Embedded Control Laboratory

Ball on inclined plane

MECHATRONICS MASTERS

Report of lab exercise 2

WS 2019/20

Group 08

1522099 Kurapati Chakradhar Reddy

1536256 K Harish

1536227 Surya Vara Prasad Alla

08.12.2019

Table of Contents

1. Introduction ……………………………………………………3
2. Objective ……………………………………………….....3
3. Model of the Closed loop system………………….....………..3
4. Tasks……………………..………………………..……..……10

1. Introduction

The main aim of the experiment is to balance the ball on the inclined plane by the Servo motor which acts as a actuator to change the angle of inclined plane with respect to the horizontal axis. The servo motor which controls the angle of the inclind plane is controlled by the Programmable Micro-controller.

2. Objective

Modelling and simulation of Ball-on-inclined-Plane (BoiP) system with closed loop control using Xcos.

Xcos:

Xcos is a Scilab tool used for mathematical modelling and simulation of dynamic and hybrid systems.

3. Model of Closed loop system

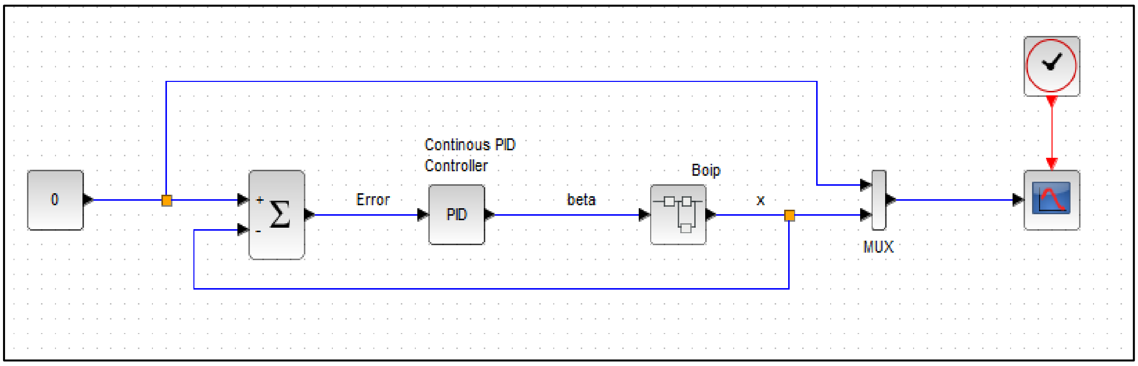


Figure 3.1 : Closed loop system of Continous PID Controller

The closed loop control system consists of the blocks,

- Reference Value – the user defined constant value which the system under control should reach. In this task the Reference value should lie between -1 and 1.
- Error –it is the difference between Reference value and measured process variable.
- PID controller – a proportional–integral–derivative controller (PID controller) is a control mechanism. A PID controller calculates an error

value as the difference between reference value and a measured process variable and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively).

- Control Signal – output signal of the PID controller which is sent to the BoiP system to control it.
- Controlled system – It is a system which is controlled. In our case it is the BoiP.
- Controlled signal – It is the output of controlled system. In our case it is the position ('x') of the ball on the inclined plane.

BoiP Sub-system model:

Boip sub-system consists of the model,

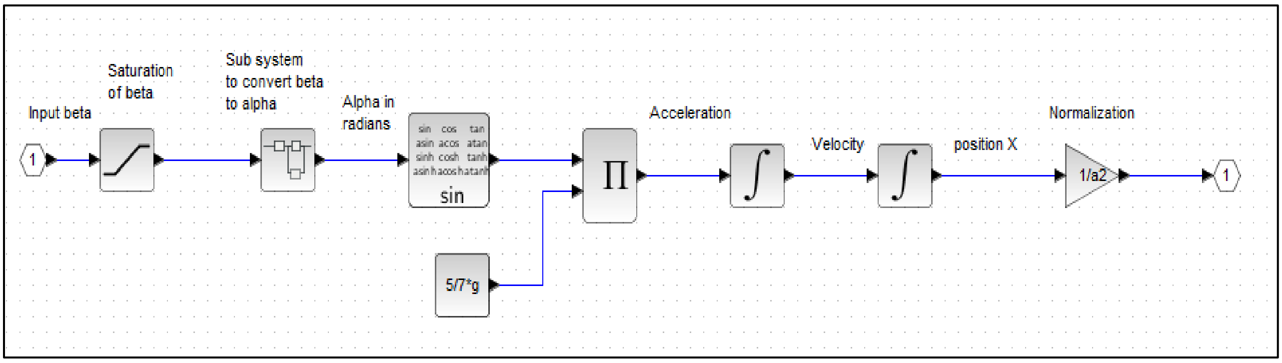
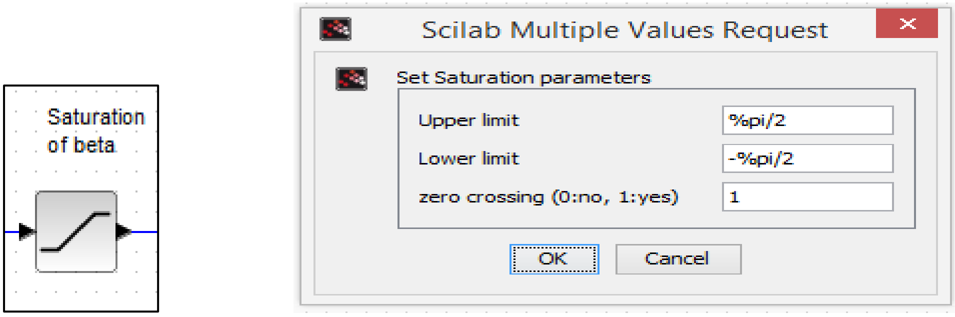


Figure 3.2 : BoiP sub-system of Continous PID Controller

Components of BoiP sub-system:

1. Saturation of Beta,



Saturation of beta block consists of the parameters to set the saturation values(upper limit, lower limit).

2. The Sub-system to convert beta to alpha(super block), consists of the equation developed to balance the ball on the inclined plane.

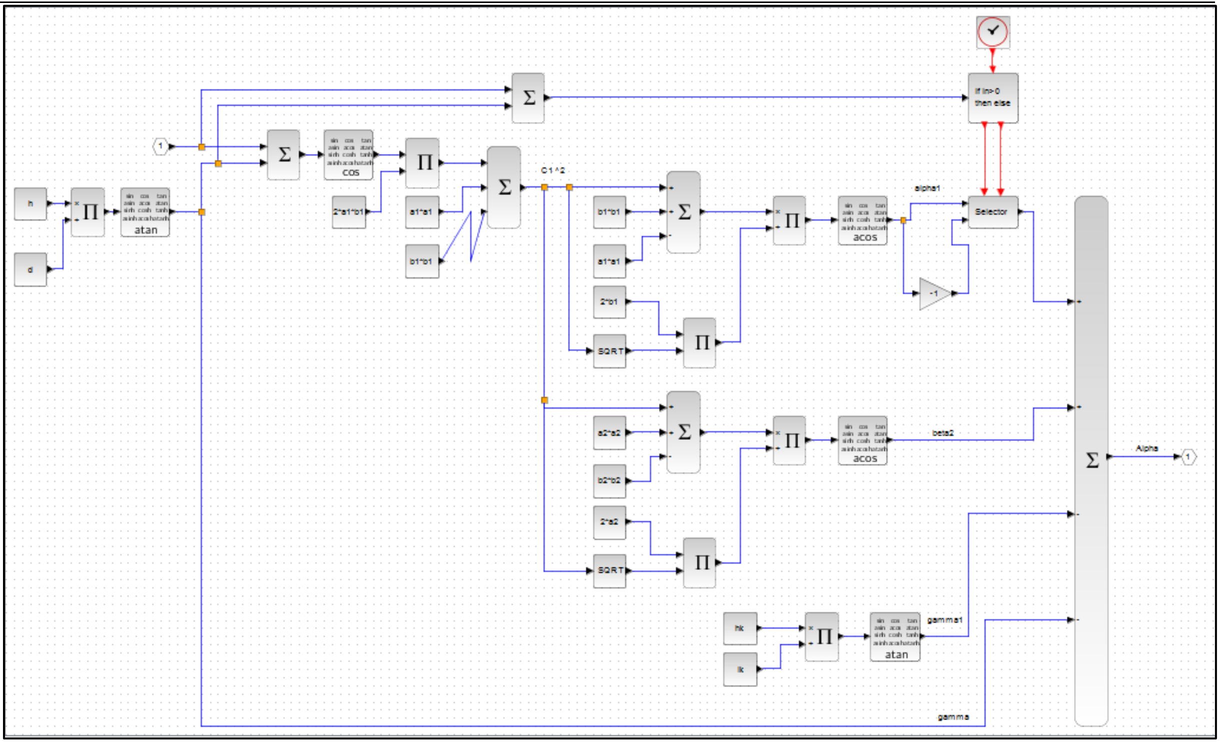


Figure 3.3 : Sub-system to convert beta to alpha.

3. Blocks to get position $\boldsymbol{x}$ from angle $\boldsymbol{\alpha}$:

From the analysis of the system, the acceleration of the ball is given by

$$a = 5/7*9.81*\sin(\alpha)$$

Single time derivation gives velocity

$$v=\int a*dt$$

and second time derivation gives position,

$$x= \int v*\mathrm{d}t$$

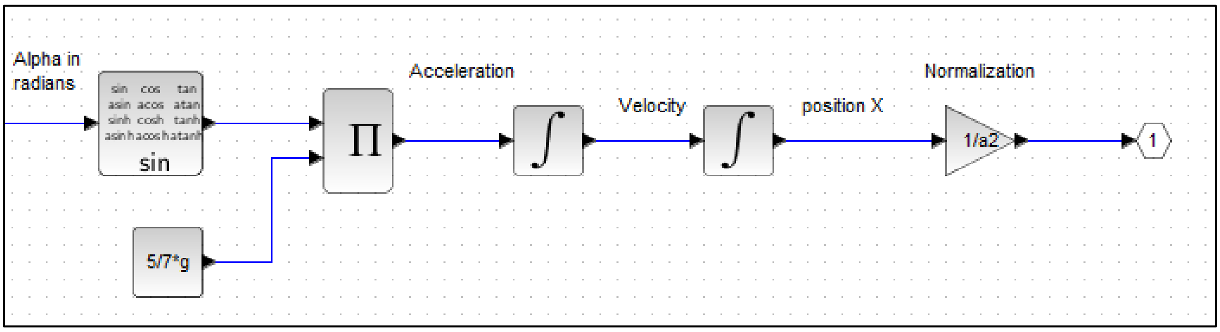
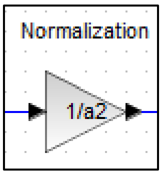


Figure 3.4 :Blocks to get position x from angle α .

4. Normalization of position:

The position of x is normalized between +1 and -1 (with 0 as the centre position). Hence after normalizing the upper limit of position is 1 and the lower limit is -1.



Normalization block

Continous PID Controller

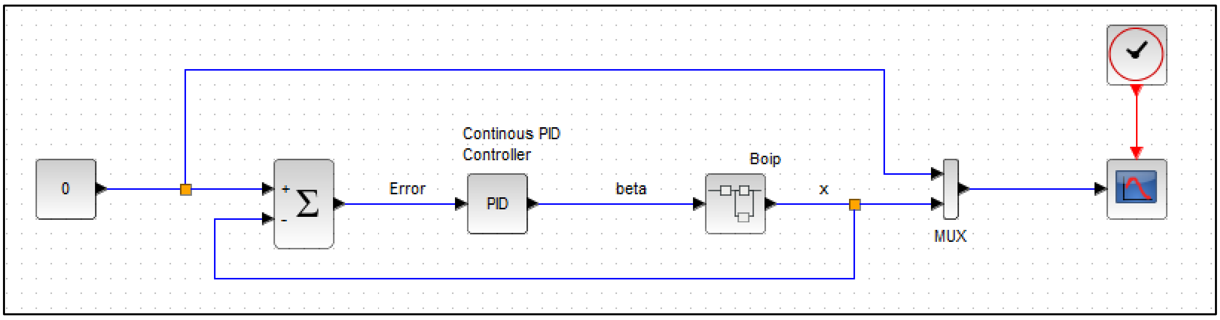


Figure 3.5 :Continous PID-Controller

The continuous PID controller output is defined as below.
Kp, Ki and Kd are the Proportional, Integral and Derivative constants used to tune the systems.
E -> is the error.

$$y(t) = \left[K_{p*}e(t) + K_I * \int_0^t e(t)\,dt + K_D * \frac{de(t)}{dt} \right]$$

The output of the Continous PID Controller,

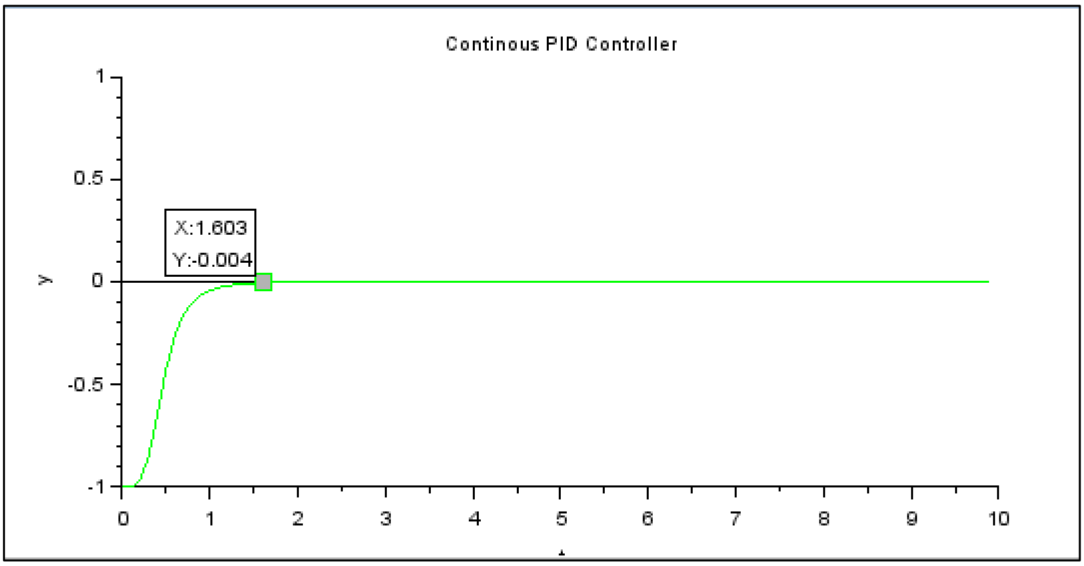


Figure 3.6 :Simulation Output of Continous PID-Controller

The PID values for the system are,

P = 4, I = 0.018, D = 1.30.

From the graph we can see that the settling time is 1.603s and there is no overshoot.

Closed loop control with Discrete PID Controller

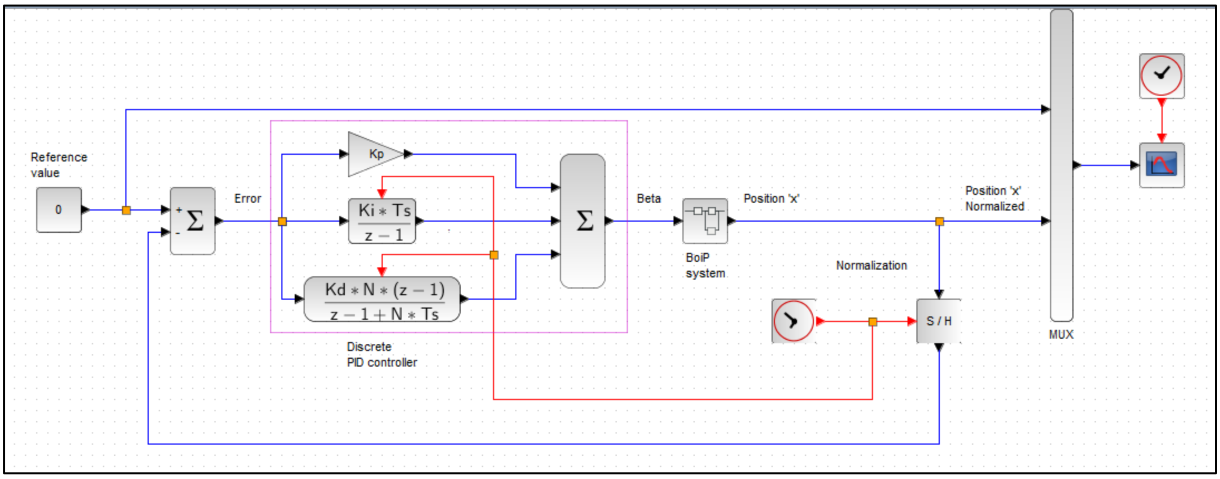


Figure 3.7 : Discrete PID-Controller

The PID controller is divided into individual blocks of Proportional, integral and derivative,

Integral term in Z domain : $\frac{K_I Ts}{z-1}$

Derviative form : $\frac{N(z-1)}{z-1+NTs}$

And the combined form of the proportional, integraland derivative is,

$$C(S) = K_P + \frac{K_I Ts}{z-1} + \frac{N(z-1)}{z-1+NTs}$$

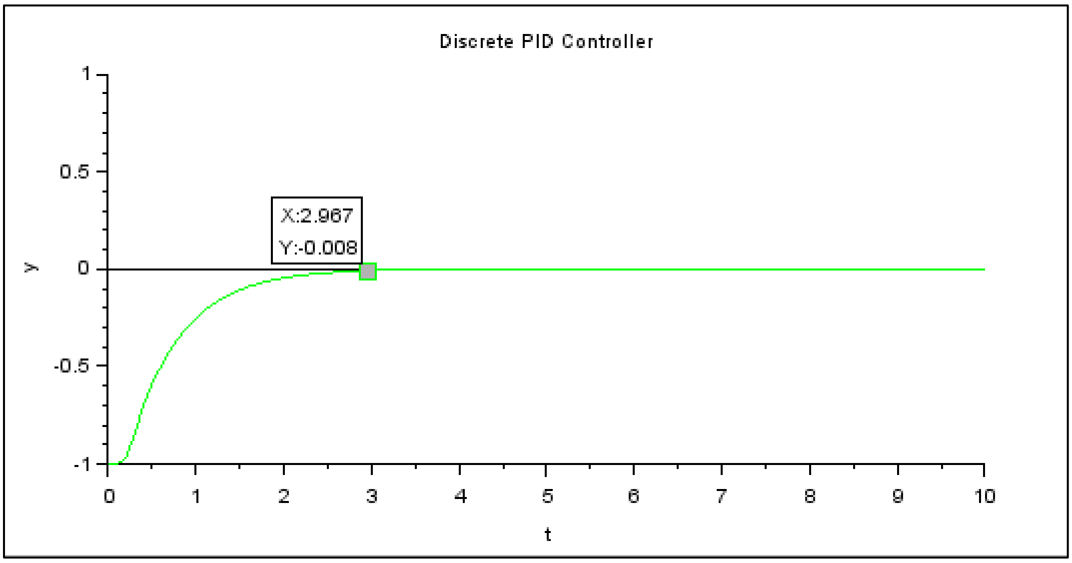


Figure 3.8 : Simulation output of Discrete PID-Controller

From the graph we can see that the settling time is 2.967s and there is no overshoot.

Closed loop control with Jumping reference

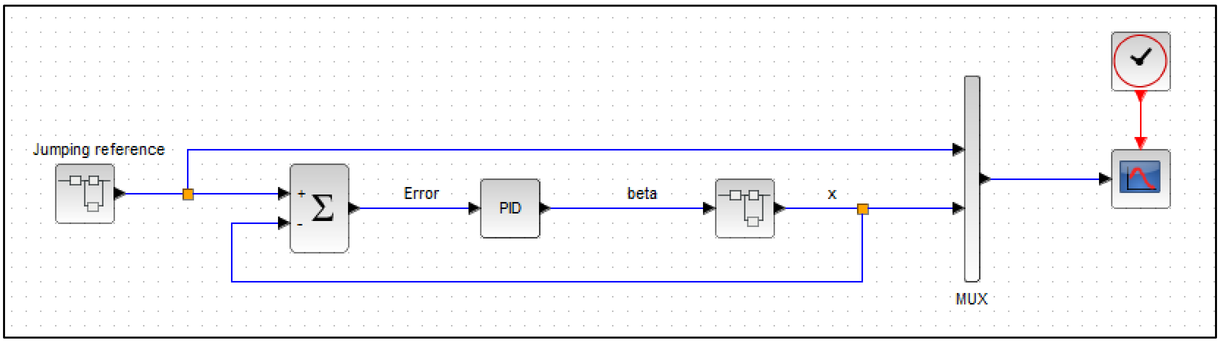


Figure 3.8 : Jumping reference for Continous PID Controller

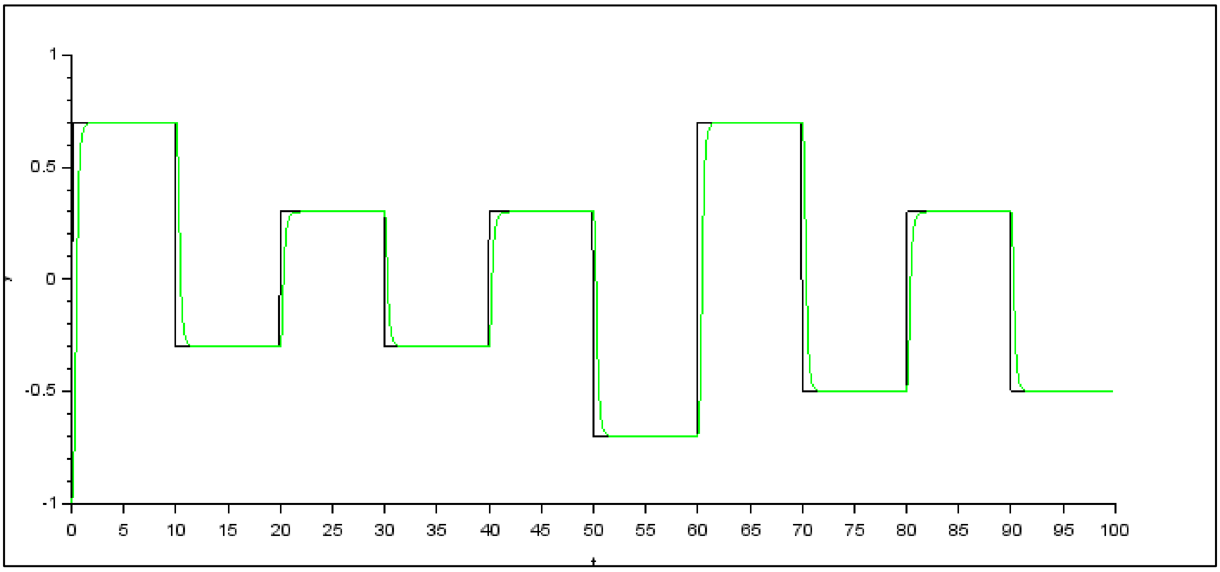


Figure 3.8 : Simulation Jumping reference for Continous PID Controller

4. Tasks

1. What are the expected responses of the controlled system when P-, I-, and D-controllers are individually tuned?

Effect of Proportional parameter(P):

- When proportional parameter is tuned individually by maintaining the integral and derivative parameter constant, where high gain is obtained. Increasing the proportional gain will increase the speed

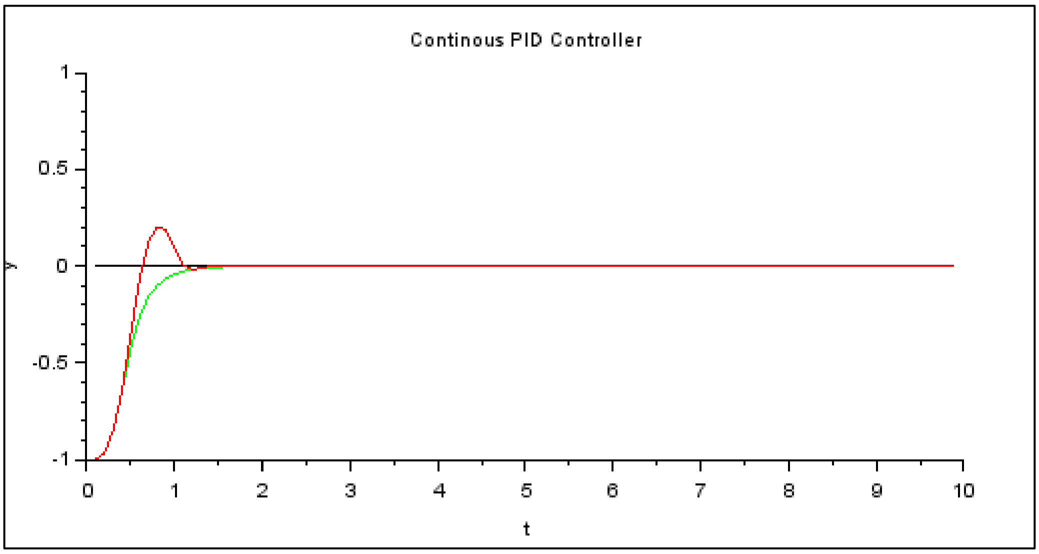


Figure 4.1 : Continous PID-Controller with different Proportional paramenter (P=4,10).

Of control gain response, if the proportional gain is too large, the process variable will begin to oscillate.

- So proper tuning of proportional gain along with the integral and derivative form leads to proper control of the system.

Effect of Integral parameter(I):

- In a PID controller the Integral parameter is the sum of the instantaneous error over time
- When integral parameter is tuned individually by maintaining the proportional and derivative parameter constant, offset settling time gets increased.
- This unsettled error error is then multiplied by the integral gain (Ki) and it is added to the controller output.

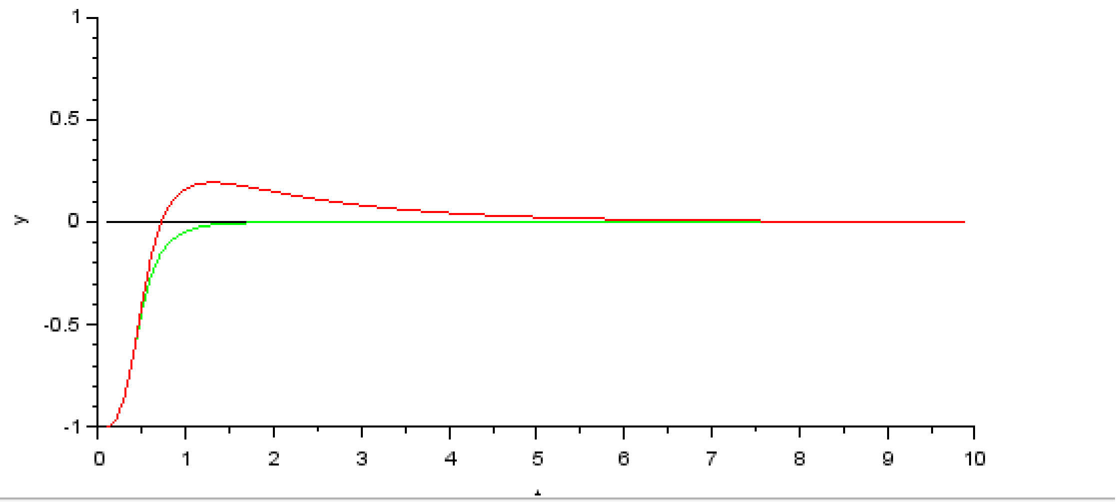


Figure 4.2 : Continous PID-Controller with different Integral paramenter(I = 0,2)

From the above simulation output the controller takes more time to settle as a stable system, as there are unsettled error fluctuating along the waveform

- So proper tuning of integral gain along with the proportional and derivative form leads to proper control of the system

Effect of Derivative parameter(D):

- Derivative component calculates the rate of change of error and multiplies it with the factor Kd. This component is higher when the error changes rapidly. It is smaller when the error changes slowly. Zero when error is constant
- When derivative parameter is tuned individually by maintaining the integral and proportional parameter constant, offset settling time gets increased.

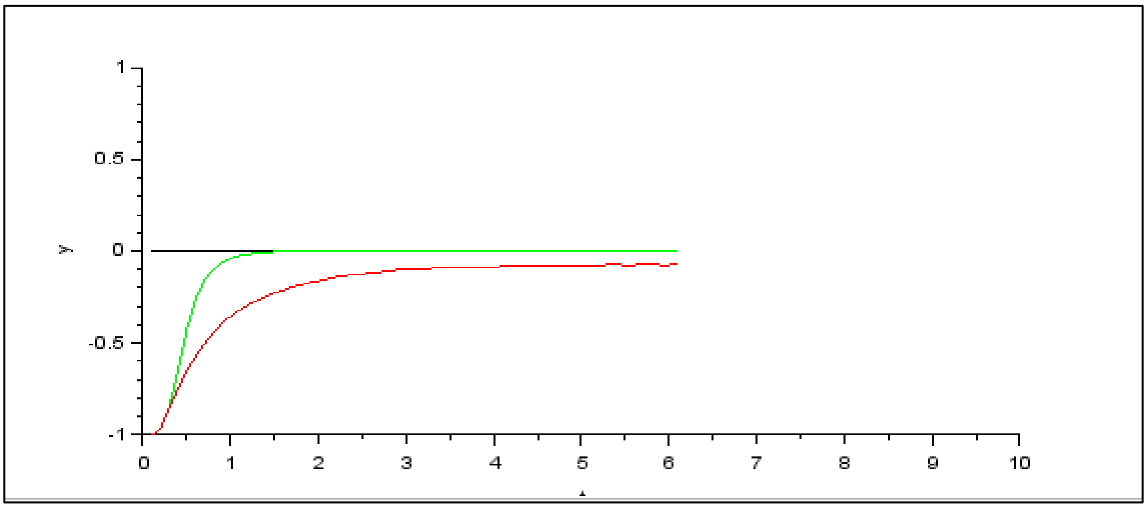


Figure 4.3 : Continous PID-Controller with different Derivative paramenter(D=1.30,4).

- The change in error continuously due to increase in derivative parameter makes the system going unstable.
- So proper tuning of integral gain along with the proportional and derivative form leads to proper control of the system

2. Why is it important to limit the input signal (angle β) to the controlled system? How can it be done? What should be the limitation?

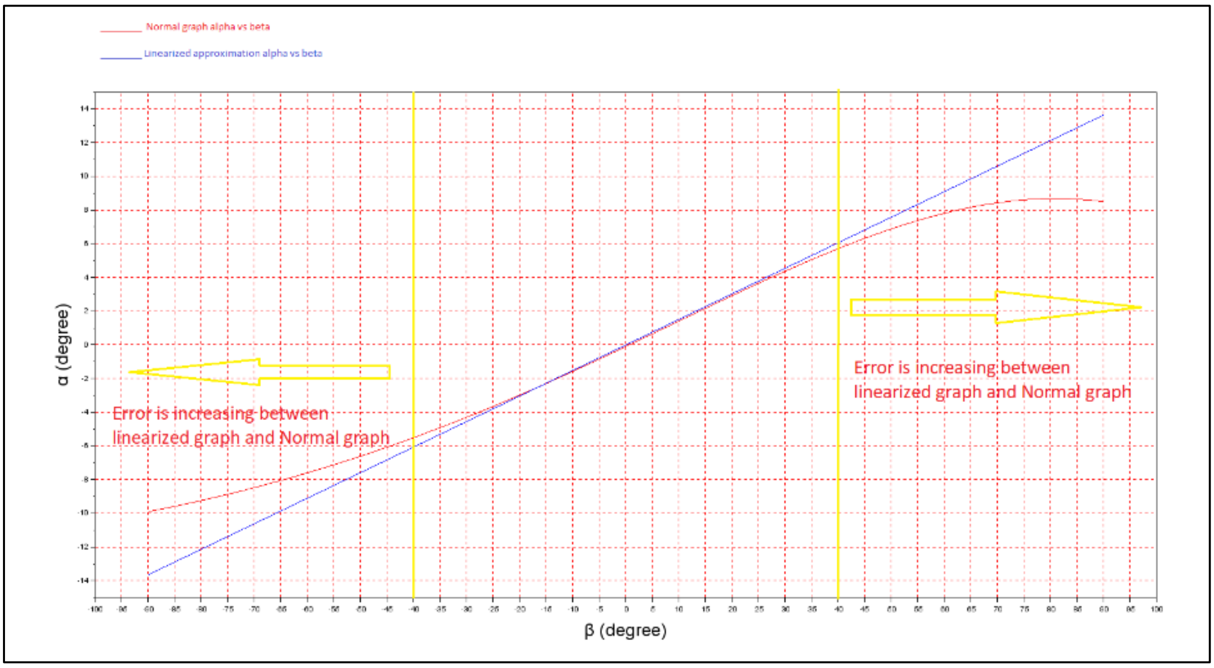


Figure 4.4 : Limiting th Beta(β) angle.

- From the plot above we can see that the actual curves (blue and red) are close to linear between -40 and 40 degrees. Beyond these values the curves start deviating from the linear approximated curve, that is the reason the limitation of the beta takes place.

3. Why and where do you use limited integrals in the controlled system? What are the limits?
   - We use the limits to calculate,
     I. Velocity from acceleration.
     II. Displacement(position) from velocity.

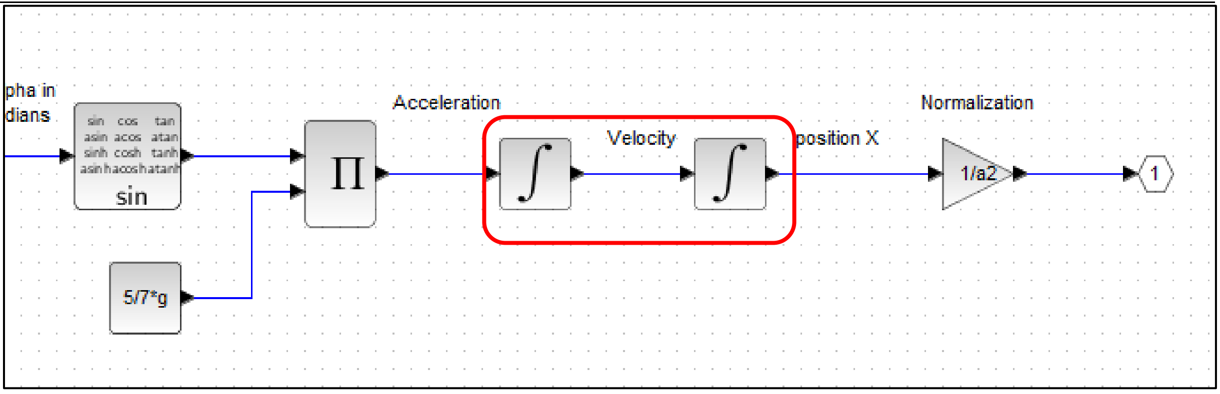
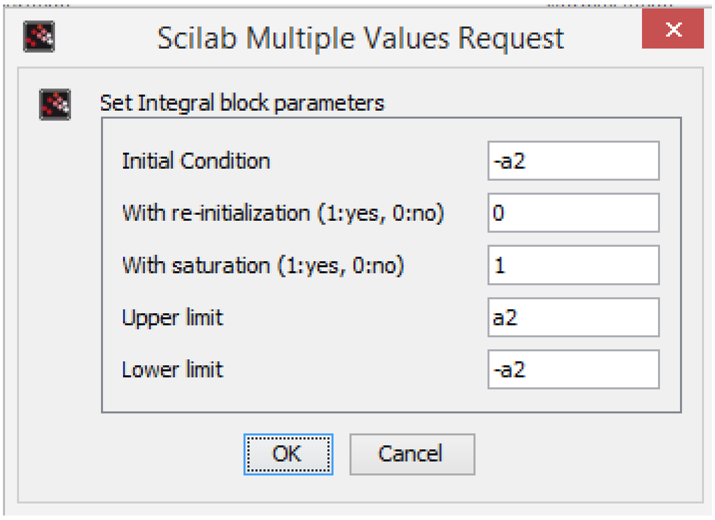


Figure 4.4 :Integrators in the control system

The integrators are indicted in red box,

- We generally use integrals as the ball has the velocity in a particular limits, in which these limits are given to the integrals.
- And the integral for the position is the ball defined with the certain limits of the position.
- Integral limits are defined as, the maximum velocity is given to the upper limit and the minimum velocity is given to the lower limits.



Integral limit

- From the above parameters the integral limits are given to maintain the position of the ball in the system.

4. Add a jumping reference for the system that randomly picks a new reference value from [0.3; 0.5; 0.7] and [-0.3; -0.5; -0.7] every 10s alternatively. An example may be [0.3; -0.5; 0.7; -0.3; 0.5; -0.3; ……]. Plot the desired ball positions and the actual ball positions for 10 jumps. What time period makes sense for the simulation interval? Can the system keep up? Why and why not?

The Continuous PID Controller with Jumping reference is shown below

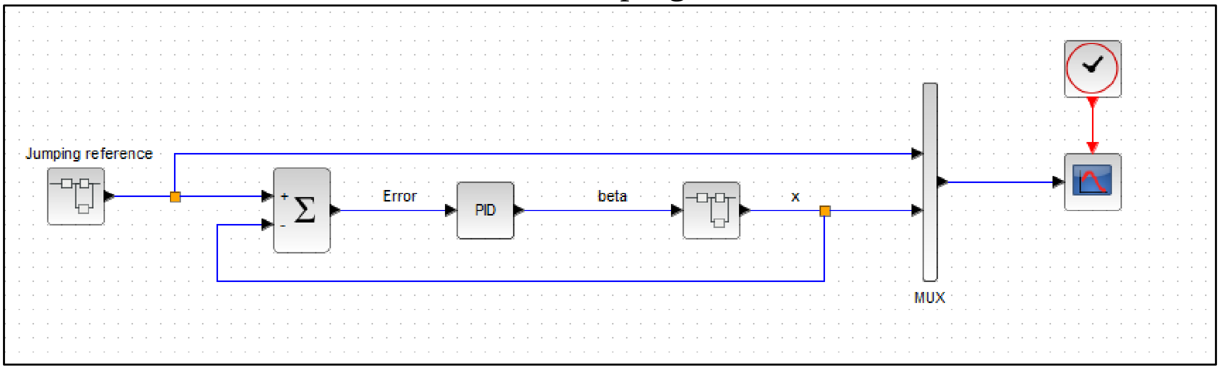


Figure 4.5 : Continuous PID controller with Jumping reference input values.

- The jumping reference block consists of the logic,

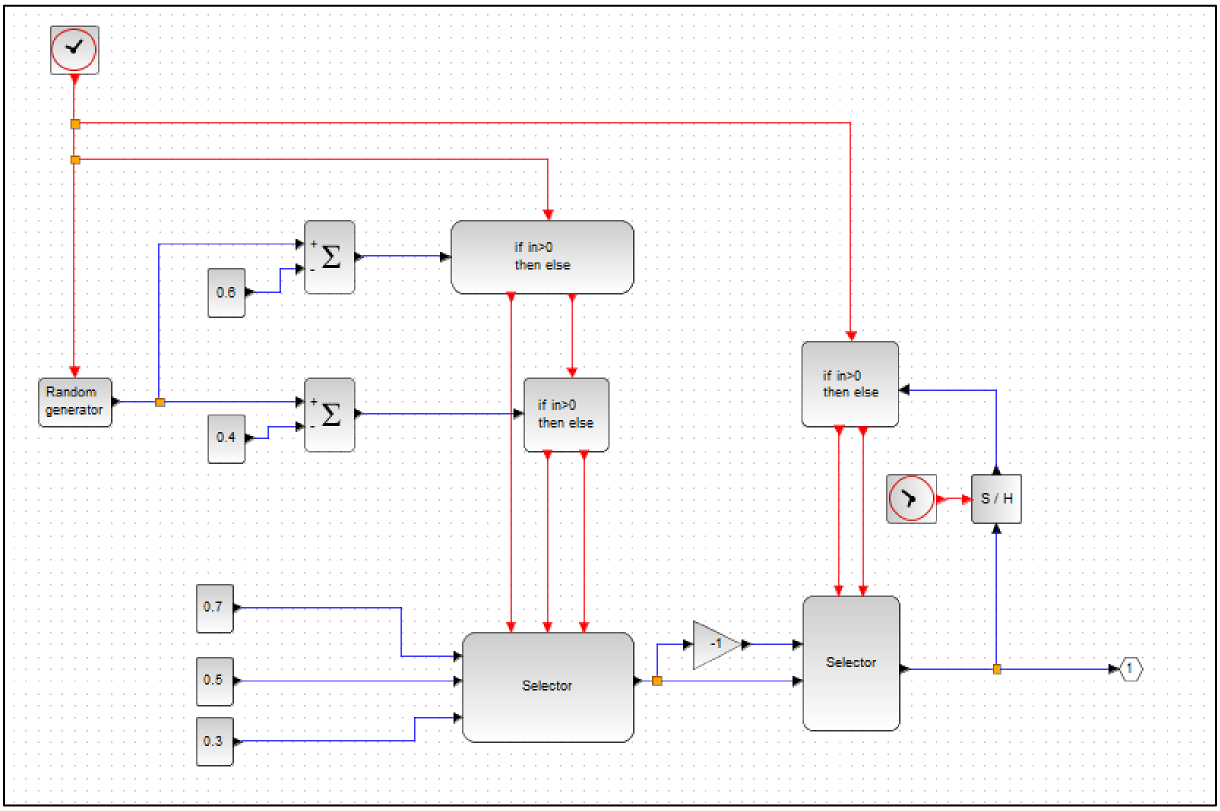


Figure 4.6 : Jumping reference block.

- Jumping reference block consists of ‘**Random generator’** which is used to generate the arbitrary numbers between the limits 0.2 and 0.8.

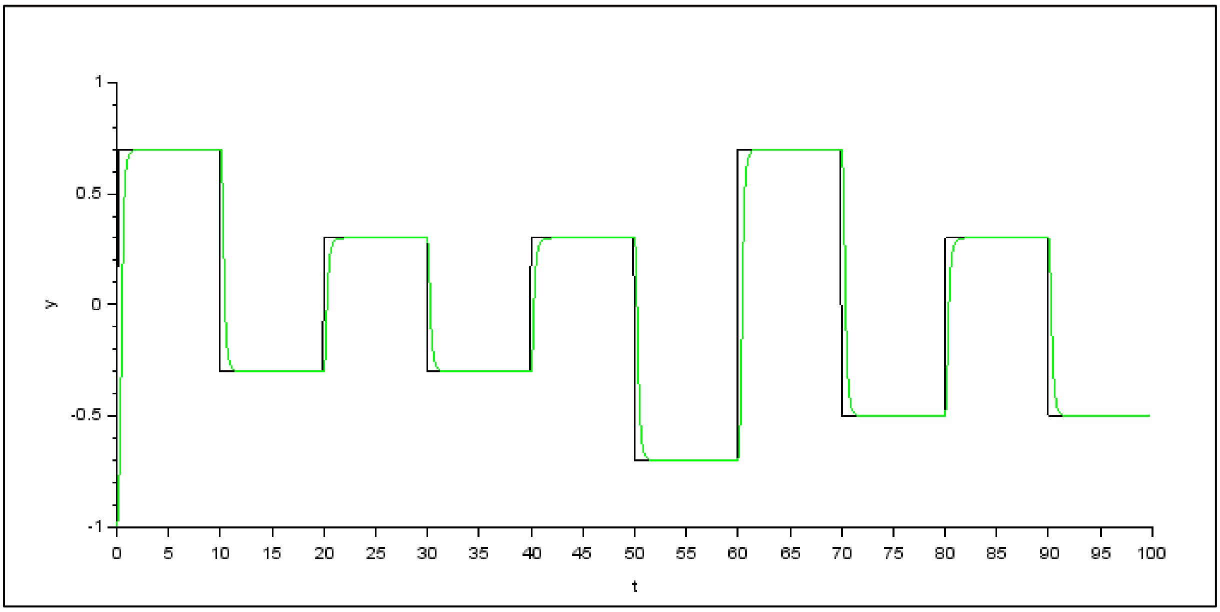


Figure 4.7 : Simulation of Continuous PID Controller with Jumping reference inputs.

- As seen from the plot the ball moves to the desired position with respect to the input valus given