McGill University

ECSE 404 - Final Project

The Inverted Pendulum

Author:
Bernard Kaminski
Bernard.kaminski@mail.mcgill.ca

Contents

1 Problem Statement

The inverted pendulum is a classic control problem where a weighted mass ball is attached to rod which is connected with a hinge to a free moving cart that is on a track. This means that the movement of the rod and the cart are limited to two dimensions. The cart has an electric motor which allows in to move freely in order to balance the mass on top of the rod. The goal is to balance the mass over the cart. The ideal way to do this is to reach the ball's unstable equilibrium point above the cart then move the cart in response to the ball's movement in order to conserve this equilibrium. Thus some sort of control system must be implemented in order to conserve this equilibrium and product the correct electric motor outputs to achieve this. The following report will explore a pole-placement, PID and LQR controllers.

2 Mathematical Models

2.1 DC Motor

The first model that must be analyzed is the electric motor model. The force that is produced by the motor is the main input that controls the system, so we must derive a model to represent the force outputted by the electric motor. From page 19-21 of [1] the third order model can be seen to be the following.

$$v_a(t) = \frac{L_a J}{K_m}\frac{d^3\theta}{dt^3} + \left(\frac{R_a J + DL_a}{K_m}\right)\frac{d^2\theta}{dt^2} + \left(\frac{R_a D}{K_m} + K_b\right)\frac{d\theta}{dt} \tag{1}$$

In the above equation L_a and R_a are the internal inductance and resistance of the motor controller. The K_m and K_b terms are constants for motor torque and back emf. While the J and D terms are the external load constants for inertia and damping .

The model just described gives us the voltage but the force is needed. Thus out put force as a result of voltage. Page 3 of [2] shows how electro-mechanical relations in order to derive the linear force of a motor as a function of voltage v_a and the linear velocity $x'(t)$ of the cart

$$F = \frac{K_m K_g}{r R_A}v_a - \frac{K_m^2 K_g^2}{r^2 R_A}x'(t) \tag{2}$$

In the above equation K_m is the contant for motor torque, K_g is the gear ratio, R_A is the internal resistance of the motor, and r is the radius of the output gear.

2.2   Non-Linear Inverted Pendulum Model

The mathematical derivation for the problem can be seen on pages 25-28 of [1]. The position of the ball can be seen to be:

$$\vec{r_{ball_x}} = (x + l \sin \theta) \tag{3}$$

$$\vec{r_{ball_y}} = l \cos \theta \tag{4}$$

The potential and kenetic energy of the system must be found in order to derive the Lagrange function and can be seen to be:

$$T = T_{ball} + T_{cart} = \frac{1}{2} m_{ball} v_{ball}^2 + \frac{1}{2} m_{cart} v_{cart}^2 \tag{5}$$

$$= \frac{1}{2} m_{ball} (\frac{dx}{dt})^2 + \frac{1}{2} m_{cart} (\frac{d}{dt}(x + l \sin \theta)^2 + \frac{d}{dt}(l \cos \theta)^2) \tag{6}$$

$$= \frac{1}{2} m_{ball} (\frac{dx}{dt})^2 + \frac{1}{2} m_{cart} ((\frac{dx}{dt} + l \frac{d\theta}{dt} \cos \theta)^2 + (-l \frac{d\theta}{dt} \sin \theta)^2) \tag{7}$$

$$V = V_{\theta=90°} + mgl \cos \theta \tag{8}$$

In order to analyze the non-linear system it is modeled using Lagrange's principle. The Lagrangre equation can be seen to be:

$$L(q, \frac{dq}{dt}) = T(q, \frac{dq}{dt}) - V(q) \tag{9}$$

$$\begin{cases} \frac{d}{dt}(\frac{dL}{dx'}) - \frac{dL}{dx} = F_1 \\ \frac{d}{dt}(\frac{dL}{d\theta'}) - \frac{dL}{d\theta} = F_2 \end{cases} \tag{10}$$

Notice that in the above equation F_2 is the work when there is a change in θ as x is fixed and $F_2 = 0$. likewise, F_1 is the work done when there is a change in x with θ constant. This value will be non-zero when the cart is moving. This forced is denoted by F . Combining this with the necessary partial derivatives results in the the non-linear system model equations:

$$(m_{cart} + m_{ball}) \frac{d^2 x}{dt^2} + m_{ball} l \frac{d^2 \theta}{dt^2} \cos \theta - m_{ball} l (\frac{d\theta}{dt})^2 \sin \theta = F \tag{11}$$

$$\frac{d^2 x}{dt^2} \cos \theta + l \frac{d^2 \theta}{dt^2} - g \sin \theta = 0 \tag{12}$$

To solve the differential equations, $x(0), \frac{dx(0)}{dt}, \theta(0), \frac{d\theta(0)}{dt}$ must be known.

2.3   Linearized Inverted Pendulum Model

It is easy to see from the $(\frac{d\theta}{dt})^2$ term in the above equations that the system is non-linear. Non-linear systems are very difficult to place in state space from thus a linearization method will be applied. The goal of the controller is to stabilize the ball in the upright position thus the no-linear equation will be linearized around this point. This point is $\theta \approx 0°$, and θ increases in the clockwise direction.

At the nominal point of $\theta \approx 0°$, the small-angle approximation will be used:

$$\begin{cases} \cos\theta \approx 1 \\ \sin\theta \approx \theta \end{cases} \tag{13}$$

Using the small-angle approximation and a nominal point, the non-linear pendulum equations become:

$$(m_{cart} + m_{ball})\frac{d^2x}{dt^2} + m_{ball}l\frac{d^2\theta}{dt^2} = F \tag{14}$$

$$\frac{d^2x}{dt^2} + l\frac{d^2\theta}{dt^2} - g\theta = 0 \tag{15}$$

Linearization about this nominal point removes the quadratic terms and removes the sinusoidal terms, creating a much simpler linear representation which can be more easily placed in the state space model.

2.4   State Space Model

The state space representation of this system as defined in [1] has the following form:

$$\vec{x'}(t) = \begin{bmatrix} x'_1(t) \\ x'_2(t) \\ ... \\ x'_n(t) \end{bmatrix} = \begin{bmatrix} f_1(x(t), u(t), t) \\ f_2(x(t), u(t), t) \\ ... \\ f_n(x(t), u(t), t) \end{bmatrix} = A\vec{x} + B\vec{u} \tag{16}$$

$$\vec{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ ... \\ y_n(t) \end{bmatrix} = \begin{bmatrix} g_1(x(t), u(t), t) \\ g_2(x(t), u(t), t) \\ ... \\ g_p(x(t), u(t), t) \end{bmatrix} = C\vec{x} + D\vec{u} \tag{17}$$

the stae variable x is an n-dimensional vector, the out put variable y is a p-dimensional vector, and the input variable u is an m-dimensional vector. Therefore A is a nxn matrix, B is a nxm, C is a nxp, and D is a mxp matrix. In the case of the inverted pendulum there are 4 states, 2 outputs, and 1 input. Therefore $n = 4, p = 2, m = 1$ [1].

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x \\ \theta \\ x' \\ \theta' \end{bmatrix} \tag{18}$$

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x \\ \theta \end{bmatrix} \tag{19}$$

$$\vec{u} = \begin{bmatrix} u_1 \end{bmatrix} = F \tag{20}$$

By placing the linear pendulum system in terms of these state variables the state space representation can be obtained:

$$(m_{cart} + m_{ball})x_3' + m_{ball}lx_4' = u_1 \tag{21}$$

$$x_3' + lx_4' - gx_2 = 0 \tag{22}$$

In order to solve the equations $\vec{x'}(t)$ must be found. Since there are two unknowns and two equations the state space representations of x_3' and x_4' can be found.

$$x_3' = \frac{u_1 - m_{ball}lx_4'}{m_{ball} + m_{cart}} \tag{23}$$

$$x_4' = \frac{gx_2 - x_3'}{l} \tag{24}$$

$$x_3' = \frac{u_1 - m_{ball}(gx_2 - x_3')}{m_{ball} + m_{cart}} \tag{25}$$

$$x_3' = \frac{u_1 - m_{ball}gx_2}{m_{cart}} \tag{26}$$

$$x_4' = \frac{gx_2 - \left(\frac{u_1 - m_{ball}gx_2}{m_{cart}}\right)}{l} = \frac{m_{cart}gx_2 - m_{ball}gx_2 - u_1}{lm_{cart}} \tag{27}$$

The complete state space form of the pendulum system is now:

$$\vec{x'}(t) = \begin{bmatrix} x_1'(t) \\ x_2'(t) \\ x_3'(t) \\ x_4'(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m_{ball}g}{m_{cart}} & 0 & 0 \\ 0 & \frac{g(m_{cart} - m_{ball})}{m_{cart}l} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_{cart}} \\ \frac{-1}{m_{cart}l} \end{bmatrix} \begin{bmatrix} u_1(t) \end{bmatrix} = A\vec{x}(t) + B\vec{u}(t) \tag{28}$$

$$\vec{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} u_1(t) \end{bmatrix} = C\vec{x}(t) + D\vec{u}(t) \tag{29}$$

Remember that the input must be exspressed in terms of voltage. Noticing that $\vec{u_1}(t) = F$ and substituting the previously derived voltage to force equations:

$$\vec{x'}(t) = \begin{bmatrix} x_1'(t) \\ x_2'(t) \\ x_3'(t) \\ x_4'(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m_{ball}g}{m_{cart}} & \frac{-K_m^2 K_g^2}{m_{cart}r^2 R_A} & 0 \\ 0 & \frac{g(m_{cart} - m_{ball})}{m_{cart}l} & \frac{K_m^2 K_g^2}{m_{cart}r^2 R_A} & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_m K_g}{m_{cart}r R_A} \\ \frac{-K_m K_g}{m_{cart}lr R_A} \end{bmatrix} \begin{bmatrix} v_{ext}(t) \end{bmatrix} \tag{30}$$

$$y(\vec{t}) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} v_{ext}(t) \end{bmatrix} \tag{31}$$

3 Open-Loop Analysis

The system will be analyzed using open-loop analysis. This is the first step in understanding how the system behaves and will make it possible to find the poles, zeros, stability and controllability. After performing open-loop analysis a closed-loop control system will be possible to implement.

3.1 Transfer Functions

In order to find the systems poles, zeros and stability the transfer functions must be found. To simplify the mathematics, the linear representation is used. Taking the Laplace transform of 30 the following is obtained:

$$s^2 X(s) = \frac{-m_{ball}g}{m_{cart}}\theta(s) - \frac{K_m^2 K_g^2}{m_{cart}r^2 R_A}sX(s) + \frac{K_m K_g}{m_{cart}r R_A}V(s) \tag{32}$$

$$s^2\theta(s) = \frac{g\Delta m}{m_{cart}l}\theta(s) + \frac{K_m^2 K_g^2}{m_{cart}r^2 R_A}sX(s) - \frac{K_m K_g}{m_{cart}r R_A}V(s) \tag{33}$$

setting all initial conditions to zero and combining constant variables:

$$x(0) = \theta(0) = \frac{d\theta(0)}{dt} = \frac{dx(0)}{dt} = 0 \tag{34}$$

$$B \equiv \frac{K_m^2 K_g^2}{m_{cart}r^2 R_A} \tag{35}$$

$$C \equiv \frac{K_m K_g}{m_{cart}r R_A} \tag{36}$$

$$\Delta m \equiv m_{cart} - m_{ball} \tag{37}$$

By isolating the terms in 32 and 33 the open-loop transfer functions can be found to be:

$$H_1(s) = \frac{\theta(s)}{V(s)} = \frac{C(s - B(l-1))}{ls^3 + Bls^2 - \frac{g\Delta m}{m_{cart}}s + (\frac{gB}{m_{cart}} * (m_{ball}l - \Delta m))} \tag{38}$$

$$H_2(s) = \frac{X(s)}{V(s)} = \frac{C(s^2 - \frac{g}{m_{cart}l}(\Delta m - m_{ball}))}{s^4 + Bs^3 - \frac{g\Delta m}{m_{cart}l}s^2 - \frac{Bg}{m_{cart}}(m_{ball} + \Delta m)} \tag{39}$$

3.2   Constant Values

In order to proceed begin numerical computations With *MATLAB* values must be assigned to the constants in the equations derived. The values used in all computations and simulations can be found in the *MATLAB* workspace file `const_val.mat`.

3.3   Stability, Poles, and Zeros

To determine the stability of the system the poles and zeros of the system transfer functions must be found. This can be done using *MATLAB*. The figures 1 and 2 show the result of the *MATLAB* scripts *stability.m*. The figures show that both systems are unstable as both have poles in the right hand plane. H_1 is the function we are interested in stabilizing as that is the function that has θ with respect to the input voltage V.

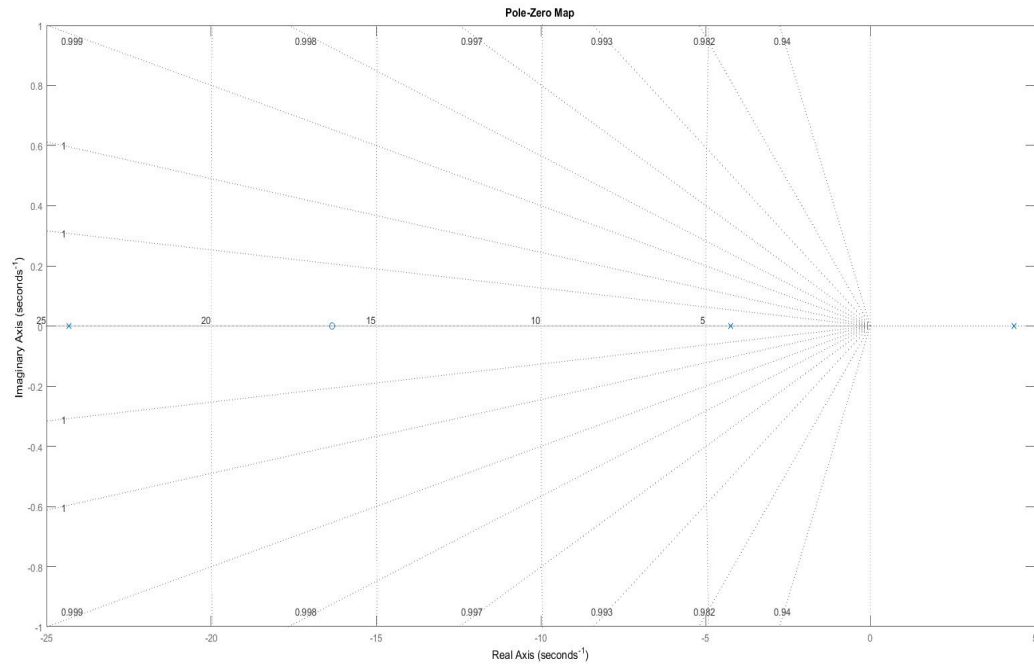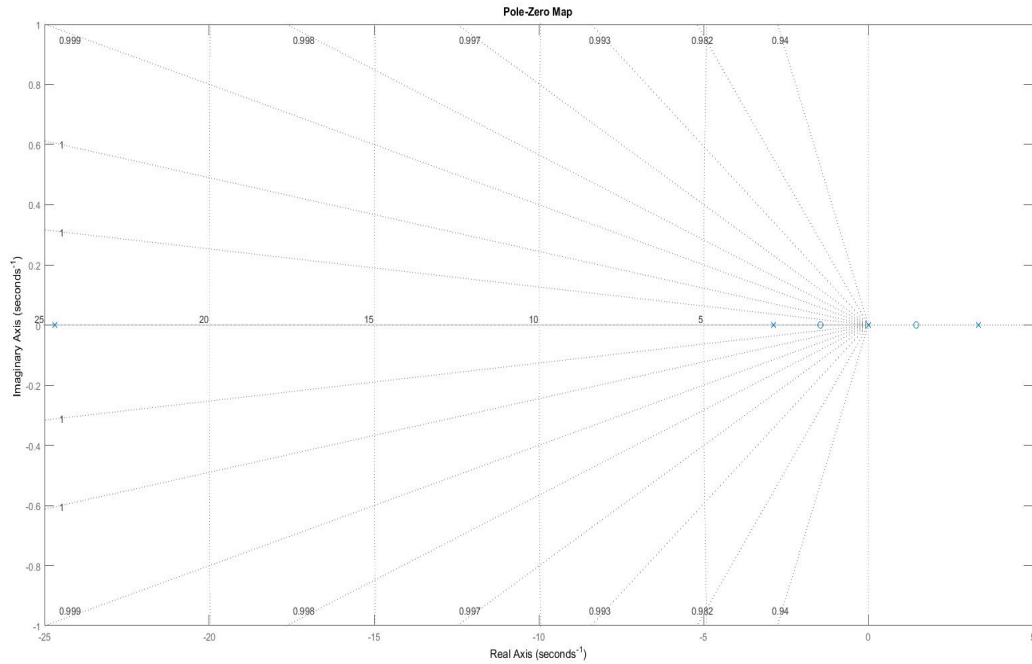Figure 1: Pole-Zero Plot of H_1 from `stability.m`

Figure 2: Pole-Zero Plot of H_2 from `stability.m`

3.4   Equilibrium Points

In order to find the equilibrium points of the open-loop linearized system the state space function in 30 , are set to equal zero, when there is no voltage applied. Solving for the zero condition the following result is obtained:

$$\begin{cases} x'_{1e} = x_3 = 0 \\ x'_{2e} = x_4 = 0 \\ x'_{3e} = -2.8340x_2 - 24.2077x_3 \\ x'_{4e} = 6.9760x_2 + 24.2077x_3 \end{cases} \tag{40}$$

From the above it can be seen that an equilibrium point exists when:

$$x_2, x_3, x_4 = 0 \quad \forall x_1 \in \mathbb{R} \tag{41}$$

3.5   Controllability

In order to determine the controllability of the controllability of the system the rank of the controllability matrix $\mathbb{C}$ must be determined and compared to the rank of the state space

matrix A state in equation 30. If $\mathbb{C}$ has a rank equal to A then the system is controllable.

$$\mathbb{C} = \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix} \tag{42}$$

$$\mathbb{C} = \begin{bmatrix} 0 & \frac{K_m K_g}{m_{cart} r R_A} & -\frac{K_m^2 K_g^2}{m_{cart}^2 r^3 R_A^2} & \frac{K_m^5 K_g^5}{m_{cart}^3 r^5 R_A^3} + \frac{K_m K_g mg}{m_{cart}^2 l r R_A} \\ 0 & -\frac{K_m K_g}{m_{cart} r R_A} & \frac{K_m^3 K_g^3}{m_{cart}^2 l r^3 R_A^2} & -\frac{K_m^5 K_g^5}{m_{cart}^3 l r^5 R_A^3} - \frac{g\Delta m K_m K_g}{m_{cart}^2 l^2 r R_A} \\ \frac{K_m K_g}{m_{cart} r R_A} & -\frac{K_m^3 K_g^3}{m_{cart}^2 r^3 R_A^2} & \frac{K_m^5 K_g^5}{m_{cart}^3 r^5 R_A^3} + \frac{mg K_m K_g}{m_{cart}^2 l r R_A} & -\frac{K_m^7 K_g^7}{m_{cart}^4 r^7 R_A^4} - \frac{mg K_m^3 K_g^3}{m_{cart}^3 l r^3 R_A^2} - \frac{K_m^3 K_g^3 mg}{m_{cart}^3 l^2 r^3 R_A^2} \\ -\frac{K_m K_g}{m_{cart} l r R_A} & -\frac{K_m^3 K_g^3}{m_{cart}^2 l r^3 R_A^2} & -\frac{K_m^5 K_g^5}{m_{cart}^3 l r^5 R_A^3} - \frac{g\Delta m K_m K_g}{m_{cart}^2 l^2 r R_A} & -\frac{K_m^7 K_g^7}{m_{cart}^4 r^7 R_A^4} - \frac{K_m^7 K_g^7}{m_{cart}^4 l^2 r^7 R_A^4} + \frac{K_m^3 K_g^3 mg}{m_{cart}^3 l^2 r^3 R_A^2} \end{bmatrix} \tag{43}$$

The controlabillity of this matrix was foun using *MATLAB*.This was done by running simulations as row reduction would be too difficult.The *MATLAB* output of `openLoopControllability.` shows that the system is controllable.

3.6   Observability

The observability matrix $\mathbb{O}$ is used in order to determine if the system is observable. If the rank of $\mathbb{O}$ is equal to the rank of the state space matrix A, then the system is observable.

$$\mathbb{O} = \begin{bmatrix} C & CA & CA^2 & CA^3 \end{bmatrix}^T \tag{44}$$

$$\mathbb{O} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m_{ball} g}{m_{cart}} & -\frac{K_m^2 K_g^2}{m_{cart} r^2 R_A} & 0 \\ 0 & \frac{g\Delta m}{m_{cart} l} & \frac{K_m^2 K_g^2}{m_{cart} l r^2 R_A} & 0 \\ 0 & \frac{K_m^2 K_g^2 m_{ball} g}{m_{cart}^2 r^2 R_A} & \frac{K_m^4 K_g^4}{m_{cart}^2 r^4 R_A^2} & -\frac{m_{ball} g}{m_{cart}} \\ 0 & -\frac{K_m^2 K_g^2 m_{ball} g}{m_{cart}^2 l r^2 R_A} & -\frac{K_m^4 K_g^4}{m_{cart}^2 l r^4 R_A^2} & \frac{g\Delta m}{m_{cart} l} \end{bmatrix} \tag{45}$$

It can be seen that the matrix in 45 has a rank of four by inspection. This can also be confirmed using the *MATLAB* script `openLoopObservability.m`. Therefore the open-loop system is observable.

4   Linear Controllers

The following section will discuss the design, simulation and analysis of controllers that can be used to stabilize the system in the upright position $\theta \approx 0°$. The output variable that we are interested in controlling is θ and not x as the linear position of the cart has complete

freedom in this control problem. This is why the only transfer function of interest is $H_1(s)$ and the only time response of interest is $\theta(t)$.

4.1   Pole-Placement

The previous section 3.3 showed that the open-loop state transfer function $H_1(s)$ is unstable because there exists a pole in the right- hand side of the s-plane. In the following section a one degree of freedom pole placement controller will be examined in order to control the closed loop system.

In order to create a one degree of freedom (ODOF) pole placement controller the transfer function $H_1(s)$ must be modified using a feedback loop and design blocks in order to create a stable closed-loop transfer function. The figure 3 represents the control system. In order to simplfy the problem, the design block $\frac{h(s)}{k(s)}$ is considered to be a proportional-

Figure 3: Block Diagram [3]
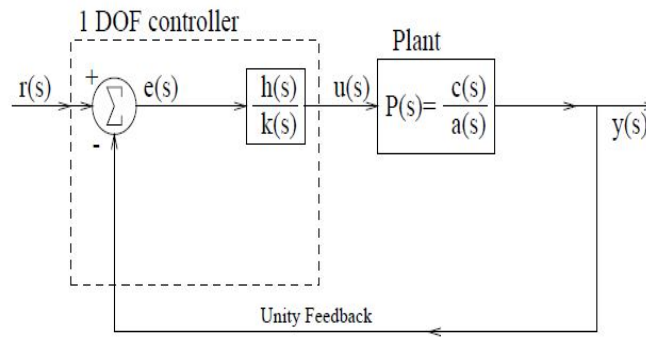
ity constant, K, with no powers of s. This simplifies the analysis, yielding a closed-loop transfer function of:

$$T(s) = \frac{Kc(s)}{a(s) + Kc(s)} \tag{46}$$

In order to create the controller, the parameter K must be edited. The script `poleplacement.m`prints the threshold value of k that make the system stable. The script also displays the poles with that K value and the R-LocSus graph. The R-Locus graph shows how the poles will move as the value of K changes. The R-Locus graph can be seen in figure 4 . Looking at the plot it shows that for many values of K the system will remain unstable. But if K gets large enough the system will become stable as no poles will be in the right hand plane. The correct value of k was tested for and it was found that its threshold value was:

$$K \geq 1.48 \tag{47}$$

The stable pole-zero plot of the closed-loop system when $K = 1.48$ can be seen in figure 5.

Figure 4: RLocus
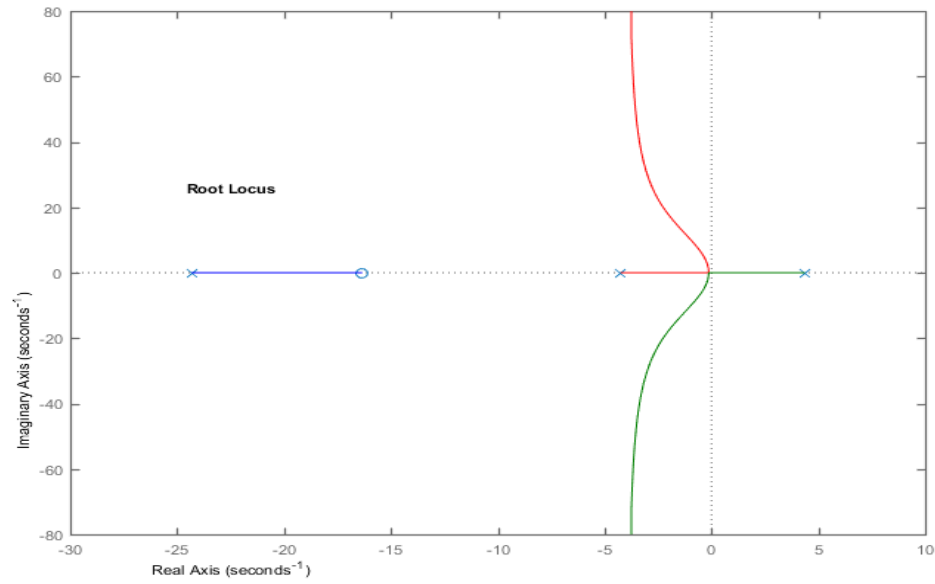
Figure 5: Pole-Zero Pot of $T(s)$ when $K = 1.48$
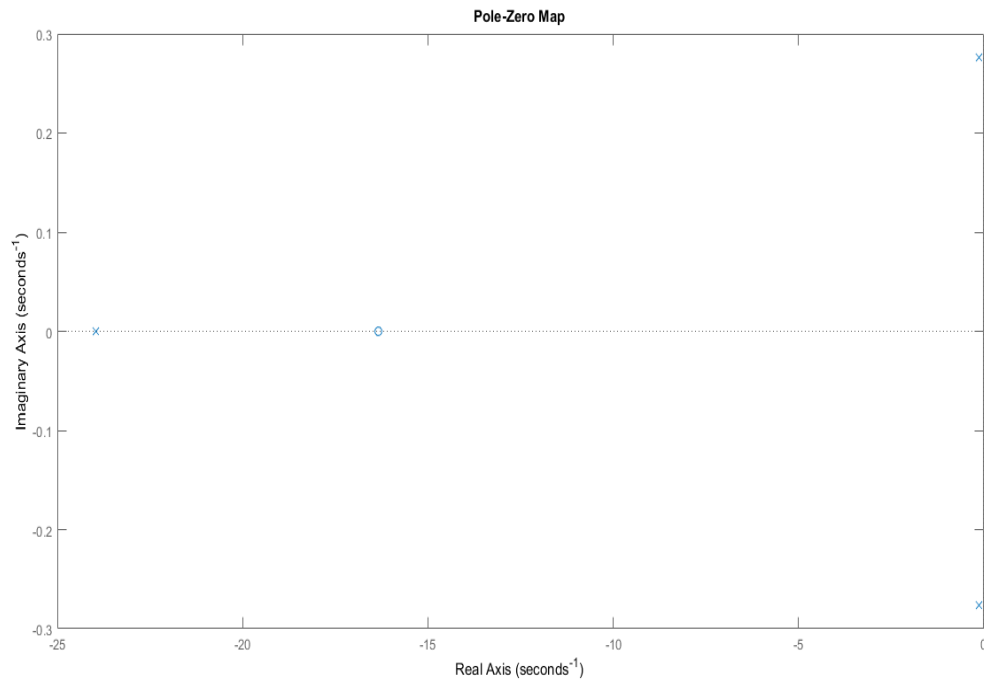
4.2   PID Controller

The following section will describe the design and the analysis of a proportional-integral-derivative (PID) controller with a single degree of freedom. A PID controller calculates the mount of error between the desired point ($\theta \approx 0°$) and the state (θ). The controller attemps to minimize the error by changing the process based on the amount of error, the past error and the future error through a negative feedback action. The single degree of freedom controller discussed in the following section is implemented in the script `pidController.m`, which produces tuned output plots. A single degree PID system can be visualized in figure 6. This controller is similar to the single degree of freedom pole

Figure 6: PID One DOF Block Diagram [5]
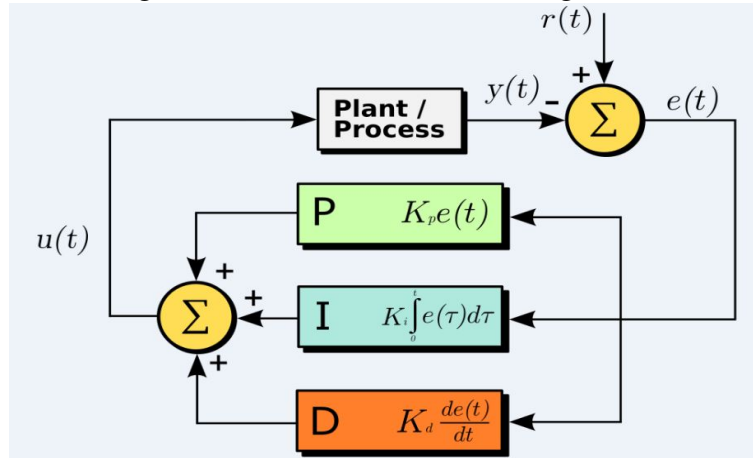
placement controller as only one design block is placed with the open-loop system. This one block is the only one that can be modified to achieve the stability desired. The block is the sum of three weighted error relations and is described mathematically in [3] in the following way:

$$PID(s) = K_p + \frac{K_i}{s} + K_d s \tag{48}$$

The terms K_p, , K_i, K_d are the constants that are used in order to control the weights of the expression. Using the PID transfer function and placing it in a negative feedback loop with the open-loop transfer function $H(s)$ obtained earlier, produces the following closed-loop transfer function:

$$T(s) = \frac{H(s)}{1 + PID(s)H(s)} = \frac{c(s)}{a(s) + PID(s)c(s)} \tag{49}$$

As the open-loop transfer function as found earlier has three poles and one zero ,

$$H(s) = \frac{c(s)}{a(s)} = \frac{z_1 s + z_0}{p_3 s^3 + p_2 s^2 + p_1 s + p_0} \tag{50}$$

Therefore the closed-loop transfer function is given by:

$$T(s) = \frac{z_1 s^2 + z_0 s}{p_3 s^4 + (p_2 + z_1 K_d)s^3 + (p_1 + z_1 K_p + z_0 K_d)s^2 + (p_0 + z_1 K_i + z_0 K_p)s + (z_0 K_i)}$$
(51)

Now this controller must be tuned. This means that the weights of K_p, K_i, and K_d must be manipulated until a stable input response with a short settling time and a small deviation from the desired point of operation. At first K_i, and K_d where set to 0 and K_p was set to the threshold value found in the previous section 4.1. The results where as expected creating a stable impulse response for $K_p \geq 1.48$. From this point the weights where gradually increased to improve the response. To increase the period of oscillations K_p was increased. To reduce settling time K_i was increased, and to reduce the peak deviation K_d was increased [6]. Figure 7 shows the impulse response with the weight parameters tuned values.

Figure 7: Impulse Response of Tuned PID Controller $K_p = 250, K_i = 5, K_d = 30$
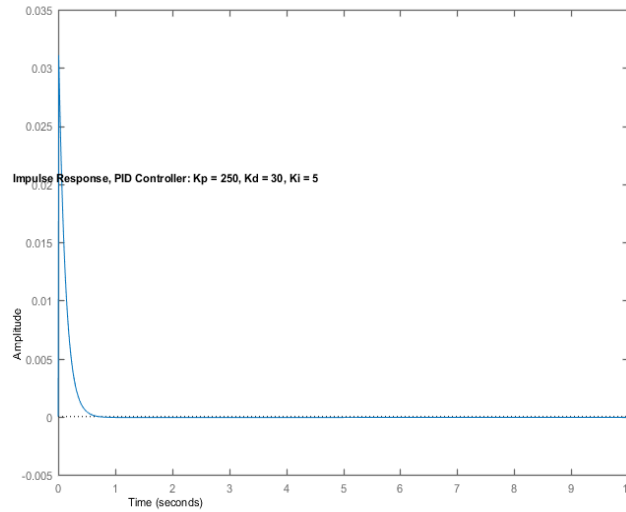
The parameter values that were chosen are $K_p = 250, K_i = 5, K_d = 30$. These values produced a stable controller with settling time close to .717 a deviation from the desired value of about .031 radians.

4.3   LQR Controller

A linear quadratic regulator controller is a controller that has a full state feedback system. This allows for the stabilization of the unstable open-loop system. This means that the un-stable system transfer function $H_1(s)$ can be stabilized. If the system was not observable it would not have been possible to implement the LQR controller for the stabilization of the

system.

Figure 8: LQR Controller Block Diagram [7]
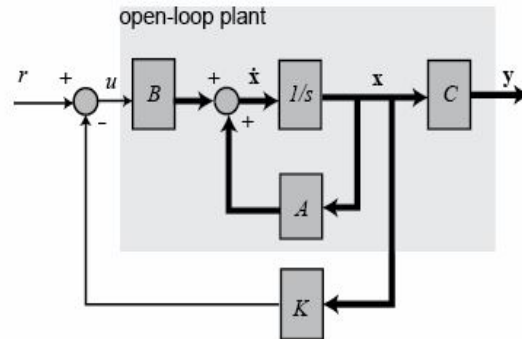
An LQR controller is used to determine the values of a system's weight parameter on the linear and angular position Q against the amount of error that is allowed in the system R [7]. When these weights are decided the controller can compute the values for the gain matrix K. The gain matrix is then used to compute a new state space model where the A matrix is altered through the subtraction by the B matrix multiplied by K. The B, C and D matrices remain the same.

In *MATLAB* the `lqr` function can be used for this process. As recommended by [6], the easiest way to begin picking values for R and Q is to start by assigning R a value of one. These values are arbitrary and what is actually important is the ratio seen between Q and R. This ratio will determine the weight of importance put on position and allowable error. Seeing as this project does not stress to much importance on the linear position, the weight of this was lowered in order to account for more accuracy elsewhere. This value is seen at the $(1, 1)$ position of the Q matrix. The $(2, 2)$ position of the Q matrix, which pertains to angular position importance, was raised substantially in order to obtain a better balancing system. The step response of the system utilizing the LQR controller and the system not utilizing it can be seen below in figures 11 and 10. As can be seen, the fluctuation in angular and linear position decreases heavily as the values of Q have been raised, showing that this is overall a very effective, balancing controller. The *MATLAB* implementation of the LQR controller discussed here can be found in `InvertedPend_LQR.m`.

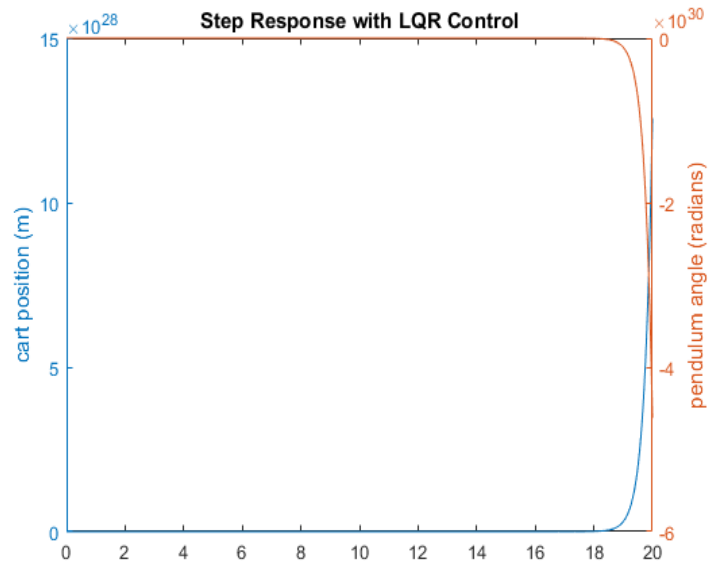Figure 9: Unstable Open-Loop Step Response without LQR

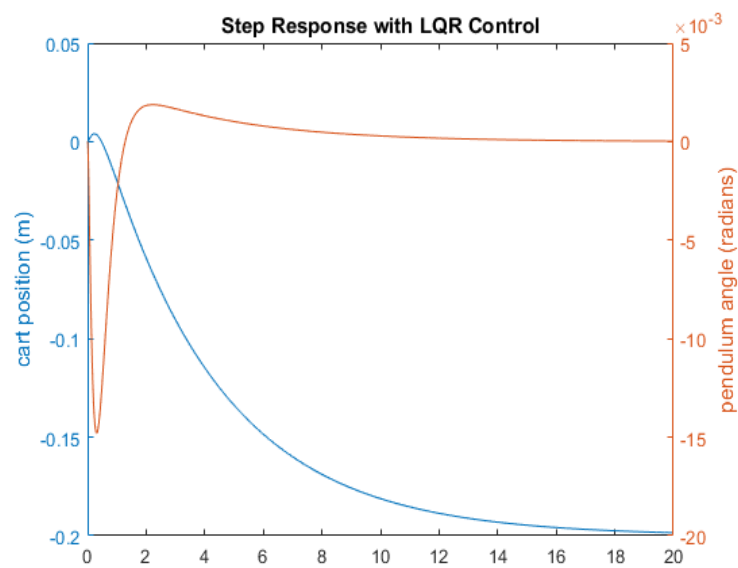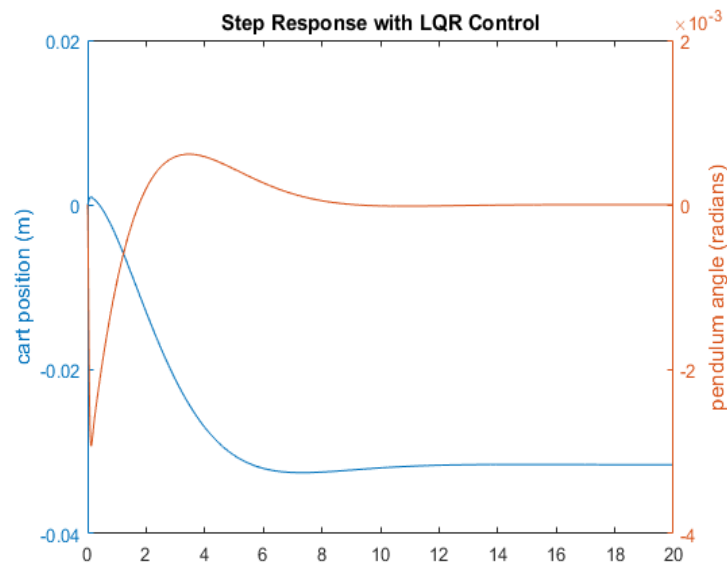Figure 10: Stable Closed-Loop Step Response with low ratio LQR

Figure 11: Unstable Open-Loop Step Response high ratio LQR

4.4 Discussion of Results

One of the controllers must be chosen in order to stabilize the system. There are several advantages and disadvantages to the controllers. The properties that will help guide which one should be chosen are stability, gain margin, phase margin, settling time, max deviation from set-point, and practicality. The best performance comes from the PID and LQR controllers which have short settling times and small deviations from the desired operating point.

4.5 Stabilization Neighborhood

5 Conclusion

The inverted pendulum control problem can be stabilized using various controllers. Each controller has its trade-offs and advantages. The most challenging aspect of the design is fine tuning the parameters of the controllers in order to obtain good stabilizations. It was concluded that the best controllers where the LQR and PID controllers as they could provide stabilization in different circumstances with little time and little deviation from the desired point. Though the PID is concluded to be better as it transitions well with the non-linear model.

6   Works Cited

[1] Michalska, H., ECSE 404: 304-404 section 1. Obtained from McGill University MyCourses. Fall 2014

[2] Technische Universität München, School of Computational Engineering., Pendulum Project [Online]. Available: http://www5.in.tum.de/wiki/index.php/Pendulum_Project

[3] Michalska, H., ECSE 404: 304-404 section 6. Obtained from McGill University MyCourses. Fall 2014

[4] Michalska, H., ECSE 404: 304-404 section 7. Obtained from McGill University MyCourses. Fall 2014

[5] Wikipedia., PID Controller [Online]. Available: en.wikipedia.org/wiki/PID_controller

[6] University of Michigan, School of Engineering., Extras: Designing Lead and Lag Compensators [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?aux=Extras_Leadlag

[7] University of Michigan, School of Engineering., Inverted Pendulum: State-Space Methods for Controller Design [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlStateSpace