Control of Inverted Pendulum Advance system design and module control

Team Member :-Vivek Verma
Krishna Koparde
Alex Waton
Jason Newton
Bhushan Ahire
Under guidance of :- Prof. Sze Zheng Yong

03 December 2019

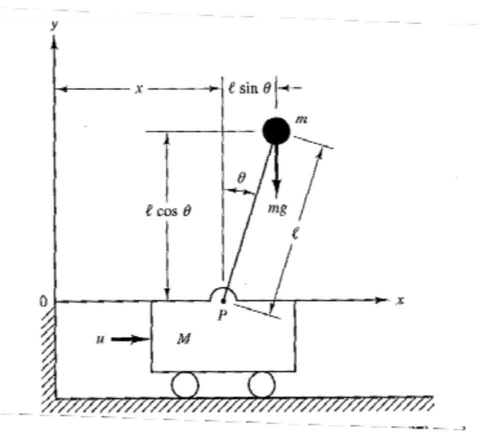
Contents

1 Introduction

The inverted pendulum system is one of the favorite experiments in control system labs. The unstable nature of the plant enables an impressive demonstration of the capabilities of feedback systems. This system is also considered to be a simplified representation of rockets flying into space. The pendulum orients downward in a stable orientation and upwards under a controlled state. Hence, in order to attain the latter state force is required. In order to frame a control law, we need to have a dynamic model of the system and we obtain the laws based on this to achieve the desired performance. This system is a non-linear one and requires linearization about a desired point (θ=0 in our case) to obtain the state space models. We'll be using Full State Feedback with closed loop poles placement, LQR and MPC controller in our project to control our system.

2 Modelling Framework

Here we consider a pendulum cart system.Figure represents the free body diagram of the system.
Here we assume that the rod of the pendulum is mass-less and the hinge to which the pendulum is fixed is frictionless. The mass of the pendulum is concentrated at the center of gravity of the pendulum which is located at pendulum ball's center. The mass of the cart is represented as M and the mass of pendulum is represented asm_p. The control force u(t) acts along the x direction of the cart. The rod's length is represented as L.The angle by which the pendulum is tilted represented as θ
Diagram for the system is



$Y_g = L * cos\theta;$ $\dot{Y}_g = -\dot{\theta} * L * sin\theta;$ $\ddot{Y}_g = -\ddot{\theta} * L * sin\theta - \dot{\theta}^2 * L * cos\theta$

$X_p = x + L * sin\theta;$ $\dot{X}_p = \dot{x} + \dot{\theta} * L * cos * \theta;$ $\ddot{X}_p = \dot{x} + \ddot{\theta} * L * cos\theta - \dot{\theta}^2 * L * sin * \theta$

$M_A = m * g * L * sin\theta = I * \ddot{\theta}$

$0 = F_x * L * cos\theta - F_y * L * sin\theta + m_p * g * L * sin\theta$

$u = (M + m) * \ddot{x} + m * \ddot{\theta} * L * cos\theta - m * \dot{\theta}^2 * L * sin\theta$

$\sum F_x = F(t) = M * \ddot{x} + m * \ddot{x_p}$

From above equation

Linearization

$0 = F_x * cos\theta + F_y * sin\theta + m * g * sin\theta$

$$-m_p * g * sin\theta = m_p(\ddot{x} + \ddot{\theta} * L * cos\theta - \dot{\theta}^2 * L * sin\theta) * cos\theta$$
$$-m_p(-\ddot{\theta} * L * sin\theta - \dot{\theta}^2 * cos\theta * L) * sin\theta$$

$$-m_p * g * sin\theta = m_p * \ddot{x} * cos\theta + m_p * \ddot{\theta} * L * cos\theta - \dot{\theta}^2 * L * sin\theta * cos\theta$$
$$+m_p * \ddot{\theta} * L * sin\theta * sin\theta - m_p * \dot{\theta}^2 * cos\theta * L * sin\theta$$

$$\ddot{x} = \frac{u - m_p * g * sin\theta * cos\theta + m_p * \dot{\theta}^2 * sin\theta}{(M + m_p) - m_p * cos^2\theta}$$

$$\ddot{\theta} = \frac{m_p * u * cos\theta + m_p * g * sin\theta + m_p^2 * \dot{\theta}^2 * L * cos\theta * sin\theta}{m_p * L * cos^2\theta - (M + m_p) * m_p * L} * cos\theta$$

State space representation

The equations above are to be represented in state space

$$\frac{dx}{dt} = f(x, u, t)$$

The state variables are

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix}; \qquad \dot{x_{eq}} = \frac{d}{dx} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \frac{d}{dx} \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} x_2 \\ \ddot{\theta}_{eq} \\ x_4 \\ \ddot{x}_{eq} \end{bmatrix}$$

where,

$\dot{\theta} = x_2$

$$\ddot{\theta} = \frac{u + m_p * l_p * sinx_1 * x_1^2 - m_p * g * cosx_1 * sinx_1}{M_c + m_p - m_p * cos^2x_1}$$

$\dot{x} = x_4$

$$\ddot{x} = \frac{u * cosx_1 - (m_c + m_p) * g * sinx_1 + m_p * l_p * (cosx_1 * sinx_1) * x^2}{m_p * l_p * cos^2x_1 - (m_c + m_p) * l_p}$$

The ouput equations are represented by

$$y = C * x \qquad or \qquad y = \begin{bmatrix} \theta \\ x \end{bmatrix}$$

$$A = \begin{bmatrix} \frac{df_1}{dx_1} & \frac{df_1}{dx_2} & \frac{df_1}{dx_3} & \frac{df_1}{dx_4} \\ \frac{df_2}{dx_1} & \frac{df_2}{dx_2} & \frac{df_2}{dx_3} & \frac{df_2}{dx_4} \\ \frac{df_3}{dx_1} & \frac{df_3}{dx_2} & \frac{df_3}{dx_3} & \frac{df_3}{dx_4} \\ \frac{df_4}{dx_1} & \frac{df_4}{dx_2} & \frac{df_4}{dx_3} & \frac{df_4}{dx_4} \end{bmatrix} \qquad B = \begin{bmatrix} \frac{df_1}{dx} \\ \frac{df_2}{dx} \\ \frac{df_3}{dx} \\ \frac{df_4}{dx} \end{bmatrix}$$

2.1 Full state feedback control with observer

Using Separate principal The Linearized model yielded state space matrix as:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 49 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -4.9 & 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ -3.3333 \\ 0 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad D = 0$$

3 Different types of controller

3.1 PID

$$MATLAB : P = ctrlb(A \qquad B) = \begin{bmatrix} 0 & -3.3333 & 0 & -163.3333 \\ -3.3333 & 0 & -163.3333 & 0 \\ 0 & 1 & 1 & 17.3333 \\ 1 & 1 & 17.3333 & 17.3333 \end{bmatrix}$$

$rank(P) = 4$ Hence system is controllable.

Computing Eigen values by fig(A) $\qquad eig(A) = \begin{bmatrix} 0 \\ 1 \\ 7 \\ -7 \end{bmatrix}$

Thus, from above few of Eigen Values lie in right plane, which means system is unstable. So, to stabilize the system we have to design 'K' which will allow us to place eigen values in the left half plane. Our desired eigen values are [-0.5,5,6,7].Therefore by placing eigen values using 'Bass Gura' or using PLACE command in MATLAB.

Characteristic Polynomial Equation.

$$a(s) = det(SI - A) = S(S-1)(S^2 - 49)$$
$$= (S^4) - (49 * S^2) + 49 * S + 0$$
$$a_0 = 0 \qquad ; a_1 = 49 \qquad ; a_2 = -49 \qquad ; a_3 = -1$$

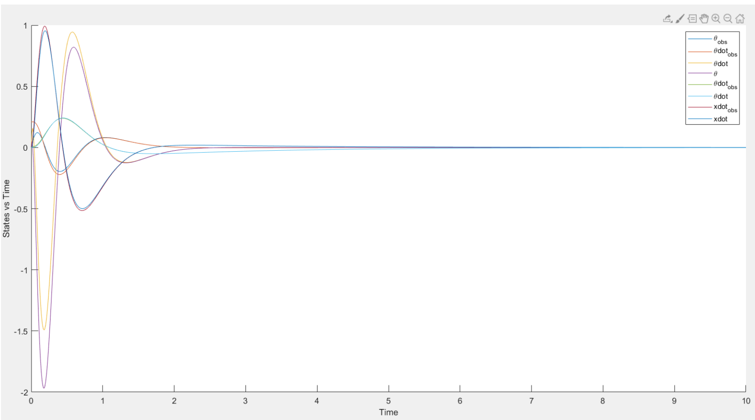
Desired Polynomial:

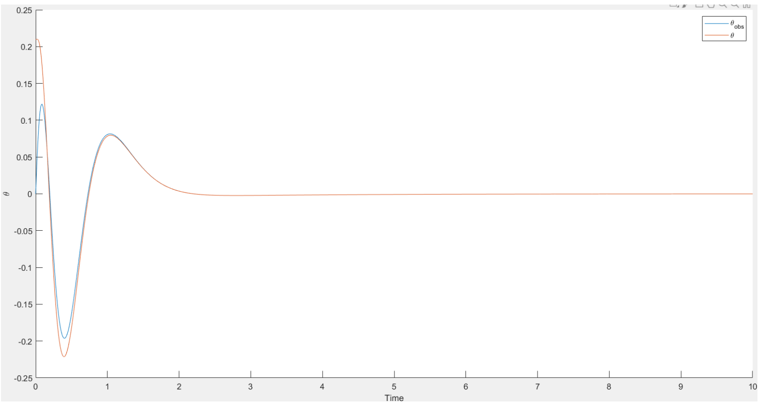
$$\alpha(S) = S^4 + 18.5 * S^3 + 131 * S^2 + 527 * S + 105$$
$$\alpha_0 = 105 \qquad ; \alpha_1 = 267.5 \qquad ; \alpha_2 = 116 \qquad ; \alpha_3 = 18.5$$

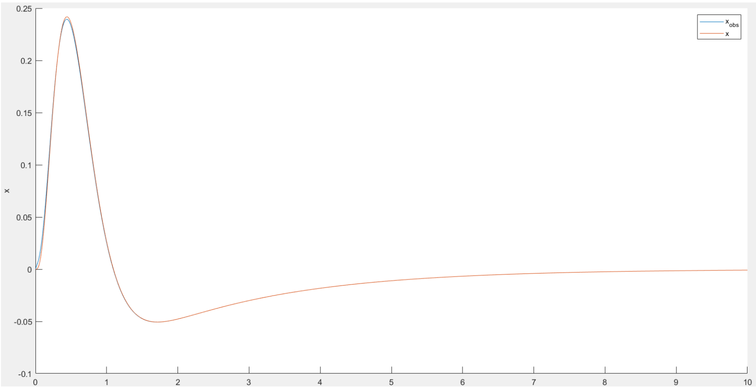
$$K = \begin{bmatrix}(\alpha_0 - a_0)(\alpha_1 - a_1)(\alpha_2 - a_2)(\alpha_3 - a_3)\end{bmatrix}$$
$$= \begin{bmatrix}105 & 214.5 & 165 & 19.5\end{bmatrix} * (ctrb(A,B) * \begin{bmatrix} 49 & -49 & -1 & 1 \\ -49 & -1 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix})^{-1}$$
$$K = \begin{bmatrix}-60.1247 & -9.6605 & -3.2143 & -12.7015\end{bmatrix}$$

We can calculate 'L' by using PLACE command in MATLAB.

$$L = \begin{bmatrix} 12.7663 & 08700 \\ 89.0034 & 5.6578 \\ 1.0857 & 14.2337 \\ 3.2522 & 57.2296 \end{bmatrix}$$







3.2 Linear Quadratic Control

The linear quadratic regulator can be used to control the system while balancing the performance characteristics and the energy consumed. In this case, the performance is the time to bring the states back to the origin and the energy consumed is the force input to the cart system. The LQR optimization equation (Winter, 2009) can be found below:

$$J = \int_{t_i}^{t_f} (x^T Q x + u^T R u) dt$$

The optimal solution to this equation is found by minimizing the value of the cost function, J. The matrix coefficients, Q and R represent the weight given to performance and force input, respectively. In other words, a Large Q and a small R demands a very fast performance while consuming a lot of energy. A small Q and a Large R indicates that the minimization of energy consumed is most important, meaning a small force and slow response time. The LQR controller can be designed to the users needs, in this case, the fastest performance possible is desired while keeping the maximum force input less than roughly 50 Newtons. This will ensure that the cart pendulum system does not demand that the force input is more force that the actuators are able to provide.

After modeling and running the system in Simulink, the Q and R matrices were tuned to give an optimal result for the performance and energy. The following coefficients were obtained upon iteration:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R = 1$$

Interestingly, it was found during tuning that the element of Q which corresponded to cart position (q33), had a much larger effect on the performance than any of the other elements in Q, hence this piece is 100 while the rest are 1. This makes sense since if the position of the cart is allowed to move large distances, then there is going to be much more time required to bring it back to the origin, since input is directly correspondent to acceleration, or dx/dt.

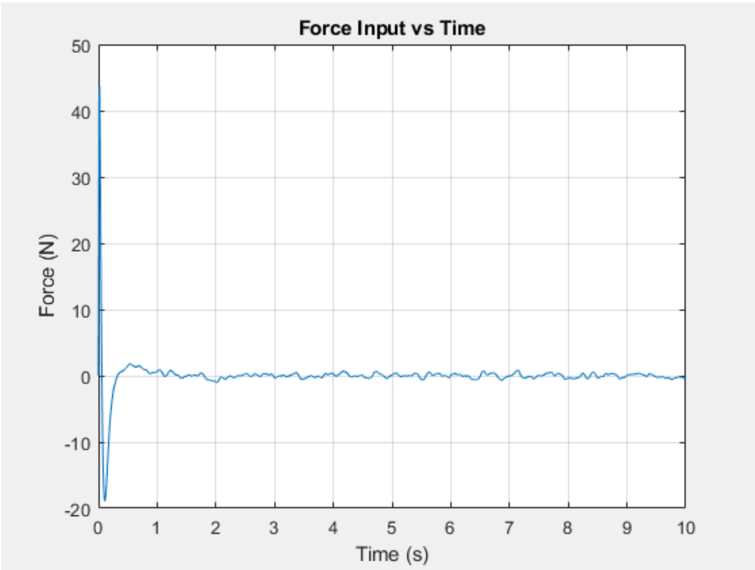
With the parameters above, the controller gain was found and put into the simulation, where the observer based compensator provided the input. The observer gain was found by placing the desired closed loop poles 10 times further to the left of the closed loop poles from the LQR system. From the simulation the following results were obtained.

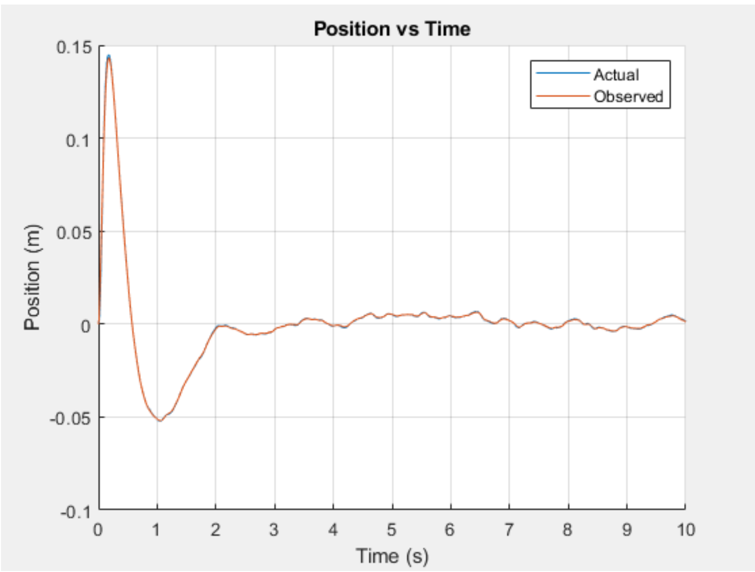
The observed and actual plots are almost identical, so the observer works fairly well with the poles ten times to the left. It also can be noted that the force input only hits a maximum of 43.6351 N, well within the actuator limits, while reaching a steady-state, upright pendulum at the origin within about 3 seconds.

3.3 Model Predictive control

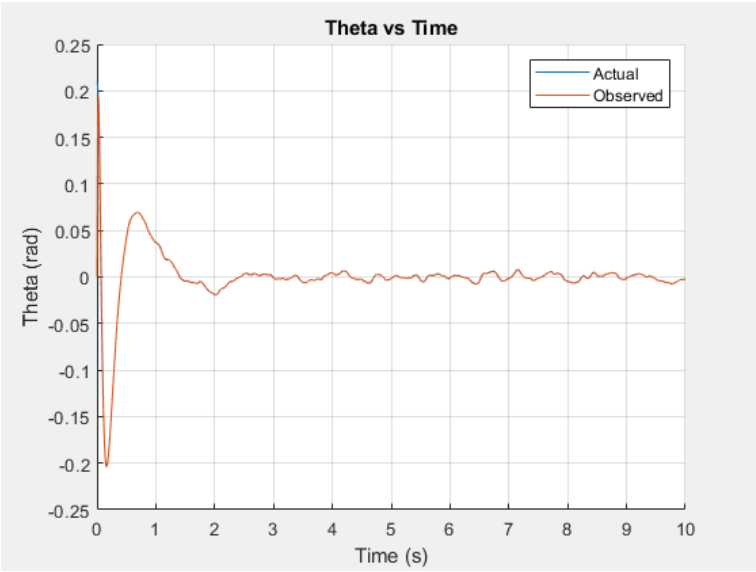
3.3.1 State-Space Model Predictive Control

Model Predictive Control (MPC) is a facet of control theory that explores the idea of optimizing control of a system for a finite time window and implementing this optimized control for a short period of time within this window before reevaluation. This process of optimization occurs at discrete time steps based on the current state of the system and generally repeats regularly over a desired time period. The attractiveness of a control





scheme like this is its robustness and ability to handle both "'soft' constraints and hard constraints" along with its "simplicity" (pp. xi). For the purposes of this project, the discrete-time MPC methodology was implemented to accomplish the following objectives:
1. Control the desired system model
2. Respond to an outside disturbance accurately
3. Approach a desired set point while responding to outside distrubances
This section of the report will detail the underlying principles of the discrete-time MPC methodology, the implementation of the MPC methods



on the desired system model, and the results corresponding to the above objectives.

3.3.2 Implementing the Desired System Model

The system model utilized in the development of the MPC controller was a discretized version of the original continuous-time system model. This was then converted into the augmented system referenced in the previous section. Using the same parameters for mass and length of the pendulum as previously stated in this report, after being discretized with a sampling rate of $T_s = 0.1s$, the following matrices were obtained,

$$A_m = \begin{bmatrix} 1.2505 & 0.1067 & 0 & 0 \\ 4.1812 & 1.2505 & 0 & 0 \\ -0.0101 & -0.0003 & 1 & 0.1 \\ -0.2091 & -0.0101 & 0 & 1 \end{bmatrix}$$

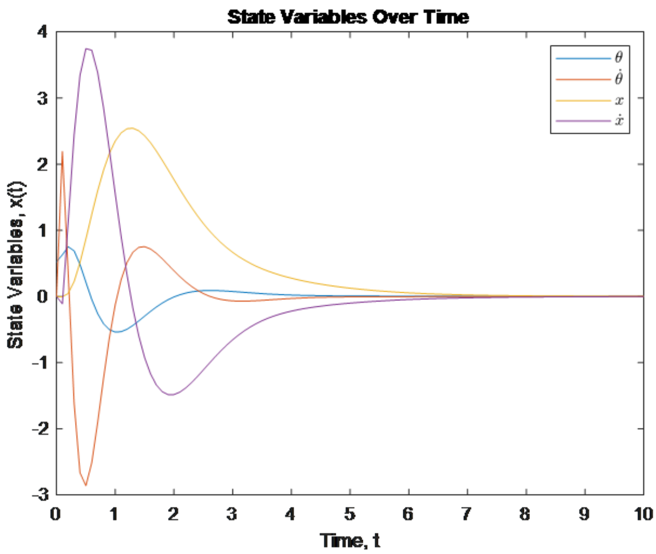
$$B_m = \begin{bmatrix} -0.0344 \\ -0.7111 \\ 0.0101 \\ 0.2022 \end{bmatrix}$$

$$C_m = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For this system, there are two tuning variables that allow for further influence over the system, (k_i) and r_w. The set point r(k_i) is used to direct the steady state of the system's output variable $y(k_i)$ as the system travels through time. This set point is used in the error calculation of the cost function J and therefore requires the states' trajectories to try to approach this set point value. The second tuning variable r_w acts as a weight to the cost function J that determines how much controller energy should be used in order to reach the desired system state. The larger the value of this weight, the more time it should take to approach the desired system state as is uses less controller energy to reach this value. This is due to the fact that this weight is applied to the Δu^2 term of the cost function J and therefore puts a greater emphasis on preventing larger changes in u from time step to time step.

3.3.3 Objectives and Results

To assess whether or not the main objectives of this section of the project have been met, the results from the simulations should be analyzed in detail. Concerning the first objective, "Control the Desired System Model", the requirement needed to achieve this objective include designing a controller that returns the system to a stable state in which all of the state variables approach zero. To assess this accurately, the values of all states were tracked and plotted against the corresponding times resulting in the graph below. The figure above shows the tendency of all state variables to

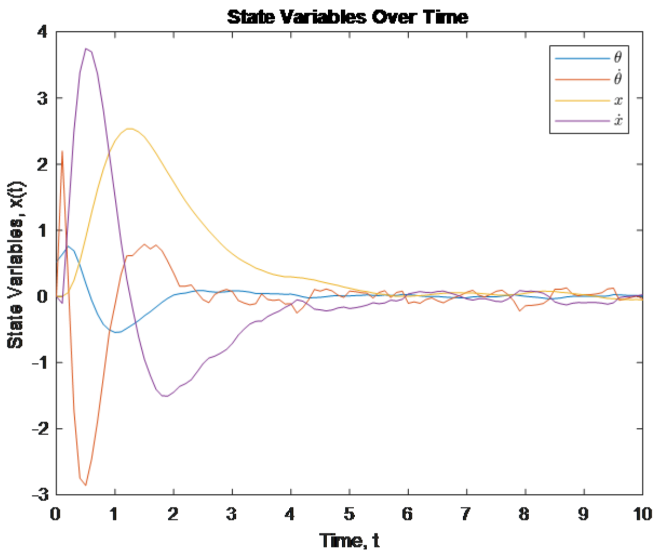


approach zero as t tends to infinity. This is regardless of the initial condition provided to the system within a reasonable magnitude. However, due to the fact that the system being used is a nonlinear system that has been linearized about zero for the purposes of this project, any large displacement from these zero points will create unpredictable results.

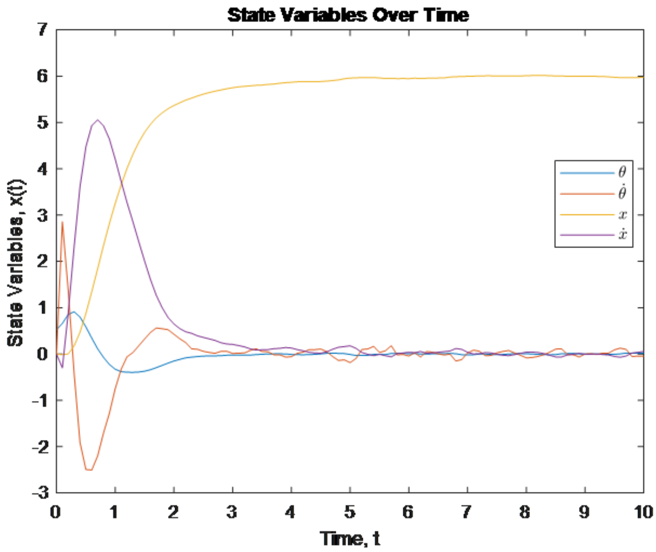
To build upon this (now controllable) system, the second objective calls for accurate response to outside disturbance. This objective necessitates that the system is able to remain controllable regardless of the introduction of random disturbance from the environment. In this simulation, this disturbance function is modeled as a disturbance added to the state variables rather than a disturbance due to inaccurate measurement. This changes the state-space equation used by the system to the following,

$$x(k+1) = A_m x(k) + B_m u(k) + B_d d(k) \tag{1}$$

where B_d is a constant and d(k) is a vector of normally distributed values. For the purposes of this simulation, the normal distribution was of the following form N(0,0.005). To assess the performance of the system under these conditions, the state variables were again plotted against the corresponding time, represented below. Figure 2. above demonstrates the



system is robust to disturbances and will hover around the zero value even when experiencing an outside disturbance. The values do not remain at zero, however they do stay well within a small range of their desired values. The final objective in the design of this system is to have the system approach a desired set point while responding to any outside disturbances introduced into the system. This is achieved directly through the minimization of the cost function J in which the error between the set point $R_s$and the output Y is being minimized. From this, a stable position can be accurately approached when setting the set point of the linear position of the cart to a certain value. This can be shown in the figure below, again, through plotting the state variables of the system. Figure 3. shows how the



system reacts to disturbances while also approaching the desired set point. As can be seen, all states approach zero except for the position of the cart which approaches the value designated by the set point. Ultimately, this final simulation demonstrates the controller's robustness as well as its success in meeting the objectives and requirements designed for this section of the project.

3.3.4 Reference

-[1] Wang, Liuping (2009). Model Predictive Control System Design and Implementation Using MATLAB. Springer Science and Business Media.