilibrium position, θ = 0, and will assume that the system stays within a small neighborhood of this equilibrium

Linearization:

Here, instead of using the linearization method (Taylor method for small perturbations), I will use simple approximation around θ = 0. This assumption is **reasonably valid** since under control it is desired that the pendulum not deviate more than 20 degrees (0.35 radians) from the vertically upward position.

So, for small θ,

$sin\,\theta = \theta$, And

$cos\,\theta = 1$, (According to the above diagram θ will lie in the first quadrant. so, the Cosine function is positive)

Also dropping all the non-linear components ($\dot{\theta}^2 = 0$) the above equations will become:

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta} = F \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$(I + ml^2)\ddot{\theta} - mgl\theta = -ml\ddot{x} \quad \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Solving for $\ddot{x}$ and $\ddot{\theta}$ independently from equation (3) and equation (4) respectively gives,

$$\ddot{x} = \frac{-ml}{M+m}\ddot{\theta} - \frac{b}{M+m}\dot{x} + \frac{1}{M+m}F \quad \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

$$\ddot{\theta} = \frac{-ml}{I+ml^2}\ddot{x} + \frac{mgl}{I+ml^2}\theta \quad \ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Substituting equation (5) in (4) and equation (6) in (3) it yields,

$$\ddot{\theta} = \frac{mgl(M+m)}{I(M+m)+Mml^2}\theta + \frac{mlb}{I(M+m)Mml^2}\dot{x} - \frac{ml}{I(M+m)+Mml^2}F$$

$$\ddot{x} = \frac{-gm^2l^2}{I(M+m)+Mml^2}\theta - \frac{b(I+ml^2)}{I(M+m)+Mml^2}\dot{x} + \frac{I+ml^2}{I(M+m)+Mml^2}F$$

Designing State Space Model of the System

Now, let $u = F,\ x_1 = \theta, x_2 = \dot{\theta} = \dot{x}_1,\ x_3 = x,\ x_4 = \dot{x} = \dot{x}_3,\ y_1 = x, y_2 = \theta$, the four state equations will be,

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{mgl(M+m)}{I(M+m)+Mml^2}x_1 + \frac{mlb}{I(M+m)Mml^2}x_4 - \frac{ml}{I(M+m)+Mml^2}u$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{-gm^2l^2}{I(M+m)+Mml^2}x_1 - \frac{b(I+ml^2)}{I(M+m)+Mml^2}x_4 + \frac{I+ml^2}{I(M+m)+Mml^2}u$$

State space model of an LTI system is

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu}$$

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du}$$

So, substituting the above equations in to the matrices properly will give the following state space representation .

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 & 0 & \frac{mlb}{I(M+m)Mml^2} \\ 0 & 0 & 0 & 1 \\ \frac{-gm^2l^2}{I(M+m)+Mml^2} & 0 & 0 & -\frac{b(I+ml^2)}{I(M+m)+Mml^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{ml}{I(M+m)+Mml^2} \\ 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \end{bmatrix} u \qquad \dots\dots\dots\dots (7)$$

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \qquad \dots\dots\dots\dots (8)$$

In MATLAB,

>> system = ss(A,B,C,D) % to find the state space model

>> G = tf(system) % to find the transfer function of the system (for both outputs)

Examining the results , especially the numerators , in the transfer functions of each output solved from the state space model , there exists some terms with very small coefficients . These terms should actually be zero , they only show up due to numerical round – off errors that accumulate in the conversion algorithms that MATLAB employs . This can be checked by solving the transfer functions of the cart and the pendulum independently from the motion equations using Laplace transform by assuming zero initial conditions.

- **Simulating the dynamic behavior of the system under impulse force and step force**
- **Analyzing the stability, controllability and observability condition of the system**

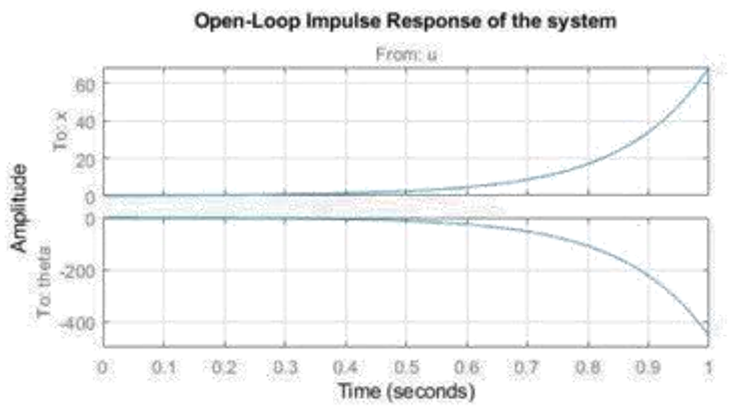
The design requirements for the Inverted Pendulum project are:

- Settling time for x and θ of less than 5 seconds.
- Rise time for x of less than 0.5 seconds.
- Overshoot of θ less than 20 degrees (0.35 radians).

<u>System Analysis</u>

1. Open-loop impulse response of the system

Examining on how the system responds to a 1-N sec impulsive force applied to the cart , I found the following result .

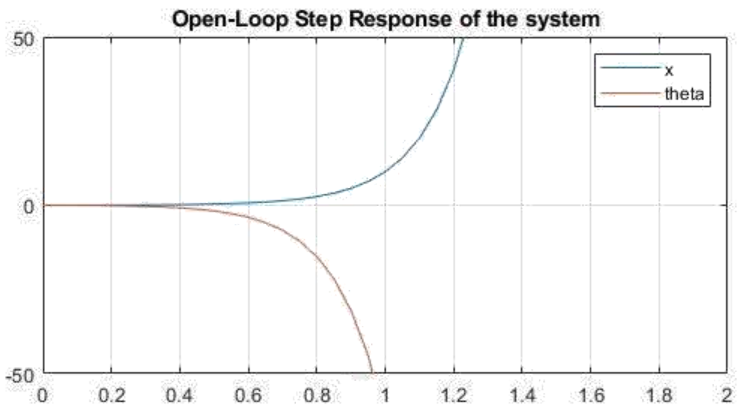


From the plot, the response is unsatisfactory. Both outputs never settle , the angle of pendulum goes to several hundred radians in a clockwise direction though it should be less than 0.35 rad. And the cart goes to the right infinitely**. So, this system is unstable in an open loop condition when there is a small impulsive applied to the cart.**

2. Open-loop step response of the system

Here also, it can be seen from the outputs that the system is unstable under 1-Newton step input applied to the cart. The outputs are found by using **lism** command of MATLAB which can be employed to simulate the response of LTI models to arbitrary inputs.

The below plot shows that the responses of the system to a step-force are unstable.



3. Stability of the system

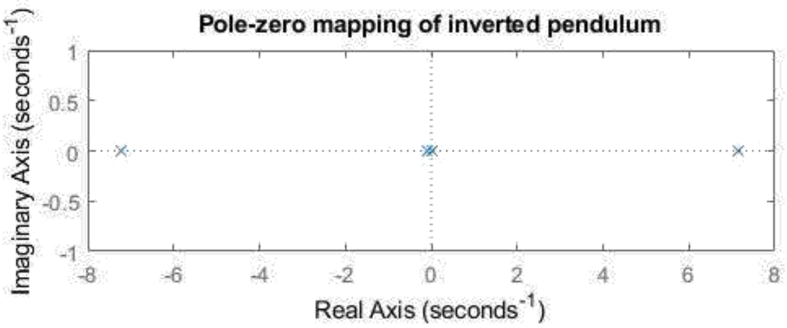
To check stability means to analyze whether the open-loop system (without any control) is stable. That has partly done by the above simulations under the impulse and step forces. But as per the definition, the **eigenvalues** of the system state matrix, A, can determine the stability. That is equivalent to finding the poles of the transfer function of the system. The eigenvalues of the A matrix are the values of s where .A system is stable if all its poles have lied in the left-half of the s-plane.

```
>>  poles = eig(A)

>>  poles =

          0
     7.1430
    -7.2220
    -0.1000
```

Or it can be shown by pole-zero mapping of the system as below using MATLAB command `>> pzmap(system)`



As can be seen from the output, there is one pole on the right-half plane at **7.1430**. This confirms the intuition that the system is **unstable** in open loop.

4.Controllability of the system

A system is controllable if there exists a control input u(t) that transfers any state of the system to zero in finite time. It can be shown that an LTI system is controllable if and only if its controllability matrix, CO, has full rank. i.e. if rank(CO)=n where **n** is the number of states. So, the necessary and sufficient condition for controllability of the system is:

$\text{rank}(CO) = \text{rank}[B\ AB\ A^2B\ ...\ A^{n-1}B] = n$

Using the following MATLAB command, the controllability matrix of the system is:-

```
>> RankCO=rank(ctrb(A,B))
>> RankCO = 4
```

So, since the controllability matrix has full rank, the system is controllable.

5. **Observability of the system**

In cases where all the state variables of a system may not be directly measurable, it is necessary to estimate the values of the unknown internal state variables using only the available system outputs.

Conceptually, a system is observable if the initial state, **x(t_0)**, can be determined from the system output, **y(t)**, over some finite time $t_0 < t < t_f$.

So, necessary and sufficient condition for Observability is:

$$rank\ (OB) = rank \begin{bmatrix} C \\ CA \\ CA^2 \\ ... \\ CA^{n-1} \end{bmatrix} = n$$

So, in MATLAB using the command

```
>> Rank(OB)=rank(obsv(A,C))
>> RankOB =

      4
```

- Designing a state feedback controller to stabilize the system by improving performance of the system using **pole placement** and **linear quadratic regulator(LQR)** methods
- If all the states are not measurable, designing a faster stable full-order state observer for state feedback controller.
- If only , are measurable, designing a reduced-order state observer for state feedback controller.
- To make design more challenging, applying a step input to the cart and yet achieving the following design requirements,
  - Settling time for and of less than 5 seconds.
  - Rise time for of less than 0.5 seconds.
  - Overshoot of less than 20 degrees (0.35 radians).

Designing full-state feedback controller

Here the main purpose is to design a controller so that when a step reference is given to the system, the pendulum should be displaced, but eventually return to zero (i.e. vertical) and the cart should move to its new commanded position.

Design procedure:

- Checking if the pair ( ) is controllable.
- Constructing equations that will govern the controller dynamics
- Placing the eigenvalues of the controller matrix in a desired position by finding an arbitrary

vector state feedback control gain vector assuming that all of the state variables are measurable. This can be accomplished using either of the two methods – **pole placement method** and **LQR (Linear Quadratic Regulation) method**.

So, it has been checked before that the system is controllable. i.e. the pair ( ) is controllable. Since, the state of the system is to be to be feedback as an input, the controller dynamics will be:

$$u = r - Kx$$

$$\dot{x} = Ax + B(r - Kx) = (A - BK)x + Br$$

$$y = Cx$$

1- **Using pole placement method**

This method depends on the performance criteria, such as rise time, settling time, and overshoot used in the design. The design requirements are,

- ✓ Settling time, $\mathbf{T_{settling}}$, for both outputs should be less than 5 seconds. And
- ✓ %Overshoot, **%OS**, of the angle of the pendulum should be less than 20 (0.35 radians).

Design procedures for pole placement :-

1. Using time domain specifications to locate dominant poles – roots of $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$. This is done by using the following formulas and finding the dominant poles at $-\sigma \pm j\omega_d$.

   - $\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}}$
   - $T_{settling} = \frac{4}{\zeta\omega_n}$, valid up to $\zeta \sim 0.7$, and
   - $\omega_d = \omega_n\sqrt{1-\zeta^2}$
   - $\sigma = \zeta\omega_n$
   - $\beta = \cos^{-1}\zeta$

2. Then placing rest of poles so they are “much faster” than the dominant second order behavior.

Typically, keeping the same damped frequency ω_d and then moving the real part to make them faster than the real part of the dominant poles so that the transient response of the real poles of the system will decay exponentially to insignificance at the settling time generated by the second order pair. While taking care of moving the poles too far to the left because it takes a lot of control effect (needs large actuating signal).

#Procedure – I

Given $\%OS = 20$ (0.35 rad) and $T_{settling} = 5sec$, calculating the parameters as follow:

ζ=0.456,

ω_n=1.78rad/sec,

$\beta = cos^{-1}\zeta = 62.87^0$,

$\sigma = \zeta\omega_n$= -0.811,

$j\omega_d = 1.584$

So, the **dominant poles** are:

=> $- \zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$ = -0.811+j1.584 and -0.811-j1.584 (which are complex conjugates)

Procedure – II

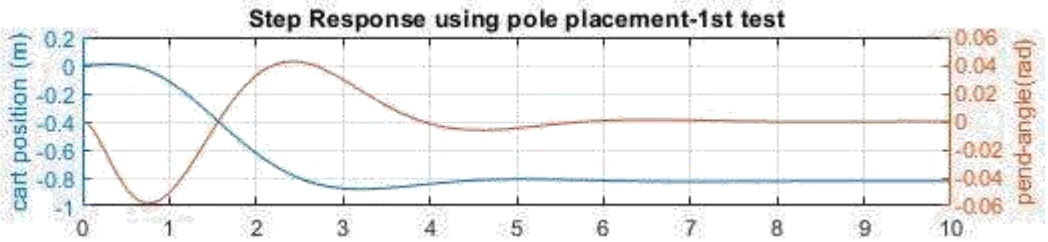
Using MATLAB software, the following are the gain vectors for different sets of desired poles.

 # Test -1

`K1 = [-12.4835   -1.2055   -0.2423   -0.4731]`

`Desired poles_1 = [-2.433 -1.622 -0.811 ± 1.584j],` by making the remaining poles **2 and 3** times

faster than the real part of the dominant poles.

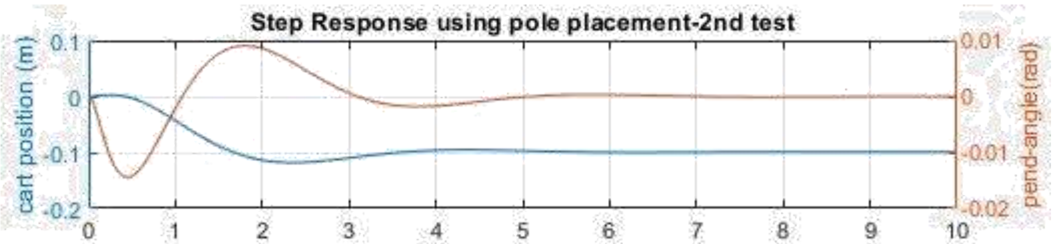


 # Test -2

`K2 = [-21.0855 -3.2251 -2.0191 -1.8811]`

`Desired poles_2 = [-8.11 -4.055 -0.811 ± 1.584j],` by making the remaining poles **5 and 10** times

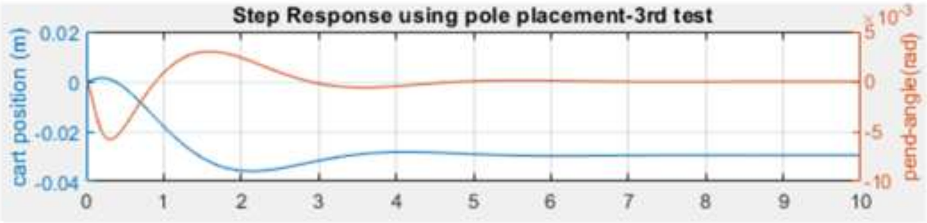
faster than the real part of the dominant poles.



Test -3

```
K3 = [-40.2010 -5.9361 -6.7842  -4.8694]
Desired poles_3 = [-11.354 -9.732 -0.811 ±  1.584j],
```
by making the remaining poles **10 and 12**

times faster than the real part of the dominant poles **.**



As can be seen from the respective plots of the system step response for each calculated gain vectors as per the desired pole locations, System design requirements are satisfied in all the three tests. And, the system response tends to be faster when the real poles go farther to the left from the real part of the dominant poles. A more faster response can be found by moving the real poles deeper in to the left half side of the s-plane, but it requires a larger actuating signal which in turn brings larger control effort.

2- Using Linear Quadratic Regulator (LQR)

This approach is to place the pole locations so that the closed-loop system optimizes the cost function given by

$$J_{QR} = \int_0^\infty [x(t)^T Q x(t) + u(t)^T R u(t)]dt$$

Where:

$x^T Q x$ - is the **state cost** with weight $\boldsymbol{Q}$

$u^T R u$ - is the **control cost** with weight $\boldsymbol{R}$

Therefore, LQR selects closed-loop poles that balance between state errors and control effort.

Design procedures for LQR :-

1. Selecting design parameter matrices **Q** and **R**
    - Easily, starting with $Q = C'C$ *and* $R = 1$ and increasing the weight on the matrix Q
    - This is done supposing that the cost $J = \int(y^2 + u^2)dt$ is to be minimized. If so, $\boldsymbol{y = Cx,\ y^2 = x'C'Cx = x'Qx}$ *while* $\boldsymbol{Q = C'C}$
2. Solving the algebraic *Riccati* equation for **P**, (done using MATLAB software)
    - $\boldsymbol{A^T P + PA + Q - PBR^{-1}B^T P = 0}$
3. Finding the state variable feedback using
    - $\boldsymbol{K = R^{-1}B^T P}$
4. Simulating results using *lsim* MATLAB command and observing
5. Verifying design, if it seen to be unsatisfactory, trying again with different weighting matrices **Q** and **R**

All the above procedures are accomplished easily in MATLAB as follow:

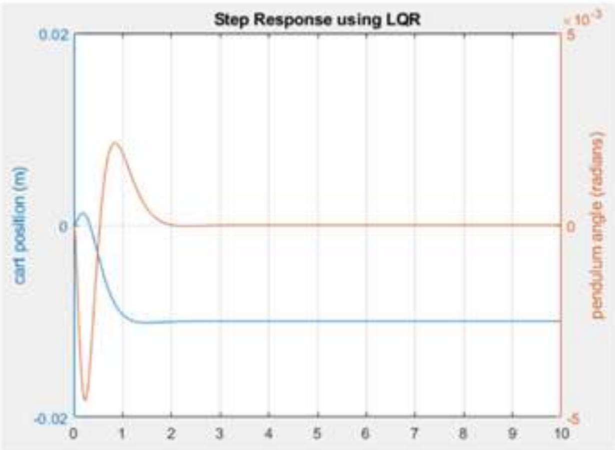
```
>> Q = C'*C;
>> Q(1,1) = 10;
>> Q(3,3) = 100
>> R = 1
>> K = lqr(A,B,Q,R)
```

```
>> Ac = A-B*K;        %control matrix
>> system_c = ss(Ac,B,C,D);
```

And all the results can be seen below.

$$Q = \begin{bmatrix} 80 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 400 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ R = 1$$

K = [-38.1423   -6.2197 -10.0000   -7.7060]



- The reason this weighting was chosen is because it just satisfies the transient design requirements.

  Increasing the magnitude of more would make the tracking error smaller, but would require greater control force. More control effort generally corresponds to greater cost (more energy, larger actuator,

  etc.). More improved responses can be seen clearly by varying the weights **GUI** that will be attached with this document.

- The above LQR design method has brought a good stability to the system and fulfilled the design requirements in a good manner. But, if the reference input is different from zero,

  $r(t) = a$ not equal to 0 , the systems performance will be degraded. So, for best quality of performance, an **asymptotic tracking of a step reference input** must be designed.

Designing an asymptotic error tracking controller for good regulation

- So far, both controller design methods, the pole placement and LQR design, helped to pick the gain vector K so that the dynamics of the system to have nice properties – more importantly to stabilize A.

This deals with performance issue of the system rather than just stability.

$\dot{x} = Ax + Bu$

$y = Cx$

$u = r - Kx$

For **good tracking performance**, it should be

$y(t) \approx r(t)\ as\ t \to \infty$

To make this, one solution is to scale the reference input $r(t)$ so that,

$u = \bar{N}r - Kx$, where $\bar{N}$ – is a **feedforward gain** used to scale the closed loop transfer function.

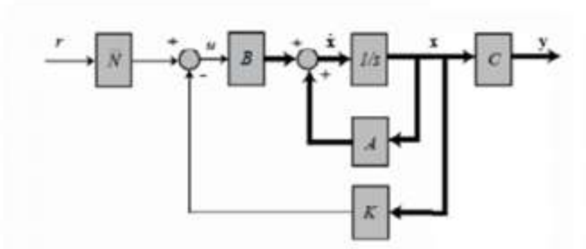


Fig. Adding a pre-compensation scale factor to the reference input command

$\dot{x} = (A - BK)x + B\bar{N}r$

$y = Cx$

Then, the new transfer function will be

$$\frac{Y(s)}{R(s)} = C(sI - (A - BK))^{-1}B\bar{N} = G_{cl}(s)\bar{N}$$

So, clearly the scaling factor $\bar{N}$ can be computed as follow.

$$\bar{N} = G_{cl}(0)^{-1} = -(C(A - BK))^{-1}B)^{-1}$$

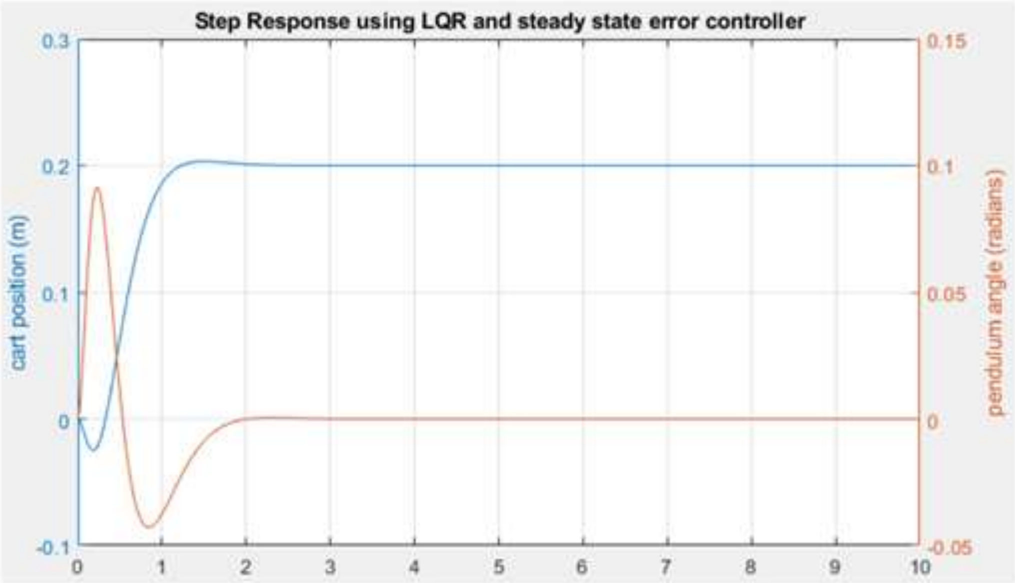
In this specific problem, before calculating $\bar{N}$, the $C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ should be modified to a new one $C_n = [0 \quad 0 \quad 1 \quad 0]$ because it is redefined that $y = x$, which means the reference input is only applied to the position of the cart.

Therefore, having K from the previous LQR design and scaling the reference input with following result, the system responses are plotted below.

$$\bar{N} = -(C_n(A - BK)^{-1}B)^{-1}$$

In MATLAB ,

```
>> Cn = [0 0 1 0]; %modification of C matrix
>> Nbar=-inv(Cn*((A-B*K)\B));
>> system_cl = ss(Ac,B*Nbar,C,D);
```



As can be seen from the above plot, all design requirements are satisfied. i.e., the overshoot is below the limits, the settling time is also < 5sec and the rise time is < 0.5 sec . And also, it can be clearly seen that the system performance becomes better and smooth than the one found using the previous controller designed by LQR only.

Designing a reduced order state observer

If θ and x are measurable, then a reduced order state estimator can be designed to estimate $\dot{\theta}$ and $\dot{x}$.

To do this taking the original state space representation

$\dot{x} = Ax + Bu$

$y = Cx$

Generally if p of he n states are measurable,

$x = \begin{bmatrix} x_m \\ x_u \end{bmatrix}$, where $x_m \in R^p$ *and* $x_u \in R^{n-p}$ are the measured and unmeasured states respectively.

And the system dynamics can be described as:

$\dot{x}_m = A_{mm}x_m + A_{mu}x_u + B_m u$

$\dot{x}_u = A_{um}x_m + A_{uu}x_u + B_u u$

$y = C_m x_m$

For $\boldsymbol{z(t)}$ to be an estimate of $\boldsymbol{Tx(t)}$, that is, for $\boldsymbol{\dot{z} = Fz + gy + hu}$ to be an estimate of $\boldsymbol{T\dot{x}}$, the following conditions must be satisfied.

- $\boldsymbol{TA - FT = gc}$
- $\boldsymbol{h = TB}$
- **All eigenvalues of F have to have negative real parts and are different from the eigenvalues of A.**

So, to estimate the states $\dot{\theta} = \dot{x}_1$ and $\dot{x} = \dot{x}_3$, the reduced dimension observer can be designed as the following:

Design Procedures :-

1. Choosing an arbitrary $2x2$ matrix F so that all eigenvalues have negative real parts and are different from those of A.

```
F =

  1.0e+003 *

   -0.0632    0.0011
   -1.0497    0.0016
```

And checking its `eig(F)= -30.8206 +10.2537i` and `-30.8206 -10.2537i`, both eigenvalues of F have negative real values and all are different from those of `eig(A)= 0,7.1430, -7.2220, -0.1000.`

2. Choosing a 2x2 vector g so that (F, g) is controllable,

$g =$
$1.0e + 004 *$

$-6.4715 \quad 0.0600$

$-0.2790 \quad 0.0034$

And checking $rank\big(ctrb(F, g)\big) = 2$ so, it is satisfied.

3. Solving 2x4 matrix T using Lyapunov equation $TA - FT = gC$ in MATLAB software

$T = lyap(-F, A, -g * C)$

```
T =

  1.0e+004 *

   -0.0035    0.0002    0.0094   -0.0068
   -0.1986    0.0084    6.4223   -0.3785
```

4. Checking if the square matrix P of order 4 is singular or non-singular

$P = \begin{bmatrix} C \\ T \end{bmatrix} =$

p =

```
 1.0e+004 *


      0         0    0.0001         0
 0.0001         0         0         0
-0.0035    0.0002    0.0094   -0.0068
-0.1986    0.0084    6.4223   -0.3785
```

Finding its inverse $inv(P) =$

$P^{-1} =$

```
  1.0e+003 *

   -0.0000    0.0010         0         0
    2.0405   -0.0009    0.0019   -0.0000
    0.0010         0         0         0
    0.0624   -0.0005    0.0000   -0.0000
```

5. If P is non-singular, computing $h = TB$

$h = TB=$

```
1.0e+003 *

  -0.1315
  -7.2163
```

Then, the reduced observer can be put as follow.

$\dot{z} = Fz + gy + hu$
$y = C\hat{x}$
All the coefficient matrices are calculated above.

Finally, the estimate is

$$\hat{x} = P^{-1}\begin{bmatrix} y \\ z \end{bmatrix}$$

And the error is

$$\dot{e} = \dot{z} - T\dot{x} = Fz + gy + hu - T(Ax + bu)$$
$$\dot{e} = Fe$$

$$\hat{x} = 1.oe + 003 * \begin{bmatrix} -0.0000 & 0.0010 & 0 & 0 \\ 2.0405 & -0.0009 & 0.0019 & -0.0000 \\ 0.0010 & 0 & 0 & 0 \\ 0.0624 & -0.0005 & 0.0000 & -0.0000 \end{bmatrix}\begin{bmatrix} y \\ z \end{bmatrix}$$

$$\dot{e} = 1.0e + 003 * \begin{bmatrix} -0.0632 & 0.0011 \\ -1.0497 & 0.0016 \end{bmatrix} e$$

Designing full order state observer (closed loop state estimator)

So far, it was assumed that there is full or partial access to the state ( ) when the controllers in each method are designed. But in reality, most often, all of this information is not available.

To address this issue, a replica of the dynamic system that provides an “estimate” of the system states based on the measured output of the system should be developed – the so called a state observer or state estimator.

Here, a **full order state observer** will be developed assuming as if all the states can’t be measured taking in to consideration that there might be lack of sensors.

Design procedures:

1- Checking whether the pair (A,C) is observable

2- Developing estimate of x(t) that will be called x^(t)

3- Selecting a suitable real constant vector so that all eigenvalues of (A-LC) have negative real parts

4- Making sure that the estimator eigenvalues are faster than the desired eigenvalues of the state feedback, because it will be used together to form a **compensator.**

So, it has been checked initially that this system is observable. i.e. the pair (A,C) is observable.

$$\dot{\hat{x}} = A\hat{x} + Bu + L\tilde{y}$$

$$= A\hat{x} + Bu + L(y - \hat{y})$$

$$= (A - LC)\hat{x} + Bu + Ly$$

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

$$u(t) = \bar{N}r - K\hat{x}$$

Then the closed loop estimator error dynamics is

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}}$$

$$= [Ax + Bu] - [A\hat{x} + Bu + L(y - \hat{y})]$$

$$= (A - LC)(x - \hat{x})$$

$$\dot{\tilde{x}} = (A - LC)\tilde{x}$$

$$\dot{e} = (A - LC)e \text{ Where } e = \tilde{x} = x - \hat{x}$$

This equation governs the estimator error. If all eigenvalues of (A - LC) can be can be assigned in the negative half plane, then all entries of the error will approach to zero at faster rates. Thus, there is no need to compute the initial state of the original state equation.

To select a common guideline is to make the estimator poles **4-10** times faster than the slowest controller pole. Making the estimator poles too fast can be problematic if the measurement is corrupted by noise or there are errors in the sensor measurement in general.

Controller poles from the above system with error tracking controller are:

```
>>contr_poles=eig(Ac)

>>contr_poles =

  -7.6280 + 3.4727i

  -7.6280 - 3.4727i

  -3.1744 + 2.1467i

  -3.1744 - 2.1467i
```

The slowest poles have real parts at **-3.1744**, so the estimator poles can be placed at -30. Since the closed-loop estimator dynamics are determined by a matrix (A-LC) that has a similar form to the matrix that determines the dynamics of the state-feedback system (A-BK).

Therefore, the same commands, that were used to find the state feedback, can be used to find the gain estimator gain .

Since eig(A-LC)=eig(A|- C|L|) and A|- C|L| have an exact forma as A-BK then the MATLAB command ‘place; can be used to compute L.

```
 %Placing obsserver poles

>>  obsr_poles = [-30 -31 -32 -33];
>>  L = place(A',C',obsr_poles)'
L =.

  1.0e+003 *

  -0.0006    0.0632
  -0.0018    1.0497
   0.0626   -0.0008
   0.9732   -0.0341
```

Final goal is to build the compensator (state feedback controller + full state observer).

Dynamic output feedback compensator is the combination of the regulator (state feedback controller developed using the LQR method) and full state estimator using

$$u = -K\hat{x}$$

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly$$

$$\dot{\hat{x}} = (A - BK - LC)\hat{x} + Ly$$

Rewriting with new state $x_c \equiv \hat{x}$

$$\dot{x}_c = A_c x_c + B_c y$$

$$u = -K x_c$$

Where

$$A_c = A - BK - LC \quad and \quad B_c = L$$

But since there is a reference input, the compensator can

be implemented using the following feedback.

$$e(t) = r(t) - y(t)$$

Then the state space model of the compensator will be

$\dot{x}_c = A_c x_c + B_c e$ (this can be seen in the block diagram)

So, equations that can describe the overall closed loop dynamics will be

Having $u = r - Kx_c$ and $e = r - y$ while $y = Cx$

$$\dot{x} = Ax + Bu = Ax - BKx_c + br$$

$$\dot{x}_c = A_c x_c + B_c e = A_c x_c + B_c(r - y) = A_c x_c + B_c(r - Cx)$$

$$= -B_c Cx + A_c x_c + B_c r$$

In short, the overall closed loop final state space model is:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} A & -BK \\ -B_c C & A_c \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + \begin{bmatrix} B \\ B_c \end{bmatrix} r$$

$$y = [C \quad 0] \begin{bmatrix} x \\ x_c \end{bmatrix}$$

Where

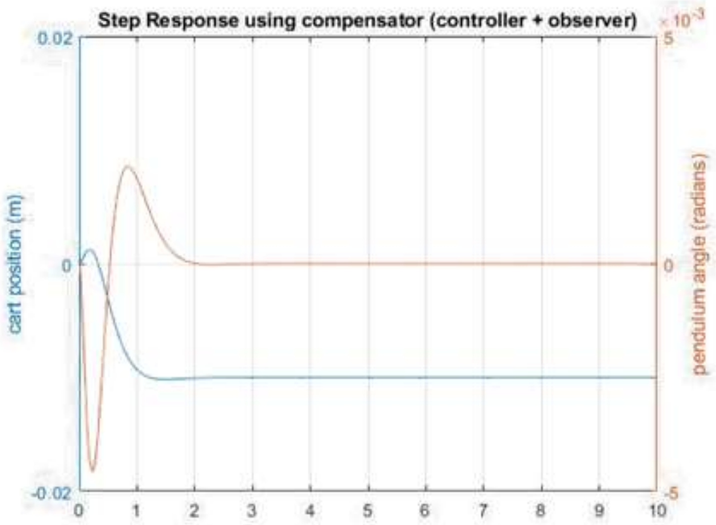
$$A_c = A - BK - LC \quad and \quad B_c = L$$

By making an ***equivalent transformation*** to the above state space model, it can be represented as the following equivalent state equation. And this equivalent state equation governs the whole **controller-estimator configuration**.

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} r \qquad \text{Where} \qquad e = \tilde{x} = x - \hat{x}$$

$$y = [C \; 0] \begin{bmatrix} x \\ e \end{bmatrix}$$

The response of the system with controller + observer is:



The overall design requirements are satisfied and the system has a relatively good performance. But it can be seen that the response graph for the inverted pendulum lucks a slight smoothness.

That means there is something which creates that vibration. i.e. there is a tracking problem of the output to the reference input. The next task will solve this problem

- **Redefining $y = x$ (step command applied to position of the cart only)**
- **Designing no steady state error tracking controller for the new SISO system to eliminate the steady state error**

Having the final *controller-observer* state space model, if the reference input is a constant different from zero, $(t) = a \neq 0$, and if it is applied only on the cart's position, it can be taken that $y = x$ so that the system will become a SISO system. So, the matrix $C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ should be modified to a new one $C_n = [0 \quad 0 \quad 1 \quad 0]$ when calculating a steady state error tracking controller that can be implemented by adding a feedforward gain $\bar{N}$ to scale the reference input.

This has the same approach as I used in the LQR controller, but slightly different in calculating the scaling factor $\bar{N}$ since it uses the **overall controller-estimator configuration closed loop matrices.**

So, taking the equivalent state equation of the new system (controller + observer),

$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} r$ Where $e = \tilde{x} = x - \hat{x}$

$y = [C \; 0] \begin{bmatrix} x \\ e \end{bmatrix}$

For **good tracking performance**, it should be

$y(t) \approx r(t) \; as \; t \to \infty$ That means DC gain of the response from r to y should be unity.

To make this, one solution is to scale the reference input $r(t)$ so that,

$u = \bar{N}r - Kx$, where $\bar{N}$ – is a **feedforward gain** used to scale the **closed loop transfer function**.

Then using the closed matrix,

$\bar{N} = -(C_{cl}(A_{cl}))^{-1}B_{cl})^{-1}$

Where $A_{cl} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}$, $B_{cl} = \begin{bmatrix} B \\ 0 \end{bmatrix}$ and $C_{cl} = [C_n \quad 0]$ in which $C_n = [0 \quad 0 \quad 1 \quad 0]$ because the step reference command is to be applied only to the cart's position $(y = x)$.

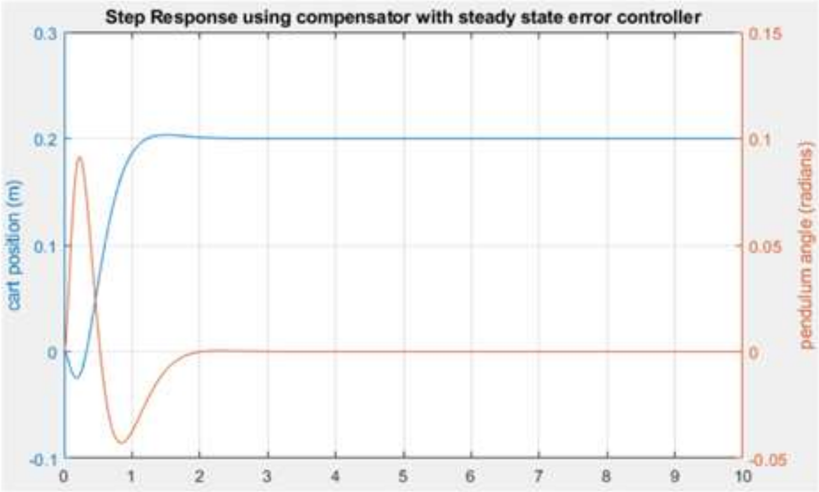
- **Then the final inverted pendulum system controlled by designing a controller, a full state observer and adding a no steady state error tracker can be described by the following state space model.**

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

$$y = [C \; 0] \begin{bmatrix} x \\ e \end{bmatrix}$$

And the system response graph can be seen below.

As can be seen clearly from the below graph that the whole design has solved clearly the problems of stability and performance. And the no steady state error tracker has eliminated the oscillation of the pendulum and made it smooth.



And it can be seen from the animation GUI graph below that, no matter the applied input force is varied, the pendulum quickly balances itself with almost zero vibration in its steady state. These graphs are draw by varying the step inputs by sliding the bar from 0.2 to 0.6 randomly and at each time pressing the "Run" command without clearing the previous output.

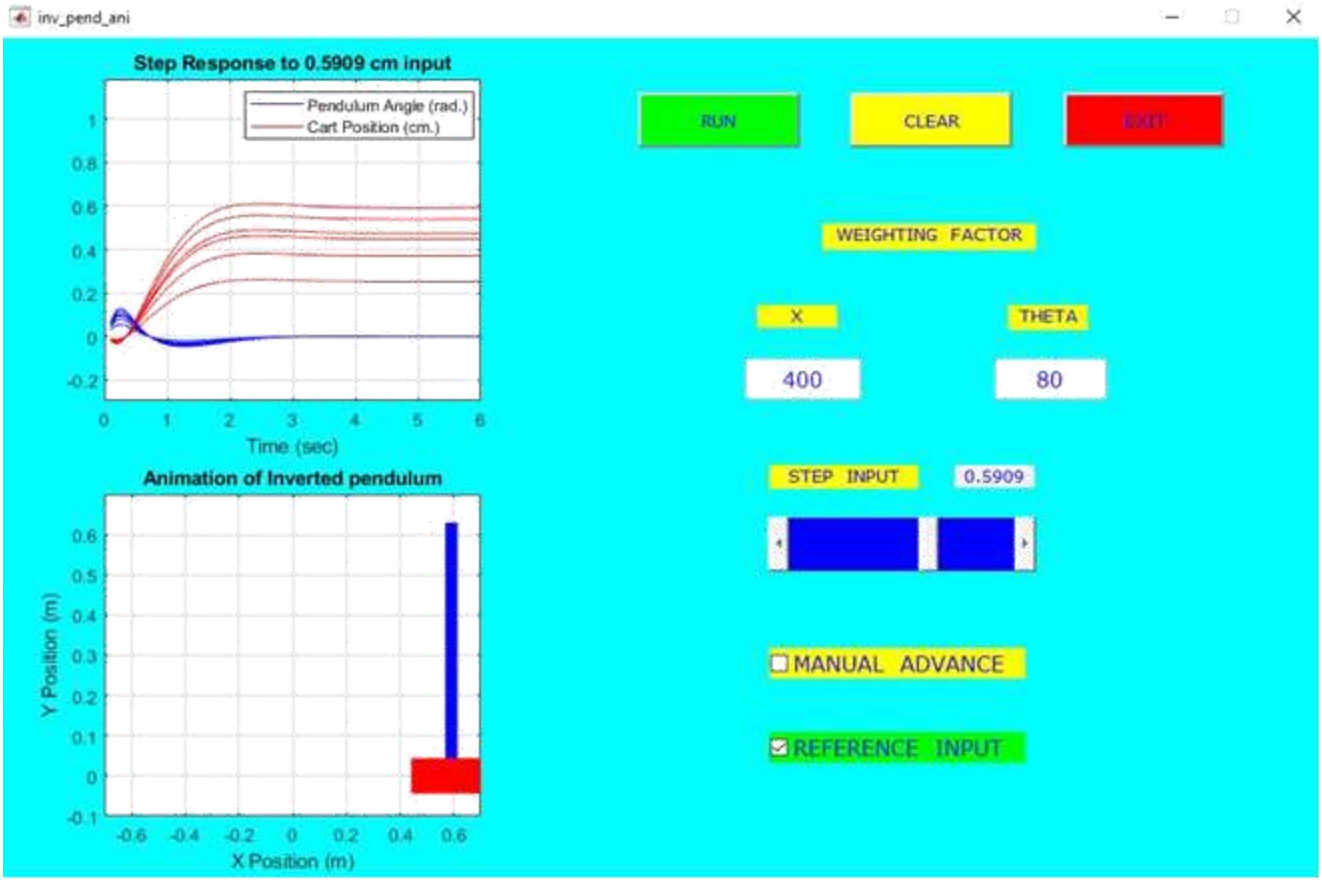


Fig . <u>Animation GUI for a controlled Inverted pendulum</u>

GUI - Explanation: -

- **Run** button performs the simulation and plots the response and the animation.
- **Clear** button clears the both plots. If the plots are not cleared, then during the next run the step response will be graphed on the same plot. This is useful if you want to graphically see the effect of varying a parameter.

- **Exit** button closes the GUI.

- **Weighting factor**
  - ***x*** - This editable text field weights the cart's position in the LQR controller. Increasing the weighting factor improves the cart's response, making it reach it's commanded position faster.

  - ***theta*** **-** This editable text field weights the pendulum's angle. Similar to the **x** weighting factor, making this larger will quicken the pendulum's response. Feel free to change the weighting factors to see what happens !

- ***Step Slider*** - The slider allows you to change the magnitude of the step disturbance on the cart.

- ***Manual Advance*** - If this control is checked, the user is able to advance the animation and plot one frame at a time. The frames are advanced by pressing any key on the keyboard. This function is useful if the animation moves too fast for the user and will allow the user to better visualize the entirety of the system's motion.

Reference Input - This box is automatically checked when the GUI is run. By un-checking it the user removes the reference input term, Nbar, from the simulation. The reference input is used to correct steady-state errors common to full-state feedback systems

CONCLUSION

Modeling of an inverted pendulum shows that the system is unstable without a controller. Results of applying state feedback controllers show that the system can be stabilized. Whereas both Pole placement and LQR controller methods are inconvenient because of selecting the constants of controller although they can provide a good outcome if done thoroughly.

When a DC reference input is applied to the cart, the system has failed somewhat to track the input and has resulted in a stable output with some oscillations – undesirable steady state performances. To remove this, a no steady state error tracking controller is designed which has brought good results as can be seen on the graphs in each steps.

And since in real scenario all the states can't be measured, a full-order state observer is designed. Finally, the state feedback controller is added with a full-order state observer to generate a pre-compensator and then a steady state error tracking mechanism is also added to the whole new system. All those have brought a really nice controlled inverted pendulum system with good performance and system characteristics.

REFERENCES

1. http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModelin

2. https://ijret.org/volumes/2012v01/i04/IJRET20120104003.pdf

3. http://eujournal.org/index.php/esj/article/viewFile/10540/10073

4. https://pdfs.semanticscholar.org/2299/ea58b2f4db3ea904db8315c4359184e4ab58.pdf

5. https://pdfs.semanticscholar.org/4d00/25bb4f826990cf3b4b97181c475db5d5b825.pdf

6. http://www.ijmer.com/papers/Vol3_Issue5/BZ3529242927.pdf

7. http://ethesis.nitrkl.ac.in/4460/1/610EE102_Thesis.pdf

8. https://pdfs.semanticscholar.org/6de6/2b128a4eaa0a89c4fa8f79d3ff527f790312.pdf?_ga=2.57667626.1450434179.1554891283-2037212729.1554891283