We aim to develop a digital controller for complete control of a Rotary Inverted Pendulum for getting a better understanding of the controller design process.
Table of Contents

Table of Figures .................................................................................................................. 3
Table of Equations .................................................................................................................. 3
1. Introduction ....................................................................................................................... 5
2. Physical System Analysis ................................................................................................. 5
3. Analysis of Digital System ............................................................................................... 7
   a. Selection of Sampling Time ......................................................................................... 8
   b. Digital Controller for Alpha ...................................................................................... 9
   c. Digital Controller with two feedbacks, $\theta$ and $\alpha$ ............................................ 15
4. Simulation in Simulink ..................................................................................................... 21
   a. Non-Linear Model of Actual System ........................................................................ 21
   b. Linear Model at Unstable Position ............................................................................ 22
   c. Stabilization Controller ............................................................................................ 22
   d. Destabilization Controller ....................................................................................... 24
   e. Mode Controller ....................................................................................................... 28
   f. Overall Model and Performance ............................................................................... 30
5. LabView Models and Physical Implementation ............................................................. 31
   a. Controller for Alpha only ......................................................................................... 31
   b. Controller for both Alpha and Theta (destabilization included) ............................. 33
6. Conclusion ......................................................................................................................... 35
7. References ........................................................................................................................ 36
8. Appendix ............................................................................................................................ 36
   a. Matlab Code for stabilization controller .................................................................... 36
   b. Matlab Code for Destabilization Controller ............................................................. 37
   c. LabView Codes for Alpha Controller Only ............................................................... 39
      I. Alpha Controller ...................................................................................................... 39
      II. Saturation Controller ............................................................................................ 39
   d. LabView Codes for Cascade Controller .................................................................... 39
      I. Alpha Stabilization Controller ............................................................................... 39
      II. Theta Stabilization Controller ............................................................................... 39
      III. Alpha,(theta destabilization is included) Destabilization Controller .................. 39
      IV. Mode Controller .................................................................................................. 39
      V. Saturation Controller ............................................................................................ 40
      VI. Modulus Operator ............................................................................................... 40
**Table of Figures**

Figure 1 - Root locus for alpha/Vm open loop ......................................................... 10  
Figure 2-Angle deficiency calculation ........................................................................ 12  
Figure 3-Root locus of control system with controller of $\alpha$ .................................. 13  
Figure 4-Impulse Response of Closed loop system for Pendulum angle $\alpha$ ........... 14  
Figure 5-Large Theta variations from system, implementing only one feedback $(\alpha)$ ................................................................. 15  
Figure 6-Bode Diagram of open loop system(Inner control loop in series with $G\theta, \alpha(z)$) ................................................................. 17  
Figure 7-Root locus of final system ........................................................................... 18  
Figure 8-Bode diagram of final system ..................................................................... 19  
Figure 9-Impulse response of $\theta$ ........................................................................... 20  
Figure 10-Non Linear Model - Simulink .................................................................. 21  
Figure 11 - Linear Model - Simulink ....................................................................... 22  
Figure 12 - Stabilization Controller ......................................................................... 23  
Figure 13 –Alpha Stabilization Simulation ................................................................. 23  
Figure 14 - Theta Stabilization Simulation ................................................................. 24  
Figure 15 - Control Voltage Stabilization Simulation .............................................. 24  
Figure 16-Impulse response of $\alpha$ from destabilization controller ......................... 26  
Figure 17- Impulse Response of $\theta$ for destabilization controller ......................... 27  
Figure 18 -Destabilization Controller - Simulink ..................................................... 28  
Figure 19 - Model Comparison Circuit – Simulink .................................................... 28  
Figure 20- Linear v/s Non-linear model - Simulink .................................................... 29  
Figure 21 - Mode Controller - Simulink .................................................................. 29  
Figure 22 - Mode Controller Sample Run - Simulink .............................................. 30  
Figure 23 - Overall Model - Simulink ...................................................................... 30  
Figure 24 – Block Diagram - Alpha Controller only – LabView ............................ 31  
Figure 25 - Block Diagram - Overall System – LabView ......................................... 33  
Figure 26 - Front Panel - Stabilization of System – Labview ..................................... 34  
Figure 27 - Front Panel - Destabilization of System - LabView ............................... 35  

**Table of Equations**

Equation 1 – differential equation $\theta$ .................................................................... 7  
Equation 2– differential equation $\alpha$ ..................................................................... 7  
Equation 3-Laplace transform of eq1 ......................................................................... 7  
Equation 4-Laplace transform of eq2 ......................................................................... 7  
Equation 5-Transfer function of $G_{a,Vm}(s)$,symbolic ........................................... 7  
Equation 6-Transfer Function $G_{\theta,a}(s)$,symbolic ............................................... 8  
Equation 7-Transfer function of $G_{a,Vm}(s)$,numerical ............................................ 8  
Equation 8 Transfer Function $G_{\theta,a}(s)$,numerical ............................................... 8  
Equation 9- $Ga, Vm(z)$ ............................................................................................ 9  
Equation 10-Open Loop Transfer function of $\alpha$ control system ......................... 12  
Equation 11-Difference equation of inner loop controller($C1(z)$) .......................... 12  
Equation 12- $G\theta, \alpha(z)$ ....................................................................................... 16  
Equation 13-C2(z) .................................................................................................. 18  
Equation 14-Difference equation for outer loop controller ...................................... 18  
Equation 15- $\alpha(z)/ Vm(z)$ for reference of $\alpha = \pi$ ............................................. 25
1. Introduction
The Rotary Inverted Pendulum is a classic control problem that is explored often as a project in control courses due to its easily developed dynamics combined with its complexity of control design. In this project, we delve into the digital control of this system for achieving the control objectives.

This system is composed of a pendulum attached to the end of a rotary arm controlled by a motor. The motor we use is a Quanser SRV-02 Plant, whose characteristics are known completely. The motor itself consists of a servomotor coupled with a gear-chain. The main objective is to keep the pendulum in upright position of unstable equilibrium. The secondary aim is to keep the motor at a particular provided angular position while performing the primary task. The final task is to destabilize the motor from the hanging down position of unstable equilibrium so that it eventually comes into the stable range, where a mode controller can kick in to initiate stabilization.

2. Physical System Analysis
The pendulum is referred to as $\alpha$ and the motor angle is referred to as $\theta$. The references are $\alpha=0$ at the upright position and $\theta=0$ at the starting position of the motor (marked on the motor). However, for the destabilization controller, the reference is $\alpha=0$ at the down most position.

We first proceed to derive the equations of motion. The main dynamic model is shown below:

We can formulate the system dynamics equation based on Euler-Lagrange formulation. We obtain the Potential and kinetic energies of the system as follows,
Potential energy,

\[ V = P \cdot E_{\text{Pendulum}} = mgL\cos\alpha \]

Kinetic energy,

Complete kinetic energy of our system is taken as

\[ T = \frac{1}{2} \cdot J_{eq} \dot{\theta}^2 + \frac{1}{2} \cdot m \left( r \dot{\theta} - L\cos\alpha (\dot{\alpha}) \right)^2 + \frac{1}{2} \cdot m \left( -L\sin\alpha (\dot{\alpha}) \right)^2 + \frac{1}{2} \cdot J_{eq} \dot{\alpha}^2 \]

We can formulate Lagrangian as follows

\[ L = T - V = \frac{1}{2} \cdot J_{eq} \dot{\theta}^2 + \frac{2}{3} \cdot mL^2 \dot{\alpha}^2 - mLr\cos\alpha (\dot{\theta}) + \frac{1}{2} \cdot mr^2 \dot{\theta}^2 - mgL\cos\alpha \]

Our two generalized co-ordinates are \( \theta \) and \( \alpha \). We therefore have 2 equations:

\[ \frac{\delta}{\delta t} \left( \frac{\delta L}{\delta \theta} \right) - \frac{\delta L}{\delta \theta} = T_{\text{output}} - B_{eq} \dot{\theta} \]

Equation 1 - Lagrangian Formulation (\( \theta \))

And

\[ \frac{\delta}{\delta t} \left( \frac{\delta L}{\delta \alpha} \right) - \frac{\delta L}{\delta \alpha} = 0 \]

Equation 2 - Lagrangian Formulation (\( \alpha \))

Output torque (\( T_{\text{output}} \)) on the load from the motor is:

\[ T_{\text{output}} = \frac{\eta_m \eta_g K_t K_g (V_m - K_g K_m \theta)}{R_m} \]

Solving above equations we get

\[
\begin{bmatrix}
\theta \\
\alpha \\
\theta' \\
\alpha'
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
b \cdot \frac{d}{E} & -\frac{cG}{E} & 0 & 0 \\
0 & \frac{b \cdot G}{E} & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\theta \\
\alpha \\
\theta' \\
\alpha'
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0 \\
\frac{c \cdot \eta_m \cdot \eta_g \cdot K_t \cdot K_g}{R_m \cdot E} \\
\frac{b \cdot \eta_m \cdot \eta_g \cdot K_t \cdot K_g}{R_m \cdot E}
\end{bmatrix}
\begin{bmatrix}
V_m \\
\end{bmatrix}
\]

Where,

\[
a = J_{eq} + mr^2 \\
b = mLr \\
c = \frac{4}{3} mL^2 \\
d = mgL
\]

\[
E = ac - b^2 \\
G = \frac{\eta_m \eta_g K_t K_m^2 + B_{eq} R_m}{R_m}
\]
Motor parameters are as per LAB assignments, can be seen in MATLAB code in Appendix.

3. Analysis of Digital System

The block diagram of our linearised model is as follows

We need to find out transfer functions for $G_{a, Vm}$ and $G_{\theta, \alpha}$. The state space representation of our system is as described earlier,

$$
\begin{bmatrix}
\theta \\
\alpha \\
\dot{\theta} \\
\alpha' \\
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & b \cdot d & -cG \cdot E & 0 \\
0 & a \cdot d & b \cdot G \cdot E & 0
\end{bmatrix}
\begin{bmatrix}
\theta \\
\alpha \\
\dot{\theta} \\
\alpha'
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0 \\
c \cdot \eta \cdot m \cdot \eta \cdot Kt \cdot Kg \\
Rm \cdot E
\end{bmatrix}
\begin{bmatrix}
Vm
\end{bmatrix}
$$

For simplicity we write above equation as follows

$$
\begin{bmatrix}
\theta \\
\alpha \\
\dot{\theta} \\
\alpha'
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & K1 & K2 & 0 \\
0 & K3 & K4 & 0
\end{bmatrix}
\begin{bmatrix}
\theta \\
\alpha \\
\dot{\theta} \\
\alpha'
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0 \\
K5 \\
K6
\end{bmatrix}
\begin{bmatrix}
Vm
\end{bmatrix}
$$

The above matrix equation can be decoupled in terms of $\dot{\theta}$, $\dot{\alpha}$, $\dot{\theta}$ and $\dot{\alpha}$

$$
\dot{\theta} = K1 \alpha + K2 \dot{\theta} + K5 V_m
$$

Equation 3 – differential equation $\dot{\theta}$

And

$$
\dot{\alpha} = K3 \alpha + K4 \dot{\theta} + K6 V_m
$$

Equation 4 – differential equation $\dot{\alpha}$

Taking Laplace transform of eq (1) and (2), and assuming initial value of all the states to be zero,

$$
s^2 \theta(s) = K1 \alpha(s) + K2 s \theta(s) + K5 V_m(s)
$$

Equation 5 – Laplace transform of eq1

And

$$
s^2 \alpha(s) = K3 \alpha(s) + K4 s \theta(s) + K6 V_m(s)
$$

Equation 6 – Laplace transform of eq2

Solving for $\alpha(s)/V_m(s)$ from the above two equations and eliminating $\theta(s)$ ,

$$
\frac{\alpha(s)}{V_m(s)} = \frac{K6 s^2 + (K4 K5 - K2 K6) s}{s^4 - K2 s^3 - K3 s^2 + (K3 K2 - K1 K4) s}
$$

Equation 7 – Transfer function of $G_{a, Vm} (s)$, symbolic
Solving for $\theta(s)/\alpha(s)$ from equations (2) and (3), eliminating $V_m(s)$

$$\frac{\theta(s)}{\alpha(s)} = \frac{s^2 + \left(\frac{K_1K_6}{K_5} - K_3\right)}{s^2 + \left(\frac{K_4}{K_5} - K_2K_6\right)}$$

Equation 8-Transfer Function $G_{\theta,\alpha}(s)$, symbolic

Substituting numerical values in equation (5)

$$\frac{\alpha(s)}{V_m(s)} = \frac{33.51 s^2}{s^4 + 22.52 s^3 - 91.17 s^2 - 945.6 s}$$

Equation 9-Transfer function of $G_{\alpha,V_m}(s)$, numerical

(The term $(K_4K_5 - K_2K_6)$ is very small, of order $10^{-13}$, hence we ignored the $(K_4K_5 - K_2K_6)s$ term of numerator).

Substituting numerical values in equation (6)

$$\frac{\theta(s)}{\alpha(s)} = \frac{39.1 s^2 - 1642}{33.51 s^2}$$

Equation 10 Transfer Function $G_{\theta,\alpha}(s)$, numerical

Again the term involving $(K_4K_5 - K_2K_6)$ in the denominator is ignored due to very less value (1.137e-013)

a. Selection of Sampling Time

The matrix $A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{b \cdot d}{E} & -\frac{cG}{E} & 0 \\ 0 & \frac{a \cdot d}{E} & -\frac{b \cdot G}{E} & 0 \end{bmatrix}$, plays very important role in controller design.

For state equations

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

The transformation matrix is given as

$$\hat{G}(s) = C(sI - A)^{-1}B + D$$

Or

$$\hat{G}(s) = \frac{1}{\det(sI - A)}C[\text{adj}(sI - A)]B + D$$

Hence, every pole of the system is an eigen value of $A$. Thus if every eigen value of $A$ (in continuous domain) has a negative real part, the system is BIBO stable.

For our system eigen values of $A$ are diagonal vector of following matrix.
\[
E = \\
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & -24.7510 & 0 & 0 \\
0 & 0 & 7.5340 & 0 \\
0 & 0 & 0 & -5.2983 \\
\end{array}
\]

Clearly one eigen value is non-negative. Our system is unstable at the moment. The desired modal damping rates and damped natural frequencies are simply the real and imaginary parts of the eigen values. We have selected 4 times the highest damped natural frequency as our sampling frequency.

\[
[V,E]=\text{eig}(A) \\
\text{ws\_Hz}=4*\text{ceil}(\text{max}(\text{max}(\text{abs}(E))))), \% \text{4 times so that no aliasing.} \\
T=1/\text{ws\_Hz}\% (=0.01 \text{ sec})
\]

**b. Digital Controller for Alpha**

Controller specification:

\( M_p<5\% \)

\( \text{Settling Time}<0.5\sec \)

We will first design our controller by taking \( \alpha \) as the only feedback. The proposed block diagram is as follows

\[
G_{\alpha,vm}(z) = (1 - z^{-1}) \frac{G_{\alpha,vm}(s)}{s}
\]

Using MATLAB c2d command we can find, \( G_{\alpha,vm}(z) \) with sampling time of 0.01ms

\[
G_{\alpha,vm}(z) = \frac{0.001558 z^2 - 0.0001126 z - 0.001445}{z^3 - 2.807 z^2 + 2.605 z - 0.7984} \\
\Rightarrow G_{\alpha,vm}(z) = \frac{0.001555(z - 1)(z + 0.9277)}{(z - 1.0763)(z - 0.9491)(z - 0.7815)}
\]

Equation 11-\( G_{\alpha,vm}(z) \)

We can calculate the location of dominant closed loop poles, according to controller specifications.

We will use \( 2^{nd} \) order continuous system approximation for calculation of dominant closed loop pole location.

For continuous domain \( 2^{nd} \) order system

\[
M_p = e^{\frac{-\zeta \pi}{\sqrt{1 - \zeta^2}}}
\]

i.e.

\[
M_p = e^{0.05} = 0.05
\]
\[ e^{\sqrt{1-\zeta^2}} = 0.05 \Rightarrow \zeta = 0.69 \]

Again settling time
\[ T_{setl} = \frac{4}{\zeta \omega_n} = 0.5 \]
i.e. \[ \frac{4}{\zeta \omega_n} = 0.5 \Rightarrow \omega_n = 11.59 \text{ rad/sec} \]

\[ \omega_d = \omega_n \sqrt{1 - \zeta^2} = 8.38 \]
\[ \omega_s = \frac{2\pi}{T} = \frac{2\pi}{0.01} = 628.31 \text{ rad/sec} \]
Clearly, sampling frequency is sufficiently large to avoid any aliasing.

Dominant pole location in s-domain
\[ s = -\zeta \omega_n + i \omega_n \sqrt{1 - \zeta^2} \]

Dominant closed loop pole location in Z-domain
\[ z = e^{\Gamma T} = 0.9199 \pm j0.0774 \]

Root locus of the current system:

![Root Locus Plot](image)

**Figure 1- Root locus for alpha/Vm open loop**

We have one pole outside of unit circle, and one zero on unit circle.
Since a root locus always starts from an open loop pole and ends at an open loop zero or infinity, the root locus will always lie between the above mentioned pole and zero, irrespective of the gain of the system. This rules out a P controller.

To bring the unstable pole inside the unit circle, we have to make multiple roots branch from the root locus which is lying outside the unit circle.

To accomplish this, we place one pole at (1,0). Now, we place two zeros inside the unit circle so that the root locus between them will branch in from the pole which is lying outside the unit circle.

Hence, we have selected a PID controller for controlling the system with only one feedback.

The basic controller form for this type of controller is:

\[
C_1(z) = K_p + \frac{K_v}{T_s} + \frac{K_i T_s}{(1-z^{-1})},
\]

which can be written as;

\[
C_1(z) = \left( K_p + \frac{K_v}{T_s} \right) \left( \frac{z^2 + z \left( \frac{K_i T_s - K_p - 2 K_v}{K_p + K_v T_s} \right) + \frac{K_v}{K_p + K_v T_s}}{z^2 - z} \right)
\]

Or

\[
C_1(z) = K \frac{(z - \alpha)(z - \beta)}{z(z - 1)}
\]

From above equation we know that we have one pole at zero and other at unit circle. We have to find out the angle deficiency, taking these two poles into account, at dominant closed loop pole to calculate the locations of the two zeros on the real axis.

Current zeros of the system
\[
Z =
1
-0.9277
\]

Current poles of the system
\[
P =
1.0763
0.9491
0.7814
\]

Locating them on real axis in z-plane and calculating angle deficiency at dominant closed loop pole

Graphical representation of location of pole zero location, to calculate angle deficiency
The pole and zero and unit circle are not considered in calculation as, angle contribution from zero will be cancelled by pole at the same location. 

Angle deficiency = $2.4 - (4.81 - 29.28 - 69.33 - 148.68) = -249.33$

Angle to be contributed by two zeros = $-(-249.33 + 180) \approx 69.7^\circ$

Here, we want to place the zeros close to each other and in such a way that there will be no pole in between them, because we want the root locus between the zeros to branch in at multiple roots. This is necessary to bring the root locus, which is outside the unit circle, inside of unit circle.

The tentative locations of zeros are calculated to be 0.8 and 0.9.

Now, open loop transfer function of our system becomes

$$G_{op} = K \frac{0.00155(z - 1)(z + 0.9277)}{(z - 1.0763)(z - 0.9491)(z - 0.7815)} \frac{(z - 0.8)(z - 0.9)}{z(z - 1)}$$

Equation 12 - Open Loop Transfer function of z control system

The gain can be calculated from the magnitude condition of root locus

$$K \frac{0.00155(z - 1)(z + 0.9277)}{(z - 1.0763)(z - 0.9491)(z - 0.7815)} \frac{(z - 0.8)(z - 0.9)}{z(z - 1)} \bigg|_{0.9199 + 0.0774i} = 1$$

However, from rlttool of MATLAB we get the gain to be 80.

Hence,

$$C_1(z) = \frac{V_{ma}(z)}{\alpha(z)} = 80 \frac{(z - 0.8)(z - 0.9)}{z(z - 1)}$$

Equation 13 - Difference equation of inner loop controller ($C_1(z)$)

The difference equation

Taking inverse z-transform of above equation

$$V_{ma}(k) = V_{ma}(k - 1) + 80[\alpha(k) - (0.8 + 0.9)\alpha(k - 1) + 0.8 \cdot 0.9\alpha(k - 2)]$$
At unit circle we have pole-zero cancellation. Otherwise, all the roots of the characteristic equation are within the constraint boundaries.
Plotting impulse response of the closed loop transfer function,

\[ G_{\alpha,vm,cl}(z) = \frac{G_{\alpha,vm,op}(z)}{1 + G_{\alpha,vm,op}(z)} \]

Figure 4 - Impulse Response of Closed loop system for Pendulum angle \( \alpha \)

Clearly, from above figure our settling time and % overshoot are within the range of the specified values.

It is worth noting that, we don’t have any control over \( \theta \) from this kind of controller. \( \theta \), the motor swing angle can go unbounded.
Green: Theta variations
Red: Voltage variations
White: alpha variations
It is clear from following simulation that theta variations are very large.

Figure 5-Large Theta variations from system, implementing only one feedback ($\alpha$)

In order to achieve the secondary task of controlling the motor angle $\theta$ and maintaining it at a specified reference value, we try to get additional feedback for $\theta$ as well.

**c. Digital Controller with two feedbacks, $\theta$ and $\alpha$**

Here, design constraint on setting time of theta and overshoot of theta are not stringent.

We will just follow, Theta $T_s >$ Alpha $T_s$

We are doing this because we do not want to saturate motor voltage. Our main objective is to keep pendulum in upright position.

The proposed block diagram

In the above block diagram we have inner loop which is controlling pendulum angle and the outer loop is controlling the motor swing angle.

The above block diagram can be reduced to
We notice here that $R_\alpha$ is always equal to zero, because pendulum has to maintain upright position. Hence, above block diagram can be reduced as

Here, $C_2(z) = C(z)/C_1(z)$. With above explanation it is clear that controller $C_2(z)$, will have negative gain.

The inner loop controller remains the same.

Here, first we need to find out transfer function for $\theta$ v/s $\alpha$ in discrete domain.

From, equation (8)

$$G_{\theta,\alpha}(s) = \frac{\theta(s)}{\alpha(s)} = \frac{39.1s^2 - 1642}{33.51s^2}$$

Transfer function of $G_{\theta,\alpha}(z)$, is taken as transfer function in continuous domain with a sampler and zero order hold.

$$G_{\theta,\alpha}(z) = (1 - z^{-1})\left(\frac{G_{\theta,\alpha}(s)}{s}\right), \text{ sampling time } = 0.01ms$$

$$G_{\theta,\alpha}(z) = \frac{1.167z^2 - 2.335z + 1.165}{(z - 1)^2}$$

Equation 14-$G_{\theta,\alpha}(z)$

Again we can reduce the above block diagram as,
Where,
\[ G_p(z) = \frac{G_{\alpha,v,m,op}(z)}{1 + G_{\alpha,v,m,op}(z)} \times G_{\theta,\alpha}(z) \]

At this point, it is very difficult to analyze the open loop transfer function with root locus analysis. We will go for bode diagram to see the type of controller required.

Bode plot of \( G_p(z) \)

![Bode Diagram]

**Figure 6-Bode Diagram of open loop system (Inner control loop in series with \( G_{\theta,\alpha}(z) \))**

From above bode plot, our system is unstable on its own because of negative phase margin. We can make our system stable with a lead compensator. A lead compensator increases systems bandwidth and improves phase margin.

Hence, the form of controller is PD
\[
C_2(z) = K_p + \frac{K_v}{T_s}(1 - z^{-1}) = \frac{K_p * T_s * z + K_v * z - K_v}{T_s * z}
\]

\[ \Rightarrow C_2(z) = \left( K_p * T_s + K_v \right) \times \frac{z - K_p * T_s + K_v}{K_v}
\]

\[ \Rightarrow C_2(z) = K \times \frac{(z - z_1)}{z} \]

Where, \( K=\frac{K_p * T_s + K_v}{T_s} \) and \( z_1 = \frac{K_v}{K_p * T_s + K_v} \)

The above information states that our controller has one pole at the origin and a real zero.
With the help of `rltool` command in MATLAB, we can place the location of the zero so as to bring all the poles of the open loop transfer function inside unit circle.

![Root locus plot](image)

**Figure 7-Root locus of final system**

Our controller figures out to be

$$C_2(z) = \frac{V_{m\theta}(z)}{\theta(z)} = (-15) \times \frac{(z - 0.94)}{z}$$

*Equation 15-C_2(z)*

The corresponding difference equation is found out using inverse z-transform

$$V_{m\theta}(k) = -15 \times [\theta(k) - 0.94 \times \theta(k - 1)]$$

*Equation 16-Difference equation for outer loop controller*
Bode plot of \( \frac{G_{a,Vm,op}(z)}{1+G_{a,Vm,op}(z)} \times G_{\beta,p}(z) \times C_2(z) \), is as follows

![Bode Diagram](image)

**Bode Diagram**

\( G_m = 12.5 \text{ dB (at 12.9 rad/sec)}, \ P_m = 79.5 \text{ deg (at 1.16 rad/sec)} \)

**Figure 8-Bode diagram of final system**

Clearly, we have achieved reasonable phase margin and gain margin to have system stable.
Figure 9-Impulse response of $\theta$

Clearly, the response takes little longer to settle down. This is beneficial and necessary since in a cascade control, we require the outer loop to have a larger settling time than the inner loop. As long as we are able to achieve the desired motor swing angle, our controller is satisfactory.
4. Simulation in Simulink
In order to validate the controller design, we try to simulate it using Simulink, a Graphics based simulation tool supported by MATLAB. We start with making a simulation model, proceed to the controllers and eventually lead up to the overall system.

a. Non-Linear Model of Actual System
The first task was to create a system to emulate the actual system in continuous domain. For this purpose, the non-linear equations for alpha and theta as a function of Voltage $V_m$ were modeled in Simulink as shown below.
**b. Linear Model at Unstable Position**

The next step was to create a linearized model for the above so that the performance of the linear model could be gauged. We use the equations for the model linearized about $\alpha=0$.

![Figure 11 - Linear Model - Simulink](image)

**c. Stabilization Controller**

Next, we try to get implement the stabilization controller for alpha and theta. We build it in the discrete domain using “discrete time transfer function” tool in Simulink. It appears below:
The responses we get are listed below.
d. Destabilization Controller

Now we try to build the destabilization controller that will amplify the oscillations of the pendulum from the hanging down position and bring around the unstable equilibrium position so that the stabilization controller can take over.

Our purpose here is to increase the amplitude of oscillations of pendulum from its downward equilibrium position until it reaches upright stability zone of +6 degrees. However, during the process we don’t want motor angle to drift too much from its set point.
To design destabilization controller we first need to linearise our dynamics equation about $\alpha=\pi$, instead of $\alpha=0$;

Complete kinetic energy of our system is taken as

$$T = \frac{1}{2} J_{eq} \dot{\theta}^2 + \frac{1}{2} m \left( r \dot{\theta} - L \cos(\alpha) \right)^2 + \frac{1}{2} m \left( -L \sin(\alpha) \right)^2 + \frac{1}{2} J_{eq} \dot{\alpha}^2$$

We can formulate Lagrangian as follows

$$L = T - V = \frac{1}{2} J_{eq} \dot{\theta}^2 + \frac{2}{3} m L^2 \dot{\alpha}^2 - mL \cos(\alpha) (\dot{\theta}) + \frac{1}{2} m r^2 \dot{\theta} - mgL \cos(\alpha)$$

By inspection of above two equations we can conclude that replacing $\alpha$ by “$\alpha+\pi$” the terms, which are in $\cos(\alpha)$ and $\sin(\alpha)$ will get their sign flipped. This leads to a total sign reversal of all the terms in $L$.

Hence, the new state equation will become as follows

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \dot{\theta}^- \\ \dot{\alpha}^- \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & K1 & K2 & 0 \\ 0 & -K3 & -K4 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \\ \theta^- \\ \alpha^- \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K5 \\ -K6 \end{bmatrix} V_m$$

Please note, $K_1$-$K_6$ are same as described earlier. The signs of $K_3$, $K_4$ and $K_6$ are flipped.

The new transfer functions

$$\frac{\alpha(z)}{V_m(z)} = \frac{-0.001555 z^2 + 0.0001124 z + 0.001443}{z^3 - 2.79 z^2 + 2.589 z - 0.7984} \text{ Equation 17 } \frac{\alpha(z)}{V_m(z)} \text{ for reference of } \alpha = \pi$$

Our objective is to place dominant closed loop poles just outside the unit circle, so that the response will slowly increase in amplitude. (Placing dominant closed loop pole far outside the unit circle will cause controller to saturate easily )

We used rltool of MALAB to accomplish it,

The final root locus for inner loop destabilizing controller (i.e destabilizing controller for pendulum angle)

Our destabilization controller for $\alpha$

$$C_1(z) = \frac{5.7 z - 3}{z^2} \text{ Equation 18-Destabilization } \alpha \text{ controller}$$
The response of destabilizing controller for alpha

Similarly, we can design destabilization controller for

\[ G_p(z) = \frac{G_{\alpha,vm,op}(z)}{1 + G_{\alpha,vm,op}(z)} \times G_{\theta,\alpha}(z) \]

Where new \( G_{\theta,\alpha}(z) \) is computed as follows (With MATLAB, code is attached)

\[ G_{\theta,\alpha}(z) = \frac{-1.167 z^2 + 2.331 z - 1.169}{z^2 - 2z + 1} \]

Equation 19 \( G_{\theta,\alpha}(z) \) for reference of \( \alpha = \pi \)
Outer loop controller is as follows

\[ C_2(z) = \frac{4}{z} \]

Equation 20 - Destabilization outer loop controller

Impulse response of destabilization controller for outer loop

Figure 17 - Impulse Response of θ for destabilization controller
e. Mode Controller

The next step is the design of a mode controller that will switch between the Stabilization and Destabilization controllers depending on the deviation of the pendulum from the upright position. The only aspect here was the value of alpha threshold at which the controllers would be switched. To ascertain that, we build a circuit for comparing the Linear and Non-linear models for the system. It is as shown below:

The response we get indicates that the two show a considerable divergence at $\alpha=\pm14^\circ$. 
Hence, the alpha threshold is set at 0.25 radians. The resulting controller and sample run output is shown below.
f. **Overall Model and Performance**

Finally, we simply put all the models as submodels in a main window and switch the stabilization and de-stabilization controllers using a switch controlled by the Mode Controller.
5. LabView Models and Physical Implementation

As the final phase of the project, we build the designed controllers in LabView. This software is used to interact with the actual system and control it.

a. Controller for Alpha only

The first trial is done by implementing only the alpha controller. It is as shown below. The simulation is not proper but has been retained for showing the overall flow of the VI. The saturation controller controls the amount of voltage being fed to the motor to avoid it from burning out.

![Figure 24 – Block Diagram - Alpha Controller only – LabView](image-url)
Red — Voltage variations
White — α Variations

We can see from above figure that the α variations are almost zero. In real world scenario we have little variations in surrounding conditions, and we did not take friction in account in deriving the dynamics equations for the system, that is why we have variations in voltages. And also the pendulum angle is not absolutely zero always.
b. Controller for both Alpha and Theta (destabilization included)

Now we aim to make the overall controller for alpha and theta combined. We have also included the Destabilization and Mode controllers.

![Block Diagram](image-url)
The Front Panel is as shown below. It is in the stabilization mode. In addition to the earlier white and red lines for the Alpha error and Control Voltage respectively, we have green line for $\theta$ error.

![Front Panel - Stabilization of System – Labview](image)

It is worth noting that we have to change the gain of outer loop controller from 15 (the MATLAB simulation) to 6, which is quite deviant from the expected/designed gain of -15. This could be attributed to friction in the bearing and probably backlash in the motor.
As can be seen, the controller starts to amplify the alpha error from hanging down position and as soon as alpha crosses zero, the stabilization controller kicks in and the alpha is stabilized. The theta error also starts decreasing and then eventually drops to zero.

6. Conclusion
In conclusion, we were successfully able to design controllers that performed the desired tasks of stabilizing/destabilizing the Rotary Inverted Pendulum System. During this process, we learnt how a physical system can be analyzed as a simplistic model and how the digital control techniques, which we learnt during the course, could be used to achieve the desired outputs. We also learnt that the physical system is not always a perfect match to the simplistic model we suppose it to be. We have to make some adjustments in the designs so as to accommodate factors like friction and backlash.

We could use PD controller to improve systems’ margin. The approximation to continuous 2\textsuperscript{nd} order system gives reasonably good results for dominant closed loop pole location. The tradeoff between settling time of pendulum and motor swing angle is very important to avoid saturation in voltage.
7. References

Books

Journal Articles
5. SWING-UP AND STABILIZATION OF ROTARY INVERTED PENDULUM, Mertl, Jaroslav Sobota, Miloš Schlegel, Pavel Balda,

8. Appendix

a. Matlab Code for stabilization controller

```matlab
function final_controller()

Beq=5.0E-3;
ng=0.9; nm=0.69; Jeq=3.87E-7; Kg=70;
Km=7.67E-3; Kt=7.67E-7; Rm=2.6;
r=0.2; L=0.35/2;
m=0.128;
g=9.81;
a=Je+ m*r^2;
b=m*L*r;
c=4/3*m*L^2;
d=m*g*L;
E=a*c-(b^2);
G=((nm*ng*Kt*Km*Kg^2) + (Beq*Rm))/Rm;

A=[ 0 0 1 0; 0 0 0 1; 0 b*d/E -c*G/E 0; 0 a*d/E -b*G/E 0];
B=[0;0;c*nm*ng*Kt*Kg/Rm/E;b*nm*ng*Kt*Kg/Rm/E];

T=.01;
k1=A(3,2);
k2=A(3,3);
k3=A(4,2);
k4=A(4,3);
k5=B(3,1);
k6=B(4,1);
```


sysc=tf([k6 0],[1 -k2 -k3 k3*k2-k1*k4]);
T=0.01;
disp('alpha/Vm transfer function');
syisd=c2d(sysc,T,'zoh');
[Z,P,K] = zpkdata(syisd,'v');
rltool(syisd);
z1=0.8; z2=0.9; k1=80; p1=1; p2=0;
sys_Gc=tf(k1*[1 -(z1+z2) z1*z2],[1 -(p1+p2) p1*p2],T);
sys_ser=series(syisd,sy_Gc);
rltool(sys_ser);
sys_close=feedback(syisd,sy_Gc);
figure(1);
impulse(sys_close,0:T:1.5);
title('Pendulum angle(\alpha) Impulse Response')

disp('Theta/alpha transfer function')
thalc=tf([k5 0 k6*k1-k3*k5],[k6 k4*k5-k2*k6 0]);
thald=c2d(thalc,T)
sys_ser1=series(sys_close,thald);
[Z P K]=zpkdata(sys_ser1,'v')
% keyboard
P(1)=1; P(2)=P(1);
P(3)=1;% Manually Putting actual values of P(1),P(2),P(3)
sys_ser1=zpk(Z,P,K,T);
[Z P K]=zpkdata(sys_ser1,'v')
% rltool(sys_ser1);
figure
margin(sys_ser1)
z1=.94; k1=-15;
sys_Gc1=tf(k1*[1 -z1],[1 0],T)
sys_ser2=series(sys_ser1,sy_Gc1);
figure
margin(sys_ser2)
rltool(sys_ser2);
sys_close1=feedback(sys_ser2,1);
figure;
impulse(sys_close1,0:T:2);
title('Motor angle(\theta) Impulse Response')

b. Matlab Code for Destabilization Controller

function final_destab()
Beq=5.0E-3;
ng=0.9; nm=0.69; Jeq=3.0E-3;Jm=3.87E-7; Kg=70;
Km=7.67E-3; Kt=7.67E-3;Rm=2.6;
r=0.2; L=0.35/2;
m=0.128;
g=9.81;
\[
\begin{align*}
a &= J_{eq} + m r^2; \\
b &= m L r; \\
c &= \frac{4}{3} m L^2; \\
d &= m g L; \\
E &= a c - (b^2); \\
G &= \frac{(n m * n g * K_t * K_m * K_g^2) + (B_{eq} * R_m)}{R_m}; \\
\end{align*}
\]

\[
A = \begin{bmatrix} 0 & 0 & 1 & 0; \\
0 & 0 & 0 & 1; \\
0 & b * d / E & - c * G / E & 0; \\
0 & a * d / E & - b * G / E & 0; \\
\end{bmatrix}; \\
B = [0; 0; c * n m * n g * K_t * K_g / R_m / E; b * n m * n g * K_t * K_g / R_m / E]; \\
T = 0.01; \\
k1 = A(3, 2); \\
k2 = A(3, 3); \\
k3 = - A(4, 2); \\
k4 = - A(4, 3); \\
k5 = B(3, 1); \\
k6 = - B(4, 1); \\
\]

\[
sy = tf([k6 0], [1 - k2 - k3 k3 * k2 - k1 * k4]); \\
T = 0.01; \\
disp('Alpha/Vm Discrete model'); \\
sysd = c2d(sy, T, 'zoh'); \\
sys_Gc = tf([5.7 -3], [1 0 0], T) \\
sys_ser = minreal(series(syd, sys_Gc)); \\
% rltool(sys_ser) \\
sys_close = feedback(syd, sys_Gc); \\
figure(1); \\
impulse(sys_close, 0:T:5); \\
title('Pendulum angle(\alpha) Impulse Response'); \\
disp('\Theta/\alpha transfer function'); \\
thalc = tf([k5 0 k6 * k1 - k3 * k5], [k6 k4 * k5 - k2 * k6 0]); \\
thald = c2d(thalc, T) \\
[Z PK] = zpkdata(thald, 'v') \\
sys_ser1 = minreal(series(sys_close, thald)); \\
[Z PK] = zpkdata(sys_ser1, 'v') \\
sys_Gc1 = tf([0 4], [1 0], T) \\
sys_ser2 = series(sys_ser1, sys_Gc1); \\
rltool(sys_ser2) \\
sys_close1 = feedback(sys_ser2, 1); \\
figure; \\
impulse(sys_close1, 0:T:3) \\
title('Motor angle(\theta) Impulse Response')
c. LabView Codes for Alpha Controller Only

I. Alpha Controller

if(s==0)
    vk=0;
else
    vk=vk1+k*(ek-(alpha+beta)*ek1+alpha*beta*ek2);

II. Saturation Controller

if(v>4.0 || v<-4.0)
    vs=0
else
    vs=v;

d. LabView Codes for Cascade Controller

I. Alpha Stabilization Controller

if(mode==1)
    {vk1=0;vk=0;}
else
    vk=vk1+k*(ek-(alpha+beta)*ek1+alpha*beta*ek2);
if(On==0)
    vk=0;

II. Theta Stabilization Controller

if(mode==1)
    {vk1=0;}
vk=k*(ek-alpha*ek1);
if(On==0)
    vk=0;

III. Alpha,(theta destabilization is included) Destabilization Controller

if(mode==0)
    vk=0;
else
    vk=2.7*(ealk-pi)-3.0*(ealk1-ealk)+4.0*ethk-4.0*(ethk-ethk1);
Please note that we have set point for pendulum angle as pi in this case.

IV. Mode Controller

if(ealk<al_thres&&ealk>(-al_thres))
    {vk=vks;
     mode=0;}
else
    {vk=vku;
     mode=1;}
if(On==0)
    vk=0;
V. Saturation Controller

\[
\begin{align*}
\text{if}(v>5.5 \text{||} v<-5.5) \\
vs=0; \\
\text{else} \\
vs=v;
\end{align*}
\]

VI. Modulus Operator

\[
\begin{align*}
\text{if}(a<-\pi) \\
\{\text{while}(a<-\pi) \\
a=a+2\pi;\}
\text{if}(a>\pi) \\
\{\text{while}(a>\pi) \\
a=a-2\pi;\}
\end{align*}
\]
\[a_1=a;\]