CHAPTER 2:

PROBLEM SETUP

Briefly, the inverted pendulum is constituted by a trolley and pendulum. The control objective is to move the carriage to the commanded position without causing the pendulum to fall over. Open-loop system is purely unstable. This configuration can be used to study the control of the open loop system unstable. It is demonstration of the benefits of stabilizing feedback control. A range of control techniques ranging from simple phase-lead compensator to neural network-controller can be applied.

2.3 Range of Possible Solutions

An inverted pendulum is a classic example of control. The process is non-linear and unstable with a different input and output. The objective is to balance the system vertically on the trolley. There are several approaches to solve the system in terms of equations of motion. Following is brief description of some approaches to understand and solve the system.

Method 1:

Observer is standing at the position of the carriage is to say, inside the carriage and observing the motion of the pendulum not of the trolley. This is the simplest approach possible.

Method 2:

Observer is standing outside the system and collectively observing the motion and equilibrium state of the system. A much more complex method to understand and streamline the system. He needs a deep understanding of non-linear physics.

Method 3:

Observer is standing outside the system and observing the system as free-body diagrams involved in the inverted pendulum system. In our case, there are a total of two free-body diagrams that cart and pendulum.

A classical and best approach to solve this system because of free body diagrams.

$$\frac{d}{dt}\underline{z} = \frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \frac{d}{dt}\begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} \rightarrow eq(13)$$

We have equation (12),

$$\ddot{\theta} = \frac{\mu \cos\theta - (M+m)g\sin\theta + ml(\cos\theta\sin\theta)\dot{\theta}}{ml\cos^2\theta - (M+m)l}$$

Substituting the assumptions in equation

$$\Rightarrow \ddot{\theta} = \frac{\mu\cos z_1 - (M+m)g\sin z_1 + ml(\cos z_1 \sin z_1)z_2^{\;2}}{ml\cos^2 z_1 - (M+m)l}$$

Again, substituting these assumptions in equation (11) and simplifying,

$$\Rightarrow \ddot{x} = \frac{\mu + ml(\sin z_1)z_2^{\;2} - mg\cos z_1 \sin z_1}{M + m - m\cos^2 z_1}$$

“Simply playing the derivatives”

$$\frac{d}{dt}\begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} z^2 \\ \dfrac{\mu\cos z_1 - (M+m)g\sin z_1 + ml(\cos z_1\sin z_1)z_2^{\;2}}{ml\cos^2 z_1 - (M+m)l} \\ z_4 \\ \dfrac{\mu + ml(\sin z_1)z_2^{\;2} - mg\cos z_1\sin z_1}{M + m - m\cos^2 z_1} \end{bmatrix} \rightarrow eq(14)$$

This expression is now in the desired form given in equation. (7.58). If both the pendulum angle and the cart position x are of interest, we have

$$\left.\frac{Jf_1}{Jz_1}\right|_{(z_\circ,u_\circ)} = \frac{J}{Jz_1} z_2 = 0$$

Using quotient rule's result and simplifying,

$$\left.\frac{Jf_2}{Jz_1}\right|_{z_\circ,u_\circ} = \frac{(M+m)g}{Ml}$$

And

$$\left.\frac{Jf_3}{Jz_1}\right|_{z_\circ,u_\circ} = \frac{J}{Jz} z_4 = 0$$

Since

$$\frac{Jf_4}{Jz_1} = \frac{J}{Jz}\left(\frac{u + ml(\sin z_1)z_2{}^2 - mg\cos z_1 \sin z_1}{M + m - m\cos^2 z_1}\right)$$

Simplifying these expressions and re-arranging results into following

$$\left.\frac{Jf_4}{Jz_1}\right|_{z_\circ,u_\circ} = \frac{-mg}{M+m-m} \Rightarrow \frac{-mg}{M}$$

$$\left.\frac{Jf_1}{Jz_2}\right|_{(z_\circ,u_\circ)} = \frac{J}{Jz} z_2 = 1$$

$$\left.\frac{Jf_3}{Jz_4}\right|_{\underline{z}_\circ,u_\circ} = J/Jz\,(z_4) = 1$$

Thus, combining all these separate terms gives System

$$\underline{\underline{J}}_z(z_\circ,u_\circ) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(M+m)g}{Ml} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{M} & 0 & 0 & 0 \end{bmatrix} \qquad eq(17)$$

Now derivative of the non linear terms with respect to μ

$$\underline{J}\left(\underline{z}_{\circ}, u_{\circ}\right) = \begin{bmatrix} \frac{Jf_1}{Ju} \\ \frac{Jf_2}{Ju} \\ \frac{Jf_3}{Ju} \\ \frac{Jf_4}{Ju} \end{bmatrix}$$

Now determining the elements of Jacobian matrices μ

$$\left.\frac{Jf_1}{Ju}\right|_{(z_{\circ}, u_{\circ})} = {}^{J}\!/_{Ju}(z_2) = 1$$

$$\frac{Jf_2}{Ju} = {}^{J}\!/_{Ju}\left(\frac{u\cos z_1 - (M+m)g\sin z_1 + ml(\cos z_1.\sin z_1)z_2^2}{ml\cos^2 z_1 - (M+m)l}\right) = \frac{-1}{Ml}$$

$$\left.\frac{Jf_4}{Ju}\right|_{\underline{z}_{\circ}, u_{\circ}} = \frac{J}{Ju}\left(\frac{u + ml(\sin z_1)z_2^2 - mg\cos z_1\sin z_1}{M + m - m\cos^2 z_1}\right) = \frac{1}{M}$$

Re-arranging all the derivative terms w.r.t input force, we have

$$\underline{J}\left(\underline{z}_{\circ}, u_{\circ}\right) = \begin{bmatrix} 0 \\ \frac{-1}{Ml} \\ 0 \\ \frac{1}{M} \end{bmatrix} \qquad eq(18)$$

Finally, after all these manipulations we can write eqn. (16) explicitly as

$$\frac{d}{dt}\delta \underline{z} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(M+m)g}{Ml} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{M} & 0 & 0 & 0 \end{bmatrix} \delta \underline{z} + \begin{bmatrix} 0 \\ \frac{-1}{Ml} \\ 0 \\ \frac{1}{M} \end{bmatrix} \qquad eq(19)$$

What is the form of LTI state standards required for the implementation in Matlab. Equation (19) together with the definition given in response eqn. (15) represents the final linear model of inverted pendulum system.

4.1 Pole Zero Map of Open Loop System

The poles position of the linearized model of the inverted pendulum (in open-loop configuration) indicates that the system is unstable, as one of the poles of the transfer function is located on the right half of the s-plane. Thus the system is absolutely unstable.

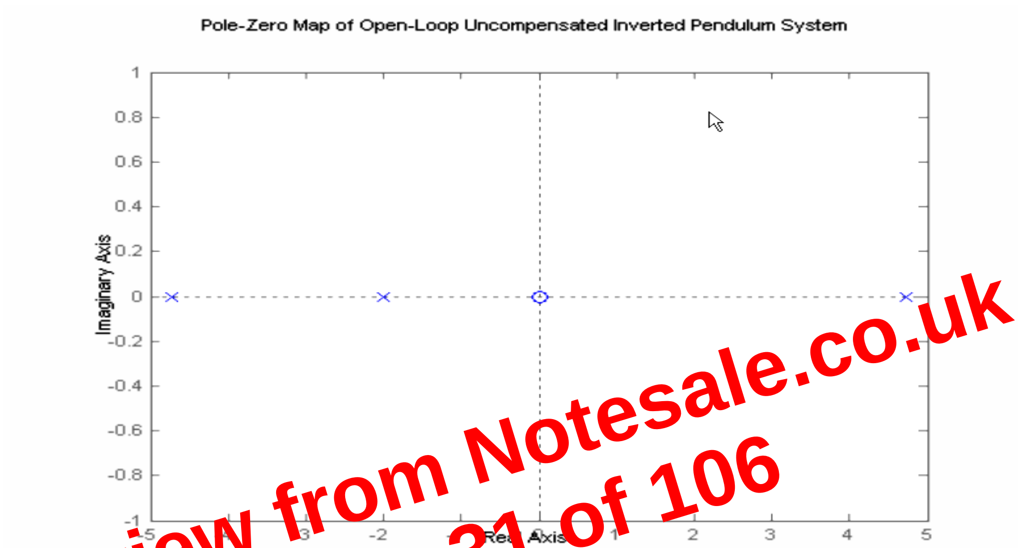

Figure 4: Pole Zero Map of Open Loop System

4.2 Impulse Response of Open Loop System

An impulse response of the system is shown in following figure. The system is highly unstable as theta diverges very rapidly. The runaway nature of the response indicates instability.

4.4 Root Locus of Open Loop System

The first step in the design of compensation for each plant is to observe the response of the unit closed feedback loop to control the stability. Many systems are unstable in open loop stable but in closed loop configuration. Vice versa is also possible that the system is stable in open loop but unstable in closed loop, although this case is rare. The closed-loop system can be studied viewing uncompensated root locus of the system graphic Ofthe. The following figure shows a plot the root locus of the system.

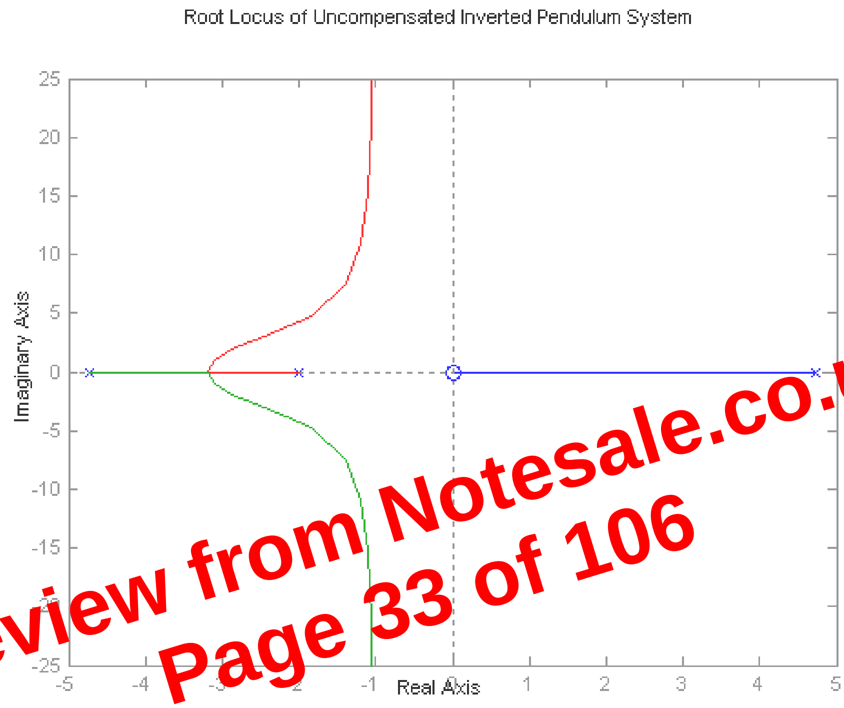

Figure 7: Root Locus of Open Loop System

The plot reveals that the system cannot be controlled by a simple loop unity feedback. Whatever the value of loop gain, ("K), a branch of the locus remains RHS (in the unstable region) of the s-plane. This makes it impossible to control by unity feedback.

The root locus has a branch on the right hand side of the imaginary axis, which indicates that the system is unstable in closed loop for all values of K.

From the analysis above, it is concluded that using only the closed loop gain compensation cannot check IP. REMODELING SYSTEM of the root locus is necessary, so that, for some set of gains, the system has all its roots in the left-half plane (stable region) s plane.

For the loop stability it is required that the loop gain should be greater than the given value (i.e., $K > K_{MIN} = 0.8$).

4. Now we set the gain such that roots of the characteristics equation correspond to z = 0.75 and $T_{S\,MAX} = 0.5$, so we select roots (selected by cross) as shown in figure. These roots correspond to value of k = 8.

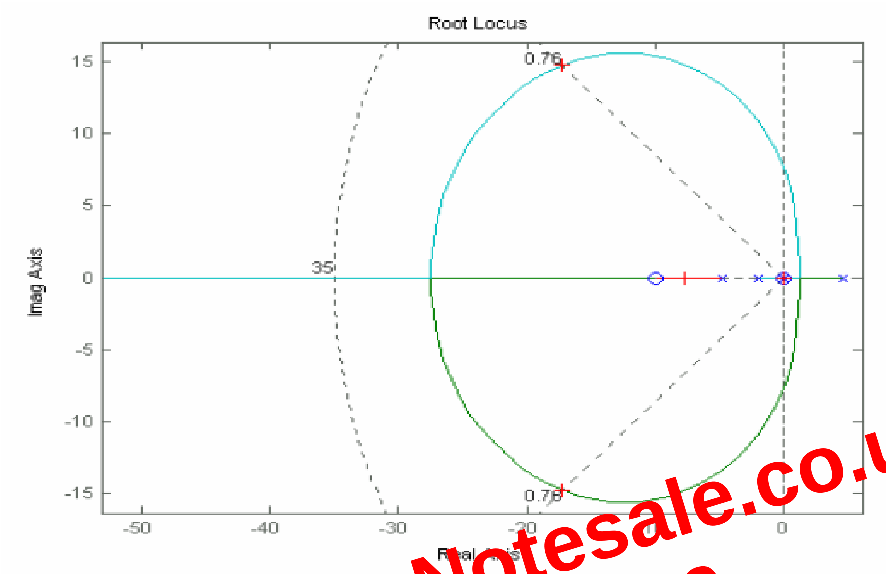

Figure 10: Root Locus after Settling Gain

The compensator equation is found to be:

$$G_c = 8 * \frac{(s^2 + 20s + 100)}{s}$$

And the general form is:

$$G_c = K_c * \frac{(K_D s^2 + K_p s + K_I)}{s}$$

Comparison gives us

$$K_C = 8; \quad K_P = 20 \quad K_I = 100 \quad K_D = 1$$

However, after simulating this PID compensated system in SIMULINK, the gain K_C was found to be best when it is 30 (minimum overshoot). So the final values of the gains are

$$K_C = 30; \quad K_P = 20 \quad K_I = 100 \quad K_D = 1$$

DESIGN 1: IDEAL PID ALGORITHM

This ideal PID algorithm has similarities with general operational amplifier. Next is a schematic representation of an operational amplifier in which input impedance is Z1 and the impedance Z2 is feedback or parallel.

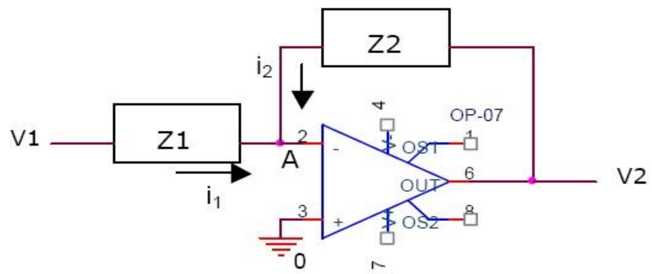

Figure 17: Ideal PID Controller Design

Overall transfer function of this amplifier is

$$V_2 / V_1 = -\{(R_f / R_i + C_i / C_f) + (1 / R_i C_f D) + (R_f C_i) * D\} = -\{K_p + K_i / D + K_d D\}$$ Appendix

Through this equation we can have

PROPORTIONAL GAIN is $K_p = (R_f / R_i + C_i / C_f)$,
INTEGRAL GAIN is $K_i = (1 / R_i C_f)$, and
DERIVATIVE GAIN is $K_d = R_f C_i$.

DESIGN 2: SERIES PID ALGORITHM

In this series of transfer function of PID controller PID controller is obtained as a product of PI and PD. Then follows this configuration is PI and PD cascade to get PID.

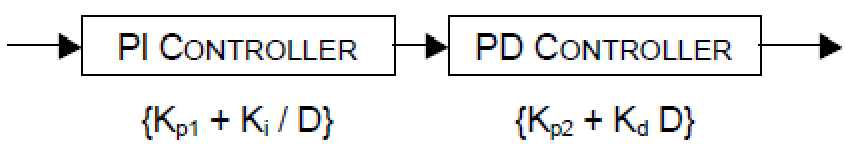

Figure 18: Block Diagram of Series PID Controller

CHAPTER 8:

FABRICATION OF SYSTEM

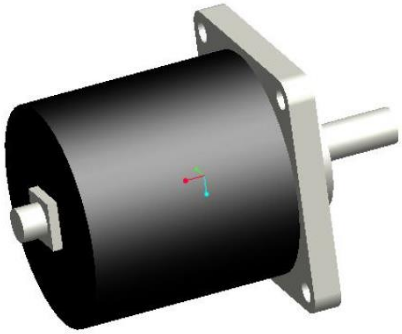

Figure 27: Angle Encoder

Transformer

A step down transformer is required to step down the input 220 AC voltages to the required voltage level.

LCD

LCD was needed for the interfacing purposes to display the angle of the pendulum read through the angle encoder.

8.5 Final Physical System Configuration

Finally, we have come across through the following design.

Figure 28: Final Physical System Model

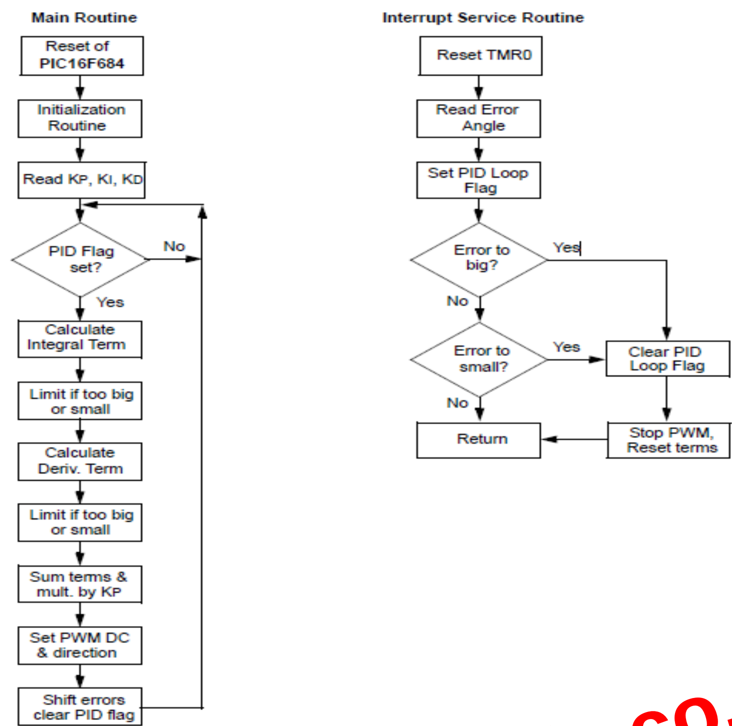

Figure 33: C CODE FLOW CHART

Here is schematic of the modern microcontroller (PIC16F684) used in the our circuit describing the pin level behavior.

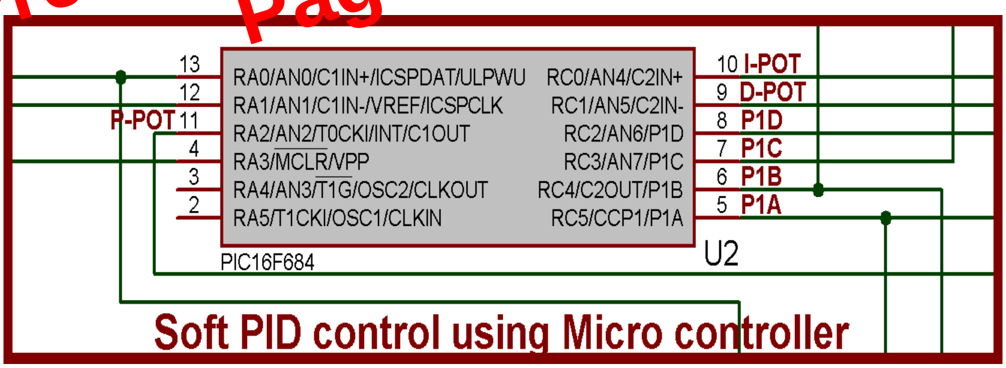

Figure 34: Pin Description of 16f684 Microcontroller

This figure demonstrates the PCB layout at rear side view of MCU.

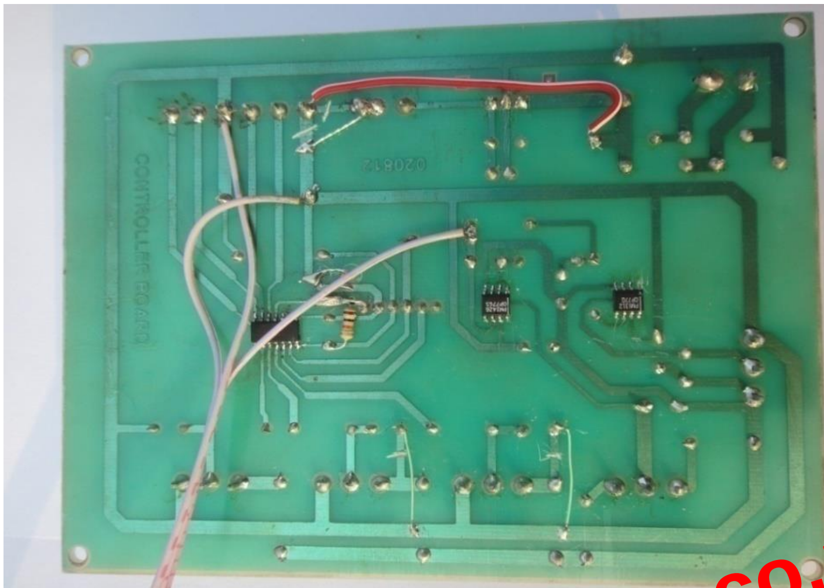

Figure 38: PCB Layout of Controller Board (1)

Here is component level view of the Microcontroller board,

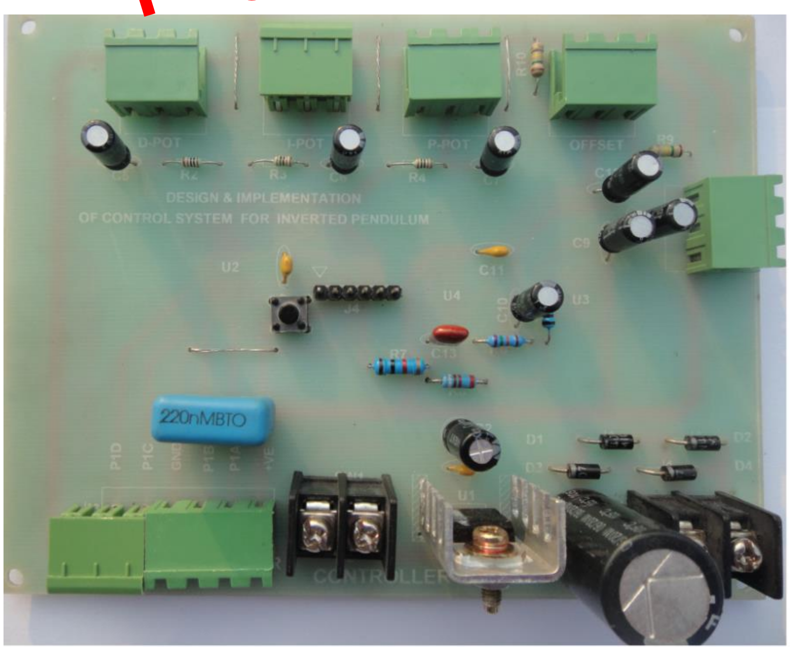

Figure 39: PCB Layout of Controller Board (2)

Therefore, overall transfer function of system will be

$$Gc(s) = \frac{V_2}{V_1} = K_P + \frac{K_I}{s} + K_D.s$$

This equation has two pole(s) and one zero(s).

(2).M-FILE FOR OPEN LOOP &CLOSED LOOP (UNCOMPENSATED) TRANSFER FUNCTION OF IP SYSTEM

```
%**********************************************
% DESIGN AND IMPLEMENTAION OF INVERTED PENDULUM
% Transfer function of the open loop & closed loop
% (Uncompensated) Inverted Pendulum System
% SST UMT, LAHORE, PAKISTAN
%**********************************************
% variables I, Kf, Km, Lp, M, b, g, l, m, r, tau
%
load ip_data

Kp = 1 / ((M + m) * g);
K  = Kf * Kp * Km * r * (M + m);
Ap = sqrt ((M + m) * m * g * l / ((M + m)*(I + (m * (l ^ 2)))- ((m * l)^2)));

% G1(s) = Theta(s) / U(s)
% ø represents a small angle from the vertical upward direction,
% u represents the input (impulse) force on the cart by pulley chain mechanism.
num_Th_U = [0 0 Kp];
den_Th_U = [Ap^(-2) 0 -1];
Th_U = tf (num_Th_U, den_Th_U);

% G2(s) = U(s) / E(s)
% u represents the input force on the cart by the pulley chain mechanism.
% e represents the input to the motor driving pulley-chain mechanism.
num_U_E = [((Km * (M + m))*r) 0];
den_U_E = [tau 1];
U_E = tf (num_U_E, den_U_E);

disp ' '
% G(s) = Theta(s) / E(s)
% Forward Transfer Function (Open Loop Without Feedback)
num_G = conv (num_U_E, num_Th_U);
den_G = conv (den_U_E, den_Th_U);
disp 'Forward Path Transfer Function of Inverted Pendulum System is:'
G = series (U_E, Th_U)

% H(s) (Feedback)
num_H = Kf;
den_H = 1;
H = tf (num_H, den_H);

% Closed Loop Transfer Function of Uncompensated System
% Gc(s) = G(s) / (1 + G(s) * H(s))
disp 'Closed Loop Transfer Function of Inverted Pendulum System is:'
Gc = feedback (G, H)

% GH(s)
% Open Loop Transfer Function
GH = series (G, H);
```

```
        ECCPAS = 0;                             // Auto_shutdown is disabled for now
        PR2 = 0x3F;                             // Sets PWM Period at 31.2 kHz
        T2CON = 0;                              // TMR2 Off with no prescale
        CCPR1L = 0;                             // Sets Duty Cycle to zero
        TMR2ON = 1;                     // Start Timer2

        ANSEL = 0b00110101;             // Configure AN0,AN2,AN4 and AN5 as analog
        VCFG = 0;                       // Use Vdd as Ref
        ADFM = 1;                       // Right justified A/D result
        ADCS0 = 1;                      // 16 TOSC prescale
        ADCS1 = 0;
        ADCS2 = 1;
        CHS0 = 0;                       // Channel select AN0
        CHS1 = 0;
        CHS2 = 0;
        ADON = 1;                       //Turn A/D on

        en0 = en1 = en2 = en3 = term1_char = term2_char =0;
        ki = kd = 0;
        kp = off_set = 0;
        temp_int = integral_term = derivative_term = un =0;
        SumE_Max = 30000;
        SumE_Min = 1 - SumE_Max;
        do_PID = 1;                     // Allowed to do PID function
        T0CS = 0;                       // Timer0 as timer not a counter
        TMR0 = 10;                      // Preload value
        PSA = 0;                        // Prescaler to Timer0
        PS0 = 0;                        // Prescale to 32 => 256 Hz
        PS1 = 0;
        PS2 = 1;
        INTCON = 0;
        PIE1 = 0;
        T0IE = 1;                       // Enable Timer0 int
        GIE = 1;
        return;
}


void PID()                              // The from of the PID is C(n) = K(E(n) + (Ts/Ti)SumE + (Td/Ts)[E(n) - E(n-1)])

{
        integral_term = derivative_term = 0;

// Calculate the integral term
        SumE = SumE + en0;                      // SumE is the summation of the error terms
        if(SumE > SumE_Max){                    // Test if the summation is too big
                SumE = SumE_Max;
        }
        if(SumE < SumE_Min){                    // Test if the summation is too small
                SumE = SumE_Min;
        }                                       // Integral term is (Ts/Ti)*SumE where Ti is Kp/Ki
                                                // and Ts is the sampling period
                                        // Actual equation used to calculate the integral term is
                                        // Ki*SumE/(Kp*Fs*X) where X is an unknown scaling factor
                                        // and Fs is the sampling frequency
        integral_term = SumE / 256;             // Divide by the sampling frequency
        integral_term = integral_term * ki;     // Multiply Ki
        integral_term = integral_term / 16;     // combination of scaling factor and Kp
```