Robot control design using fractional order control

Major Project Report

Submitted in Partial Fulfillment of the Requirements *for Semester-IV of*

Master of Technology

In

Instrumentation and Control Engineering

By

Yash B. Patel

14MICC21

Instrumentation and Control Engineering

Institute of Technology

NIRMA UNIVERSITY

Ahmedabad 382 481

MAY 2016

Declaration

This is to certify that

- The thesis comprises my original work towards the degree of Master of Technology in Instrumentation and control engineering at Nirma University and has not been submitted elsewhere for a degree.
- Due acknowledgement has been made in the text to all other material used.

Yash patel(14micc21)

Certificate

This is to certify that the Major Project Report entitled **"Robot control design using fractional order control"** submitted by **Yash B. Patel (14MICC21)**, towards the partial fulfillment of the requirements for the award of degree in **Master of Technology** in **Instrumentation and Control Engineering (Control and Automation)** of Nirma University is the record of work carried out by him under our supervision and guidance. The work submitted has reached a level required for being accepted for examination. The results embodied in this major project to the best of my knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

Date -

Place - Ahmedabad

Guide	**Program Coordinator**
____________	____________
Prof. Sandip A. Mehta	**Dr. Jignesh B.Patel**
Assistant Professor (I & C)	Sr. Associate Professor (I & C)
Institute of Technology	Institute of Technology
Nirma University	Nirma University

Head of Department (EE)	**Director,ITNU**
____________	____________
Dr. P.N.Tekwani	**Dr. P.N.Tekwani**
Institute of Technology	Institute of Technology
Nirma University	Nirma University

Acknowledgements

I feel deep sense of pleasure of submitting this report for person like me who is exposed by only to the four walls of the classroom two sides of a page of my prescribed textbook, project work is something new and unexplored experience, I would like to take opportunity to thank to **Instrumentation and Control Engineering department**

I would like to express my deepest gratitude to my internal guide **Prof. Sandip A. Mehata** and for their valuable guidance. It is only with their guidance that I could take up such a large project and transfer idea into a nice product.

I am lucky to represent myself and my work under the guidance of all the faculty members of my collage in MY project work period they gave me excellent opportunity under this direction. Their guidance at every step was steeping-stones in the completion of my project.

- Yash B. Patel

Abstract

In this project, design and implementation of self balancing two wheeled robot system has been done successfully. PID(Proportional integral derivative) control, FOPID(Fractional order proportional integral derivative) control and LQR(Linear quadratic regulator) has been done successfully on MATLAB simulation and do the comparative analysis. The PID and FOPID control algorithms are successfully implemented on the hardware(self balancing two wheeled robot) system and it has been proved that fractional order control gives better response compare to integer order control.

Contents

Declaration **ii**

Certificate **iii**

Acknowledgements **iv**

Abstract **v**

List of Figures **viii**

1 Introduction **2**

2 Motivation **4**

3 Objective And Block diagram **5**
3.1 Objective 5
3.2 Blockdiagram 5

4 System modelling **7**
4.1 system modelling 7
4.1.1 modelling of a DC motor 7
4.1.2 Mathematical model of two wheeled 9
4.1.3 Mathematical model of self balance robot's chassis 12
4.2 Convert integer order to fractional order system 16

5 Design different controller for self balancing system **18**
5.1 Linear Quadratic Regulator (LQR) control 18
5.1.1 Applying LQR controller on integer order system 19
5.1.2 Applying LQR controller on fractional order system 25
5.2 Apply PID and FOPID control algorithm 27

6 Hardware implementation of self balancing system **33**
6.1 Parts description 34
6.1.1 MPU6050 Sensor calibration 34
6.1.2 20A motor driver circuit 37

6.2 Connection diagram of hardware model 37
6.3 Implement PID and FOPID on hardware model 38

7 Conclusion and Future work 43

References 44

Index 44

List of Figures

3.1 project flow diagram 1 5
3.2 project flow diagram 2 6

4.1 Diagram of a DC motor 8
4.2 Free body diagram of the wheels 10
4.3 Diagram of the self balancing robot chassis 12

5.1 LQR control block diagram 20
5.2 Appropriate response with appropriate weighting 21
5.3 Response with high value of R 22
5.4 Response with high value of x(position) 23
5.5 Response with high value of chassis's angle 24
5.6 Response of the system when α is 0.9 25
5.7 Response of the system when α is 0.5 26
5.8 Response of the system when α is 0.1 27
5.9 Integer order model 28
5.10 Fractional order model 29
5.11 closed loop model 30
5.12 output response 30
5.13 closed loop model 31
5.14 output response 31
5.15 closed loop model 32
5.16 output response 32

6.1 Hardware of self balancing robot 33
6.2 MPU6050 gyro sensor 34
6.3 Block diagram of complementary filter 35
6.4 Simulink model of the complementary filter 36
6.5 Response of the complementary filter 36
6.6 20A motor driver circuit 37
6.7 system connection diagram 37
6.8 Simulink model of the system 38
6.9 PID response for the KP=4,KI=0.01,KD=0 39
6.10 PID response for the KP=5,KI=0.01,KD=0 40

6.11 PID response for the KP=7,KI=0.01,KD=0 40
6.12 FOPID response for the KP=4,KI=0.01,KD=0,alpha=0.1 41
6.13 FOPID response for the KP=4,KI=0.01,KD=0,alpha=0.5 41
6.14 FOPID response for the KP=4,KI=0.01,KD=0,alpha=0.9 42
6.15 Comparison of PID and FOPID control response 42

Chapter 1

Introduction

In modern calculus the differentiation term $D = \frac{d}{dx}$ is a well known tool.For function f the n^{th} derivative is a well-defined as $D^n f(x) = \frac{df(x)}{dx^n}$, here n is a positive integer number.we extend this concept like "if n is arbitrary,e.g. fractional then what happen",

In the recent years fractional order controller was not popular.when benefits stemming from using its concepts became evident in various scientific fields, including system modeling and automatic control. The rise of interest to the topic of fractional differentiation is also related to accessibility of more efficient and powerful computational tools. The introduction of computer algebra systems, such as MATLAB and Mathematic, led to new possibilities for evaluating the theoretical aspects of fractional calculus in specific application.

There are three definition for fractional calculus [5]

Definition 1 Riemann-Liouville definition

$$aD_t^{\alpha} f(t) = \frac{1}{\Gamma(m-\alpha)} (\frac{d}{dt})^m \int_a^t \frac{f^m(\tau)}{(t-\tau)^{\alpha-m+1}} d\tau$$

where $m - 1 < \alpha < m, m \in N, \alpha \in R^+, \Gamma(.) is a eulers gamma function$

Definition 2 Caputo definition

$$0D_t^{\alpha}f(t) = \frac{1}{\Gamma(m-\alpha)}\int_0^t \frac{f^m(\tau)}{(t-\tau)^{\alpha-m+1}}d\tau$$

where $m-1 < \alpha < m, m \in N$,

Definition 3 Grunwald Letnikov definition)

$$aD_t^{\alpha}f(t) = \lim_{h\to 0}\frac{1}{h^{\alpha}}\sum_{j=0}^{\ulcorner\frac{t-a}{h}\urcorner}(-1)^j\alpha j f(t-jh)$$

Chapter 2

Motivation

We found that PID control algorithm have wide acceptance and mainly in industries, but for faster response of the system and specially for nonlinear system PID control is not give desired output, so for that we found one new control algorithm is FOPID control algorithm.

It is remarkable to note the increasing the number of studies of fractional order controller .For the faster response of the system the fractional order controller which are very effective compare to PID controller in more precise and robust control performance.And in FOPID control have two more parameter compare to PID for tuning, and there are λ and μ and λ is varies between 0 to 1.

Chapter 3

Objective And Block diagram

3.1 Objective

Here in this project, modelling of self balancing two wheeled robot system and applied PID, LQR and FOPID control algorithm on MTLAB simulation. Then implement PID and FOPID controller on the self balancing hardware and do the comparative analysis.

3.2 Blockdiagram

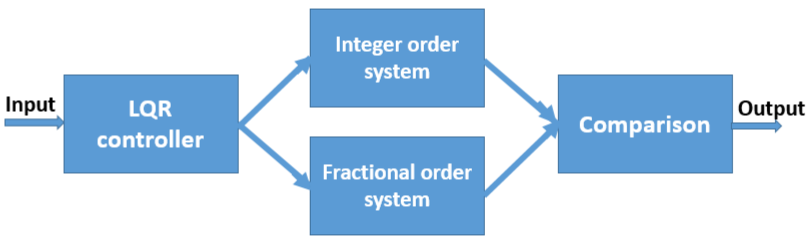

Figure 3.1: project flow diagram 1

The LQR control algorithm is applied on the integer and fractional order system and do the comparative analysis into the MATLAB simulation as shown in fig 3.1.

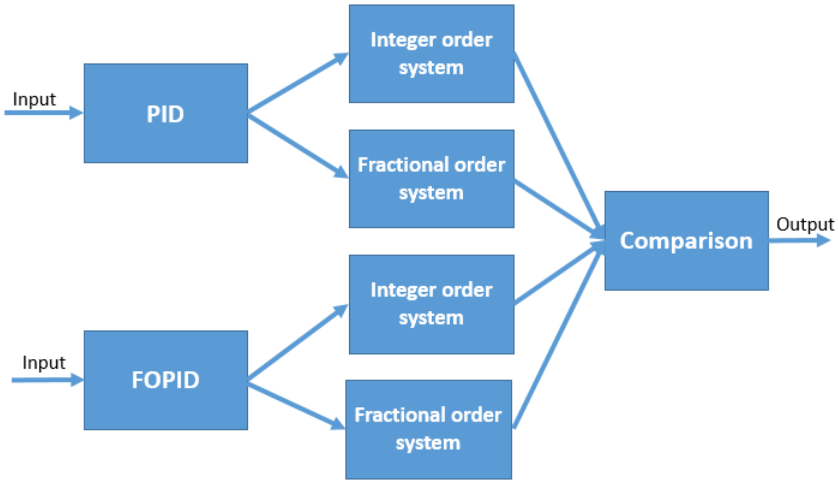

Figure 3.2: project flow diagram 2

The PID and FOPID control algorithm are applied on integer and fractional order system and do the comparative analysis into the MATLAB simulation. and also applied PID and FOPID control algorithm on self balancing two wheeled robot hardware system as shwon in fig 3.2.

Chapter 4

System modelling

4.1 system modelling

Here in this chapter mathematical model of the self balancing robot system has to be described, equation of inverted pendulum, wheeled and DC motor are derived in details.

4.1.1 modelling of a DC motor

[?] The self balancing robot is balancing on the two wheeled and this wheeled are connected with two DC motor. The mathematical equation of the DC motors are derived. These mathematical equations are provide relationship between input voltage to motor and required torque to balance the system. Free body diagram of the motor is shown in fig 4.1.

Motor torque is proportional to the current I, we can write as per the equation (4.1)

$$\tau_{motor} = K_{motor} * I......(4.1)$$

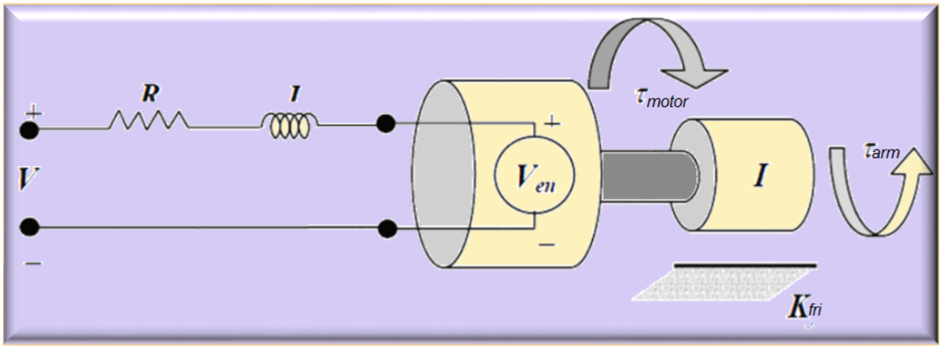

Figure 4.1: Diagram of a DC motor

Magnetic field is present into the motor, when Coil of the motor moving into the magnetic field back emf is produced,

$$V_{emf} = K_{emf} * \omega......(4.2)$$

Apply Kirchof voltage law at the figure 4.1

$$V - R.I - L\frac{dI}{dt} = 0.....(4.3)$$

Summation of the all the torques produce on the motor shaft are related to the acceleration of the shaft by the inertia load of the armature I_R,as per Newton's law of motion.

$$\tau_{motor} - K_{fri} * \omega - \tau_{arm} = I_R * \omega....(4.4)$$

Put equation (4.1) and (4.2) into(4.3) and (4.4),

$$\frac{dI}{dt} = \frac{R}{L}I + \frac{K_{emf}}{L}\omega + \frac{V}{L}...(4.5)$$

$$\frac{d\omega}{dt} = \frac{K_{motor}}{I_R}i + \frac{-K_{emf}}{I_R}\omega - \frac{\tau_{arm}}{I_R}...(4.6)$$

Equation (4.5) and (4.6) are function of Velocity and I and its have first order derivative therefore motor inductance and friction are negligible,

$$I = \frac{-K_{emf}}{R}\omega + \frac{1}{R}V.....(4.7)$$

$$\frac{d\omega}{dt} = \frac{K_{motor}}{I_R}I - \frac{\tau_{arm}}{I_R}.....(4.8)$$

put equation (4.7) into (4.8),

$$\frac{d\omega}{dt} = -\frac{K_{motor}.K_{emf}}{I_R.R}\omega + \frac{1}{I_R.R}V - \frac{\tau_{arm}}{I_R}...(4.9)$$

The DC motor mathematical equation can represented with a state space model, and this system have parameters are position θ and velocity ω, as shown in equation (4.10) and (4.11)

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-K_{motor}.K_{emf}}{I_R.R} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{K_{motor}}{I_R.R} & \frac{-1}{I_R} \end{bmatrix} \begin{bmatrix} V \\ \tau_{arm} \end{bmatrix}...(4.10)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} V \\ \tau_{arm} \end{bmatrix}...(4.11)$$

4.1.2 Mathematical model of two wheeled

[?]

Here in this section we have to derived mathematical equation of the left and right wheel, free body diagram of the wheeled as shown in the fig 4.2.

Summation of the force on the horizontal direction as per the Newton's law is

$$F = m.a$$

$$M_w\ddot{x} = F_r - G_r.....(4.12)$$

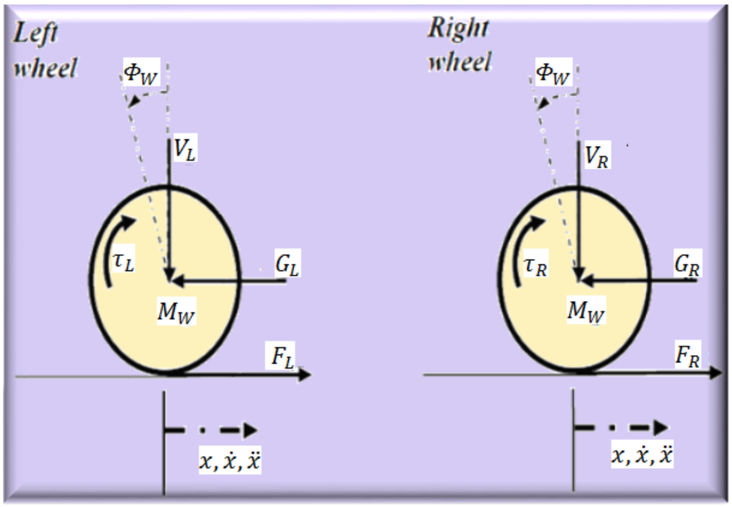

Figure 4.2: Free body diagram of the wheels

Here F_r is friction or conflict force between the plane and right wheel and G_r is back force between the right wheel and robot chassis,
Summation of the force acting on the wheel's center is

$$M = I.\alpha$$

$$I_w\ddot{\phi}_w = \tau_R - F_r.r.....(4.13)$$

Where I_w is wheel's moment of inertia, $\ddot{\phi}_w$ is angular acceleration and τ_R is applied torque from the motor to wheel.
As per above derivation of the motor torque which derived in the mathematical model of DC motor(as per equation (4.8))

$$\tau_{motor} = I_R\frac{dw}{dt} + \tau_{arm}....(4.14)$$

substitute value $\frac{dw}{dt}$ from equation (4.9)

$$\tau_w = -\frac{K_{motor}.K_{emf}}{R}\dot{\phi_w} + \frac{K_{motor}}{R}V....(4.15)$$

substitute value of τ from the equation (4.15) to equation (4.13)

$$I_w\ddot{\phi_w} = -\frac{K_{motor}K_{emf}}{R}\dot{\phi_w} + \frac{K_{motor}}{R}V - F_r.r...(4.16)$$

Thus,

$$F_r = -\frac{K_{motor}.K_{emf}}{Rr}\dot{\phi_w} + \frac{K_{motor}}{R.r}V - \frac{I_w}{r}\ddot{\phi_w}...(4.17)$$

substitute F_r value from equation (4.17) to equation (4.12)

Equation of left side wheel

$$M_w\ddot{x} = -\frac{K_{motor}.K_{emf}}{R.r}\dot{\phi_w} + \frac{K_{motor}}{R.r}V - \frac{I_w}{r}\ddot{\phi_w} - G_L...(4.18)$$

Equation of right side wheel

$$M_w\ddot{x} = -\frac{K_{motor}.K_{emf}}{R.r}\dot{\phi_w} + \frac{K_{motor}}{R.r}V - \frac{I_w}{r}\ddot{\phi_w} - G_R...(4.19)$$

Here we have to transform angular rotation to linear motion, for that we can used below equation

$$\ddot{\phi_w}.r = \ddot{x} \Longrightarrow \ddot{\phi_w} = \frac{\ddot{x}}{r}$$

$$\dot{\phi_w}r = \dot{x} \Longrightarrow \dot{\phi_w} = \frac{\dot{x}}{r}$$

After this transformation above left and right side wheel equation (4.18) and (4.19) becomes like,

Equation of left side wheel

$$M_w\ddot{x} = -\frac{K_{motor}.K_{emf}}{R.r}\frac{\dot{x}}{r} + \frac{K_{motor}}{R.r}V - \frac{I_w}{r}\frac{\ddot{x}}{r} - G_L...(4.20)$$

Equation of right side wheel

$$M_w\ddot{x} = -\frac{K_{motor}.K_{emf}}{R.r}\frac{\dot{x}}{r} + \frac{K_{motor}}{R.r}V - \frac{I_w}{r}\frac{\ddot{x}}{r} - G_R ...(4.21)$$

After the summation of the equation (4.20) and (4.21) we get,

$$2(M_w + \frac{I_w}{r^2})\ddot{x} = -\frac{2.K_{motor}K_{emf}}{R.r}\frac{\dot{x}}{r} + \frac{2K_{motor}}{R.r}V - (G_L + G_R)....(4.22)$$

4.1.3 Mathematical model of self balance robot's chassis

[?]

Here we have to derived mathematical equation of the robot's chassis, fig 4.3 display free body diagram of the self balancing robot chassis,

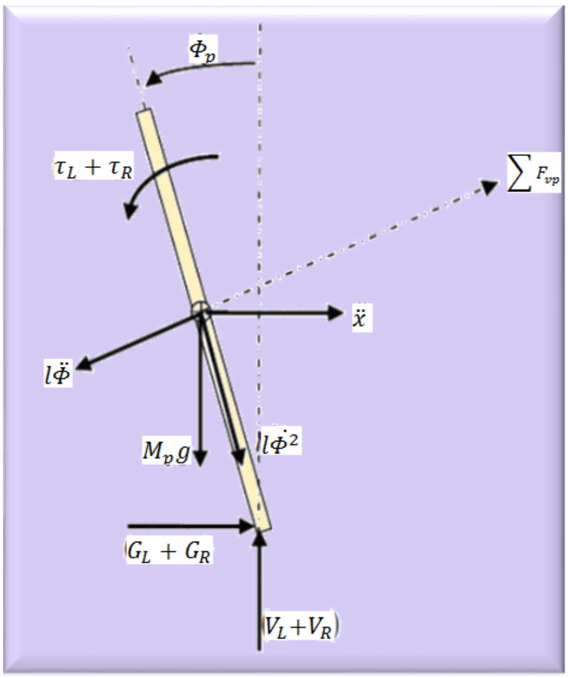

Figure 4.3: Diagram of the self balancing robot chassis

As per Newton's law of motion,

$$F_x = M_p.\ddot{x}$$

$$(G_L + G_R) - lM_P\ddot{\phi}_p \cos\phi_p + lM_P\dot{\phi}_p^{\,2} \sin\phi_p = M_p.\ddot{x} ....(4.23)$$

Thus

$$(G_L + G_R) = M_p.\ddot{x} + lM_P\ddot{\phi}_p \cos\phi_p - lM_P\dot{\phi}_p^{\,2} \sin\phi_p ....(4.24)$$

Summation of forces acting vertical to the chassis are

$$\sum F_{vp} = M_p.\ddot{x}\cos\phi_p$$

$$(G_L + G_R)\cos\phi_p + (V_L + V_R)\sin\phi_p - M_p g \sin\phi_p - lM_p\ddot{\phi}_p = M_p\ddot{x}\cos\phi_p ....(4.25)$$

Summation of moments to the center of mass of self balancing robot's chassis,

$$\sum M = I.\alpha$$

$$-(G_L + G_R)l\cos\phi_p - (V_L + V_R)l\sin\phi_p - (\tau_L + \tau_R) = I_p\ddot{\phi}_p ....(4.26)$$

The torque applied on the robot's chassis from the right motor and left motor which defined in the equation (4.15), and after the linearised its becomes

$$(\tau_L + \tau_R) = -\frac{2K_{motor}K_{emf}}{R.r}\frac{\dot{x}}{r} + \frac{2K_{motor}}{R.r}V$$

put the value of $\tau_L + \tau_R$ in equation (4.26),

$$-(G_L + G_R)l\cos\phi_p - (V_L + V_R)l\sin\phi_p - \left(-\frac{2K_{motor}.K_{emf}}{R.r}\frac{\dot{x}}{r} + \frac{2K_{motor}}{R.r}V\right) = I_p\ddot{\phi}_p$$

thus

$$-(G_L + G_R)l\cos\phi_p - (V_L + V_R)l\sin\phi_p = I_p\ddot{\phi}_p - \frac{2K_{motor}.K_{emf}}{R.r}\frac{\dot{x}}{r} + \frac{2K_{motor}}{R.r}V ....(4.27)$$

Both the side multiplying with l to the equation (4.25)

$$(G_L + G_R)l\cos\phi_p + (V_L + V_R)l\sin\phi_p - lM_p g \sin\phi_p - M_p l^2\ddot{\phi}_p = M_p l\ddot{x}\cos\phi_p ....(4.28)$$

Put the value of equation (4.27) in equation (4.28)

$$I_p\ddot{\phi}_p - \frac{2K_{motor}K_{emf}}{R.r}\frac{\dot{x}}{r} + \frac{2K_{motor}}{R.r}V - lM_pg\sin\phi_p - M_pl^2\ddot{\phi}_p = M_pl\ddot{x}\cos\phi_p....(4.29)$$

Put the value of(GL+GR) from the equation (4.24) into the (4.22)

$$2(M_w+\frac{I_w}{r^2})\ddot{x} = -\frac{2K_{motor}.K_{emf}}{R.r}\frac{\dot{x}}{r}+\frac{2K_{motor}}{R.r}V-M_p\ddot{x}-M_pl\ddot{\phi}_p\cos\phi_p+M_pl\dot{\phi}_p^{\ 2}\sin\phi_p....(4.30)$$

Rearranging equation (4.29) and (4.30)

$$(I_p+M_pl^2)\ddot{\phi}_p - \frac{2K_{motor}K_{emf}}{R}\frac{\dot{x}}{r} + \frac{2K_{motor}}{R.r}V + lM_pg\sin\phi_p = -M_pl\ddot{x}\cos\phi_p....(4.31)$$

$$\frac{2K_{motor}}{R.r}V = (2M_w+\frac{2I_w}{r^2}+2M_p)\ddot{x}+\frac{2K_{motor}K_{emf}}{R}\frac{\dot{x}}{r}+M_pl\ddot{\phi}_p\cos\phi_p-M_pl\dot{\phi}_p^{\ 2}\sin\phi_p...(4.32)$$

Above equation (4.31) and (4.32) are nonlinear, so for linearising we have to assume $\phi_p = \pi + \phi$,

$$\cos\phi_p = -1, \sin\phi_p = -\phi, and(\frac{d\phi_p}{dt})^2 = 0$$

After the linearised these equation of motion are

$$(I_p+M_pl^2)\ddot{\phi} - \frac{2K_{motor}.K_{emf}}{R}\frac{\dot{x}}{r} + \frac{2K_{motor}}{R.r}V + lM_pg\phi_p = -M_pl\ddot{x}.....(4.33)$$

$$\frac{2K_{motor}}{R.r}V = (2M_w + \frac{2I_w}{r^2} + 2M_p)\ddot{x} + \frac{2K_{motor}.K_{emf}}{R}\frac{\dot{x}}{r} - M_pl\ddot{\phi}.....(4.34)$$

When rearranging equation (4.33) and (4.34) for the state space representation of the system its becomes

$$\ddot{\phi} = \frac{M_pl}{(I_p+M_pl^2)}\ddot{x} + \frac{2K_{emf}.K_{motor}}{R(I_p+M_pl^2)}\frac{\dot{x}}{r} - \frac{2K_{motor}}{R(I_p+M_pl^2)}v + \frac{M_pgl}{I_p+M_pl^2}\phi..(4.35)$$

$$\ddot{x} = \frac{2K_{motor}}{R(\frac{2I_w}{r^2}+2M_w+M_p)r}v + \frac{2K_{emf}K_{motor}}{R(\frac{2I_w}{r^2}+2M_w+M_p)r^2}\dot{x} - \frac{M_pl}{(\frac{2I_w}{r^2}+2M_w+M_p)}\ddot{\phi}..(4.36)$$

Put the value of (4.35)into(4.34)and (4.36) into (4.33)

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2K_{motor}K_{emf}(M_p lr - M_p l^2)}{Rr^2\alpha} & \frac{{M_p}^2 g l^2}{\alpha} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2K_{motor}K_{emf}(r\beta - M_p l)}{Rr^2\alpha} & \frac{M_p g l \beta}{\alpha} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2K_{motor}(I_p + M_p l^2 - M_P lr)}{Rr\alpha} \\ 0 \\ \frac{2K_{motor}(M_p l - r\beta)}{Rr\alpha} \end{bmatrix} V..(4.37)$$

where

$$\beta = (\frac{2I_w}{r^2} + 2M_w + M_p), \alpha = [I_p\beta + 2l^2 M_p(\frac{I_w}{r^2} + M_w)]$$

Here

$M_w = 0.04$,

$r = 0.052$,

$I_w = 0.000049$,

$I_p = 0.0042$,

$M_p = 1.14$,

$l = 0.08$,

$K_{emf} = 0.006091$,

$R = 4$,

$g = 9.81$,

$K_{motor} = 0.006131$,

Put these all above value in equation (4.37),we get

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0076 & 13.3230 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.0292 & 183.5182 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0.0650 \\ 0 \\ 0.2491 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

4.2 Convert integer order to fractional order system

Here we convert system from integer order to fractional order system.[1]
Representation of state space model for integer order system is shown in below,
state equation

$$\dot{x} = Ax + Bu$$

Representation of state space model for fractional order system is shown below,
state equation

$$D_x^{\alpha} = Ax + Bu$$

Where$\alpha = [\alpha 1, \alpha 2, \alpha 3......\alpha n]$,A is state matrix, B is input matrix,C is output matrix and D is the direct transmission matrix.
We have one laplace transform's properties of fractional order is,[2]

$$L[aD_t^{\alpha}.F(t)] = s^{\alpha}.L[F(t)]....(4.38)$$

[6]
Here we have to rearranging equation (4.38),

$$[aD_t^{\alpha}.F(t)] = L^{-1}[s^{\alpha}.L[F(t)]]$$

$$[aD_t^{\alpha}.F(t)] = L^{-1}[s^{\alpha}].L^{-1}[L[F(t)]]$$

$$[aD_t^{\alpha}.F(t)] = L^{-1}[s^{\alpha}].[F(t)]...(4.39)$$

as per the laplace transform properties $L^{-1}[s^{\alpha}] = \frac{\Gamma(\alpha)}{t^{\alpha-1}}$
Put this above laplace transform properties into the equation (4.39) and we get new equation as shown below in equation (4.40)

$$[aD_t^{\alpha}.F(t)] = \frac{\Gamma(\alpha)}{t^{\alpha-1}}.[F(t)]...(4.41)$$

We can write these two equation as shown below from the integer order state space model,

$$\ddot{x} = -0.0076.\dot{x} + 13.3230.\phi + 0.0650.V_a....(4.42)$$

$$\ddot{\phi} = -0.0292.\dot{x} + 183.5182.\phi + 0.2491.V_a...(4.43)$$

For the fractional state space model we have to find out $[aD_t^{\alpha}.\ddot{x}]$ and $[aD_t^{\alpha}.\ddot{\phi}]$ using equation (4.41),(4.42) and (4.43). And we can write fractional state space equation are

$$[aD_t^{\alpha}.\ddot{x}] = -0.0076.\frac{\Gamma(\alpha)}{t^{\alpha-1}}.\dot{x} + 13.3230.\frac{\Gamma(\alpha)}{t^{\alpha-1}}.\phi + 0.0650.\frac{\Gamma(\alpha)}{t^{\alpha-1}}.V_a....(4.44)$$

$$[aD_t^{\alpha}.\ddot{\phi}] = -0.0292.\frac{\Gamma(\alpha)}{t^{\alpha-1}}.\dot{x} + 183.5182.\frac{\Gamma(\alpha)}{t^{\alpha-1}}.\phi + 0.2491.\frac{\Gamma(\alpha)}{t^{\alpha-1}}.V_a...(4.45)$$

From the above two equation we can write fractional state space model is shown in equation(4.46)

$$\begin{bmatrix} aD_t^{\alpha}\dot{x} \\ aD_t^{\alpha}.\ddot{x} \\ aD_t^{\alpha}.\dot{\phi} \\ aD_t^{\alpha}.\ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0076.\frac{\Gamma(\alpha)}{t^{\alpha-1}} & 13.3230.\frac{\Gamma(\alpha)}{t^{\alpha-1}} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.0292.\frac{\Gamma(\alpha)}{t^{\alpha-1}} & 183.5182.\frac{\Gamma(\alpha)}{t^{\alpha-1}} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.0650.\frac{\Gamma(\alpha)}{t^{\alpha-1}} \\ 0 \\ 0.2491.\frac{\Gamma(\alpha)}{t^{\alpha-1}} \end{bmatrix} V_a..(4.46)$$

Chapter 5

Design different controller for self balancing system

The self balancing robot system described in chapter 4 is nonlinear and fully unstable system, control objective of this system is always challenging. Here in this chapter we applied LQR(Linear Quadratic Regulator), PID(Proportional Integral Derivative) and FOPID(Fractional Order Proportional Integral Derivative) control algorithm on the integer order system and fractional order system and do the comparative analysis on the MATLAB simulation.

5.1 Linear Quadratic Regulator (LQR) control

The LQR control is a well known state space technique for designing robust and optimal regulators. LQR control refers only linear system and quadratic performance index as per the,

$$\dot{x}(t) = Ax(t) + Bu(t)...(5.1)$$

$$x(0) = x_0...(5.2)$$

Here below equation (5.3) is representing integral performance index

$$J = \frac{1}{2}\int_0^{\infty}[x'(t)Qx(t) + U'(t)RU(t)]dt...(5.3)$$

Linear quadratic regulator control law is

$$u = -R^{-1}B'\bar{P}x...(5.4)$$

Where $\bar{P} = \bar{P}\prime \geq 0$, and we have to solve the Ricati equation as shown in equation(5.5)

$$0 = PA + A'P - PBR^{-1}B'P + Q...(5.5)$$

Here in LQR control have one gain vector is K, and its determine some amount of feedback into the system and $K = R^{-1}B'\bar{P}$. Here for LQR have two another tuning parameter are Q and R, and its value are always positive,Q matrix value is depend on the size of the system state matrix, and R matrix size depend on the system's control input. Here fig 5.1 is shown the block diagram of the linear quadratic regulator control.

5.1.1 Applying LQR controller on integer order system

First of all we have to check system is controllable or uncontrollable,controllability is depend upon the matrix A and B, if rank of $CB = [BABA^{n-1}B]$ is N,here N is no state of the self balancing system.

MATLAB command for checking controllability

>>CB=ctrb(A,B)

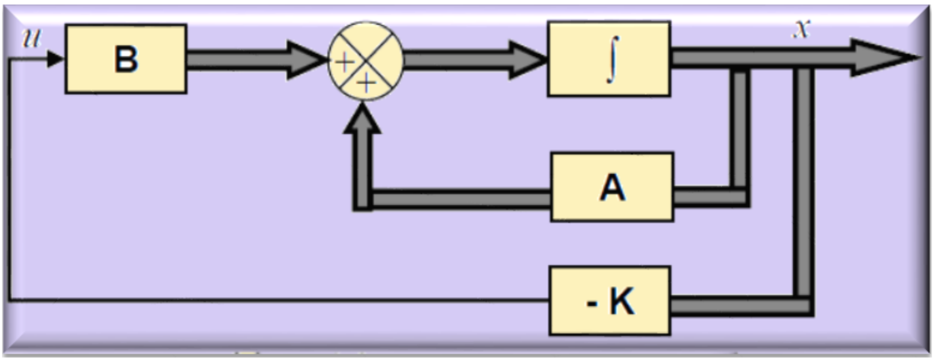

Figure 5.1: LQR control block diagram

>>rank(CB)

Here rank of CB is 4 and system state is also 4 it means system is controllable, so we can apply LQR control algorithm on the self balancing robot system.

For LQR we have to choose two parameter Q and R, normally we assume R=1 and Q=c*c',here our system have 4 state so Q matrix is like

$$Q = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{bmatrix}$$

Here for our system $a = x$,$b = \dot{x}$,$c = \phi$ and $d = \dot{\phi}$, the value of $Q = c' * c$ matrix is

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

Here in the Q matrix position weighting is (1x1) and chassis angle weighting is (3x3) When we apply LQR control algorithm on the system and solve the ricati equation for position(1x1) weighting is 1000,chassis's angle(3x3) weighting is 5000 and R=1 and plot the response for the system state, response shown below for the different weighting

State response

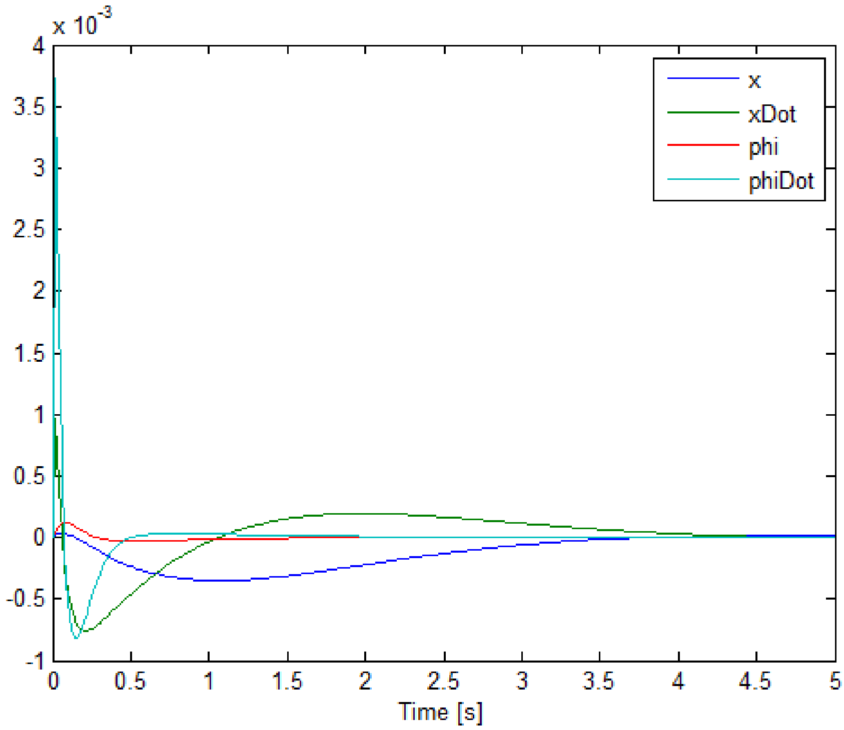

Figure 5.2: Appropriate response with appropriate weighting

Here fig 5.2 shows the most acceptable response, here weighting of position is 1000, chassis's angle weighting is 5000 and R=1,

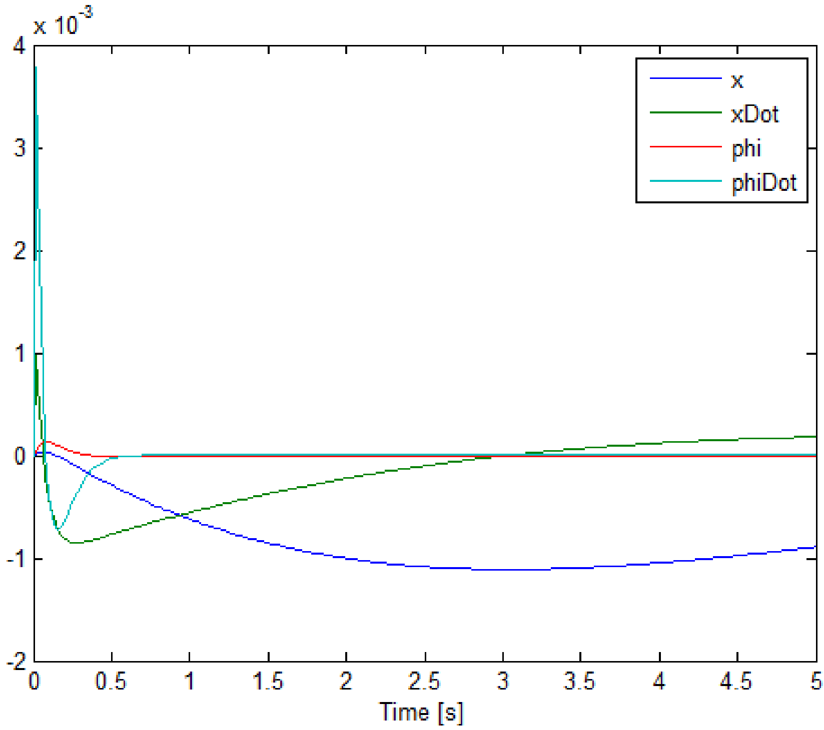

Figure 5.3: Response with high value of R

Here fig 5.3 shows the response with high weighting of R, and R value is 100, here we can see the less motor control, this response in a low gain value for x and $\dot{x}$, it causes the self balancing robot to continuously moving.

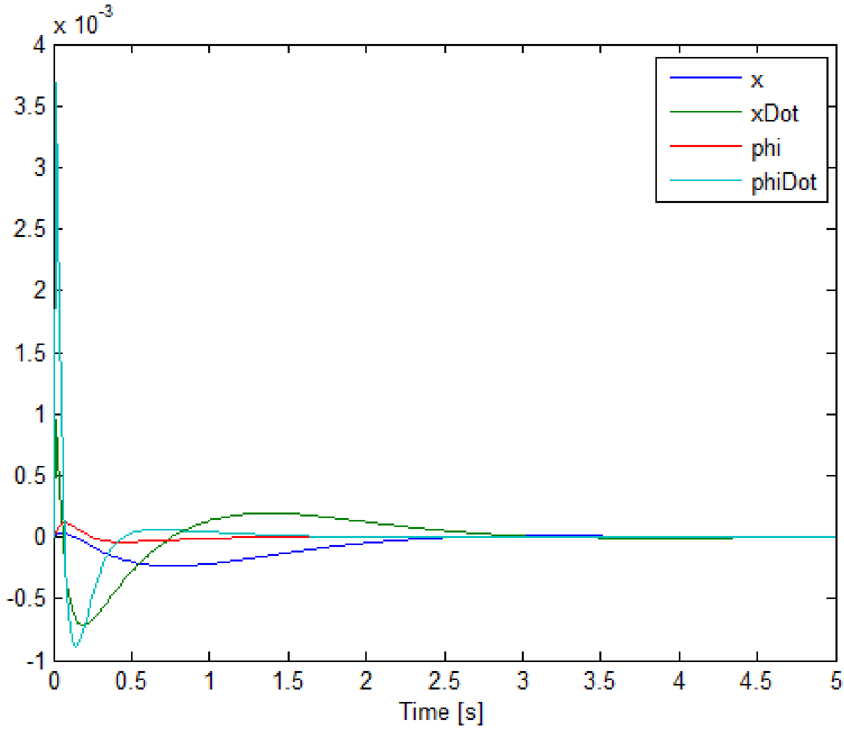

Figure 5.4: Response with high value of x(position)

Here fig 5.4 shows the response with high value of position, we can see the settling time is very less to every state response, but motor will not get appropriate response because motor have to required maximum torque.

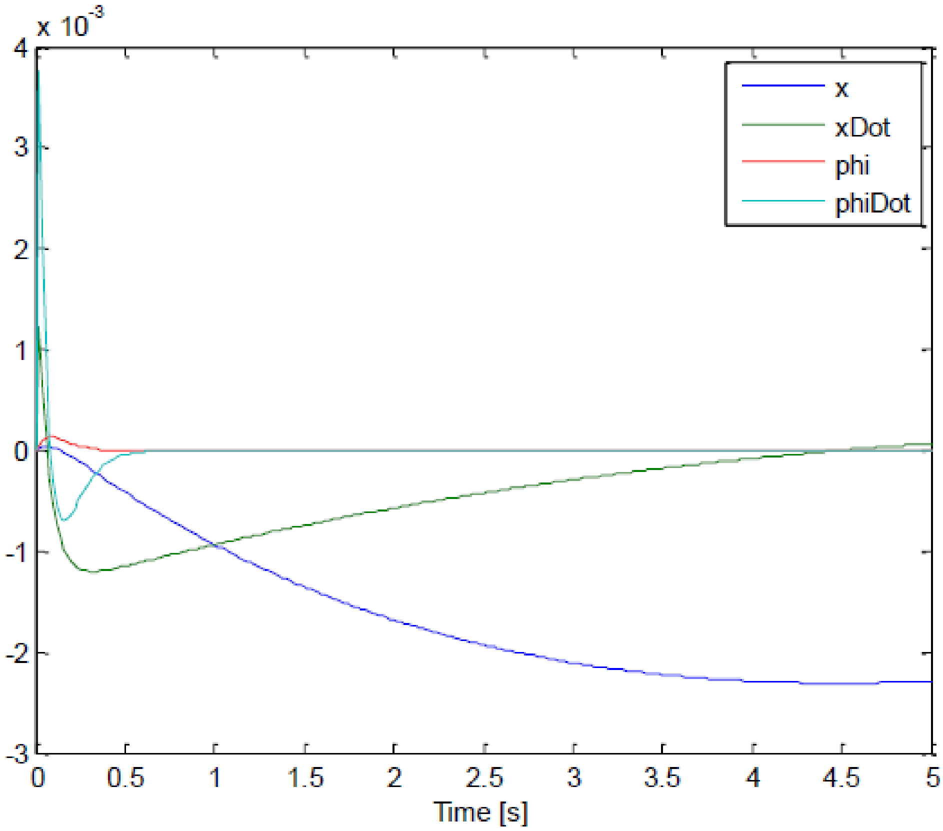

Figure 5.5: Response with high value of chassis's angle

Here fig 5.5 shows the response with high value of chassis's angle, here when we increasing weighting of chassis's angle then we have to compromise with response of the state and settling time.

5.1.2 Applying LQR controller on fractional order system

Here we have to applied LQR control algorithm on the fractional model which described in equation (4.46), here we have three different parameters for tuning, like Q and R are same as integer order system and one new parameter is α, in fractional model we can varies α between 0 to 1, here we have one more tuning parameter is α compare to integer model.

When applying LQR control algorithm on fractional order control system we get different response as shown below,

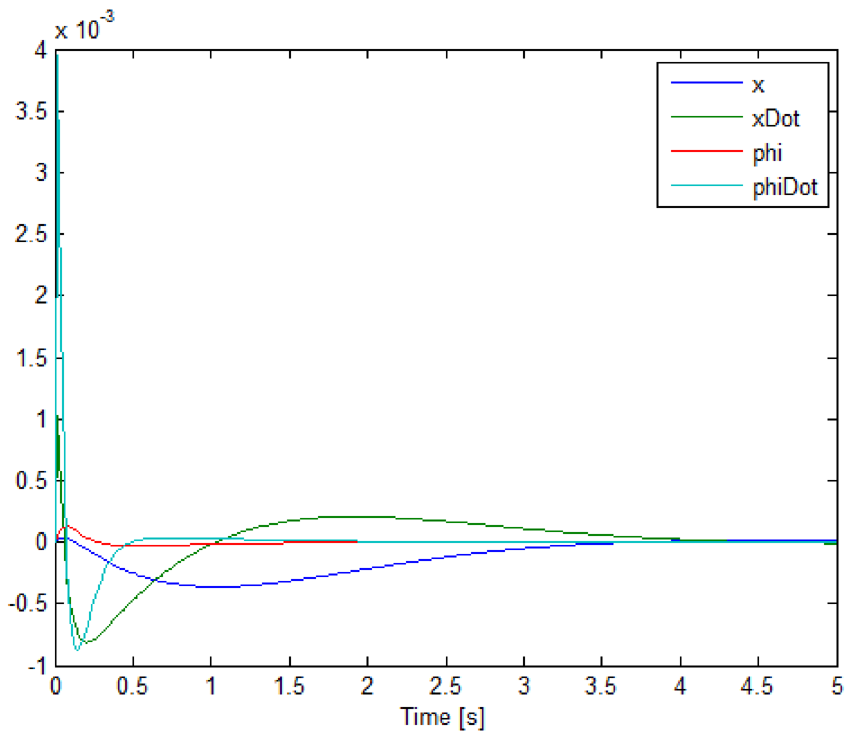

Figure 5.6: Response of the system when α is 0.9

Here fig 5.6 shows that here response is good but maximum peak overshoot and settling time is more

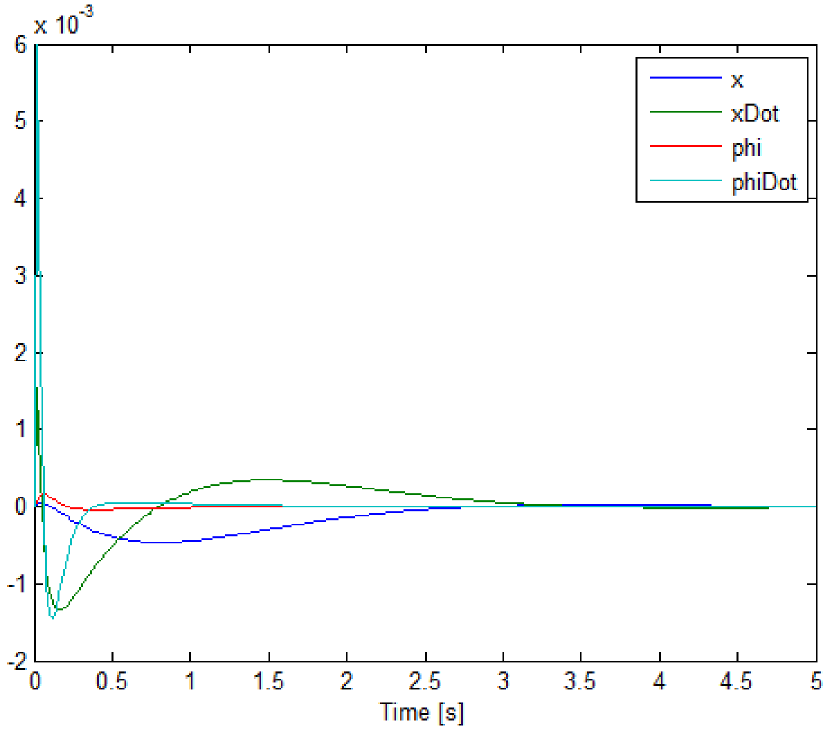

Figure 5.7: Response of the system when α is 0.5

Here fig 5.7 shows that here settling time is less compare to above response but peak overshoot is more

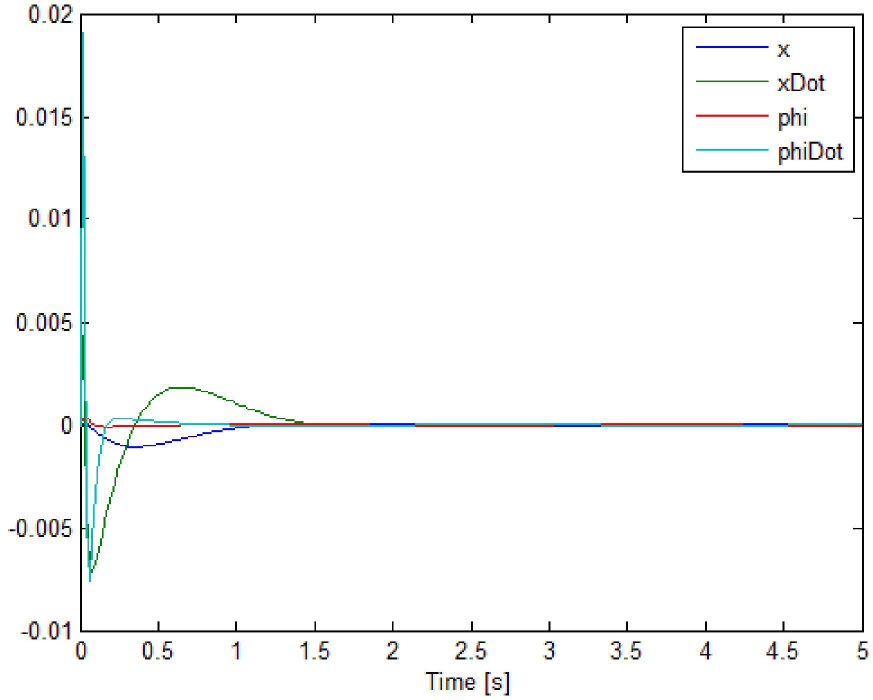

Figure 5.8: Response of the system when α is 0.1

Here fig 5.8 shows that system is more faster and less peak overshoot compare to above two response,it means when α is nearest to 0 then we get good response, but here when system become faster that time high torque is required for our self balancing system.

5.2 Apply PID and FOPID control algorithm

Here in this chapter PID and FOPID control algorithms are applied on the integer order and fractional order system and do the comparative analysis. For applying PID and FOPID control algorithm on the system before that we have to modelling system into the MATLAB simulink.

Integer model and fractional model are derived in the chapter 4, from the state space equation of the model we have to design system model into the MATLAB simulink,

Model of integer order system

Here these below two equations are for integer order, using these equations we have to modelling into the MATLAB,

$$\ddot{x} = 2.5783x + 2.9346\dot{x} - 122.6800\phi - 10.3701\dot{\phi} + 0.0815V_a$$

$$\ddot{\phi} = 7.7680x + 8.8416\dot{x} - 231.1195\phi - 31.24333\dot{\phi} + 0.2456V_a$$

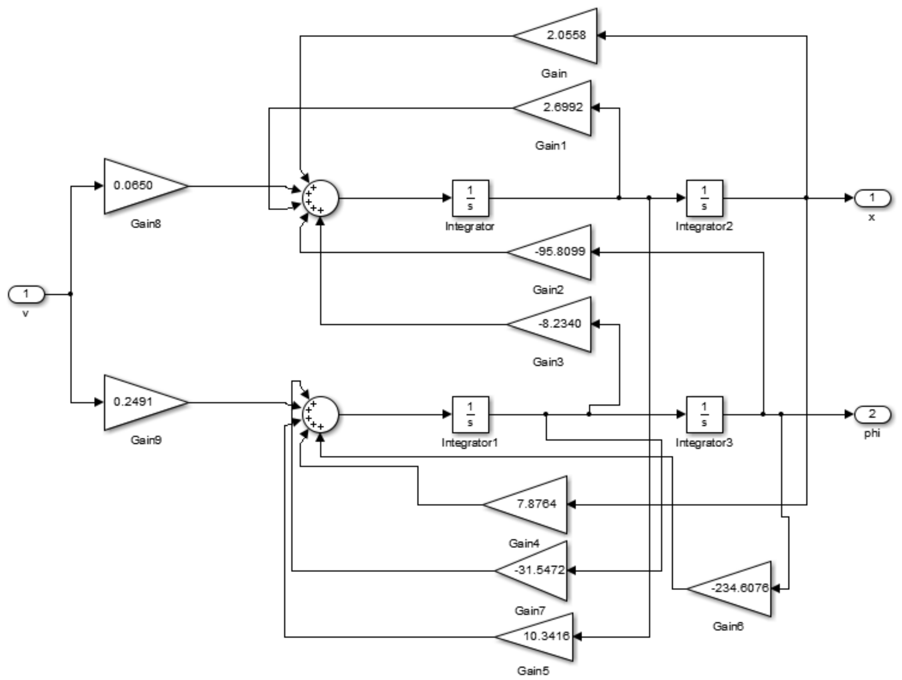

Figure 5.9: Integer order model

Model of fractional order system

Here these below two equations are for fractional order, using these equations we have to modelling into the MATLAB,

$$\ddot{x} = 3.6432x + 3.5945\dot{x} - 169.7749\phi - 10.9601\dot{\phi} + 0.1152V_a$$

$$\ddot{\phi} = 13.9620x + 13.7750\dot{x} - 415.8496\phi - 42.0024\dot{\phi} + 0.4415V_a$$

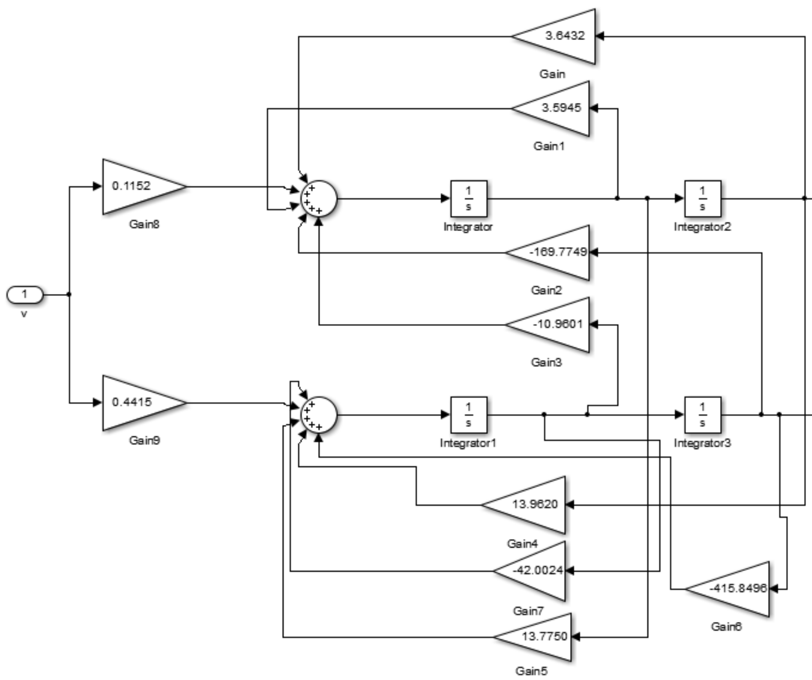

Figure 5.10: Fractional order model

Apply PID controller on integer order and FOPID controller on fractional order system

Here fig 5.11 shows the closed loop model of the self balancing system,and fig 5.12 shows the output response of the self balancing system, here yellow response is for the PID controller and purple response is for the fractional order system. As per the fig 5.12 we can say that the FOPID controller response is faster compare PID controller response, oscillation is less compare to PID controller, and also peak overshoot is less compare to PID controller. so as per the response we can say FOPID gives better response compare to PID controller.

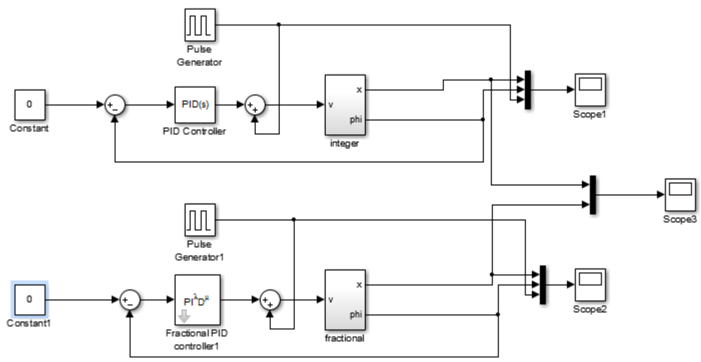

Figure 5.11: closed loop model

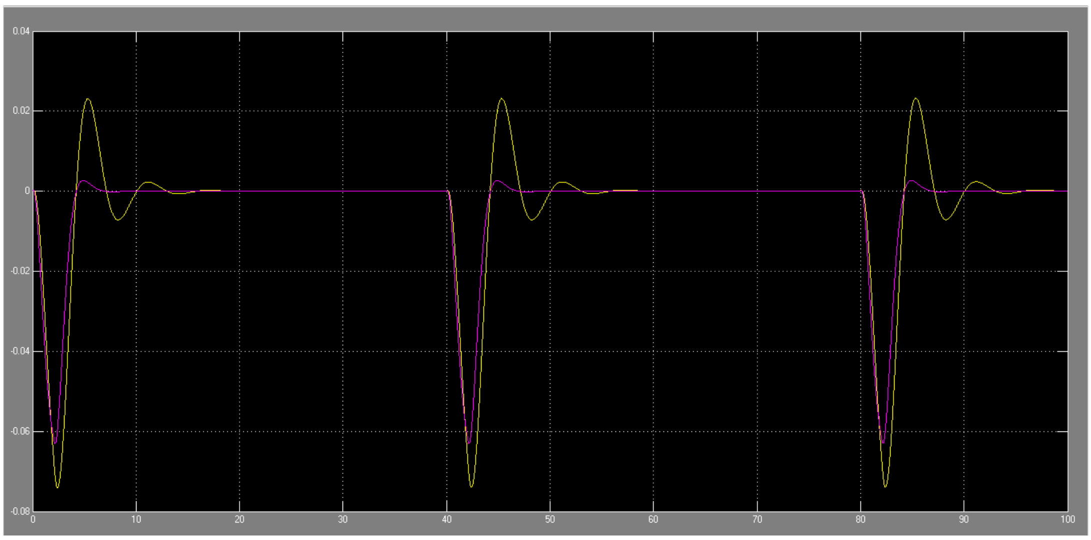

Figure 5.12: output response

Apply PID and FOPID controller on the fractional order system[3]

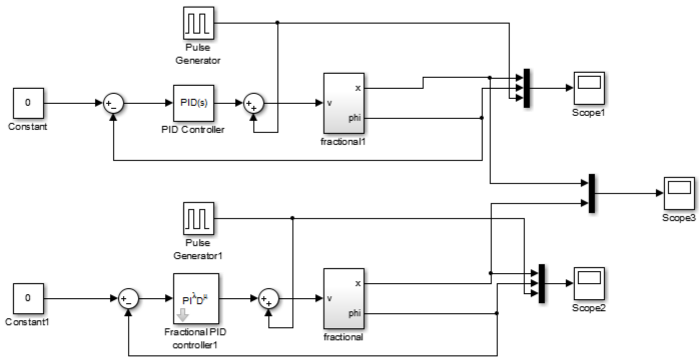

Figure 5.13: closed loop model

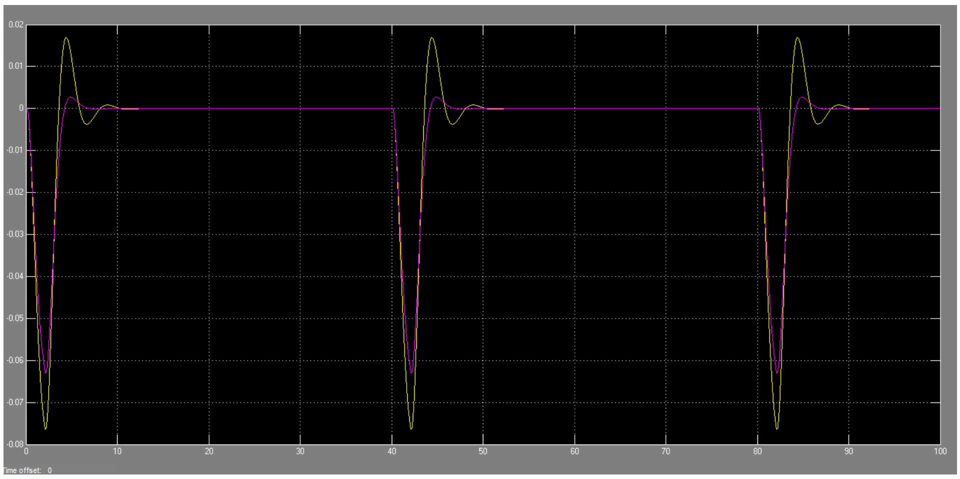

Figure 5.14: output response

Here we applied PID and FOPID control algorithm on the fractional order system and its closed loop is shown in fig 5.13, and fig 5.14 shows output response of the self balancing robot system, here we observe that PID and FOPID controller's settling time is same but peak overshoot of PID controller is more compare to FOPID controller.[4]

Apply PID and FOPID controller on the integer order system

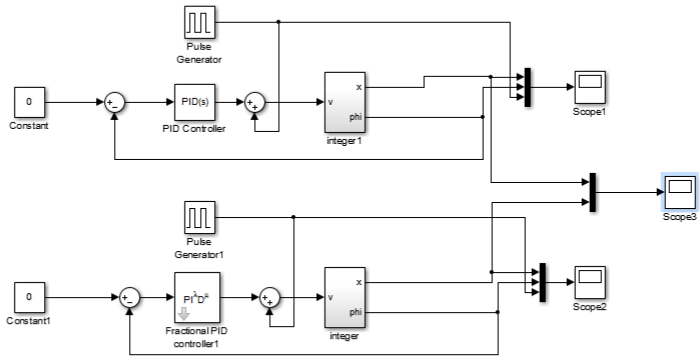

Figure 5.15: closed loop model

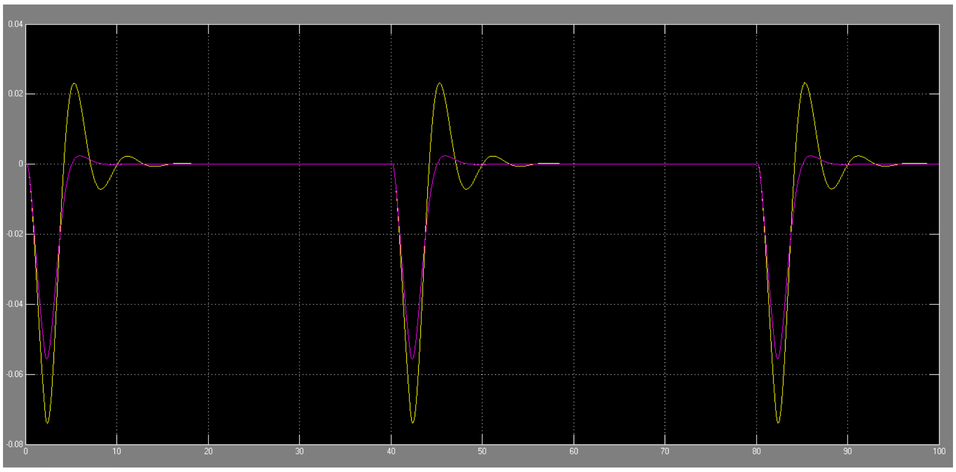

Figure 5.16: output response

Here we applied PID and FOPID control algorithm on the integer order system and its closed loop is shown in fig 5.15, and fig 5.16 shows the output response of the self balancing robot system, here we observe that settling time and peak overshoot of the FOPID controller is less compare to PID controller, and oscillation of the FOPID response is less compare to PID response.[7]

Chapter 6

Hardware implementation of self balancing system

Here in this chapter we have to implement PID and FOPID control algorithm on the hardware system of self balancing robot, and do the comparative analysis and try to understand which algorithm gives better response. Fig 6.1 shows the hardware model

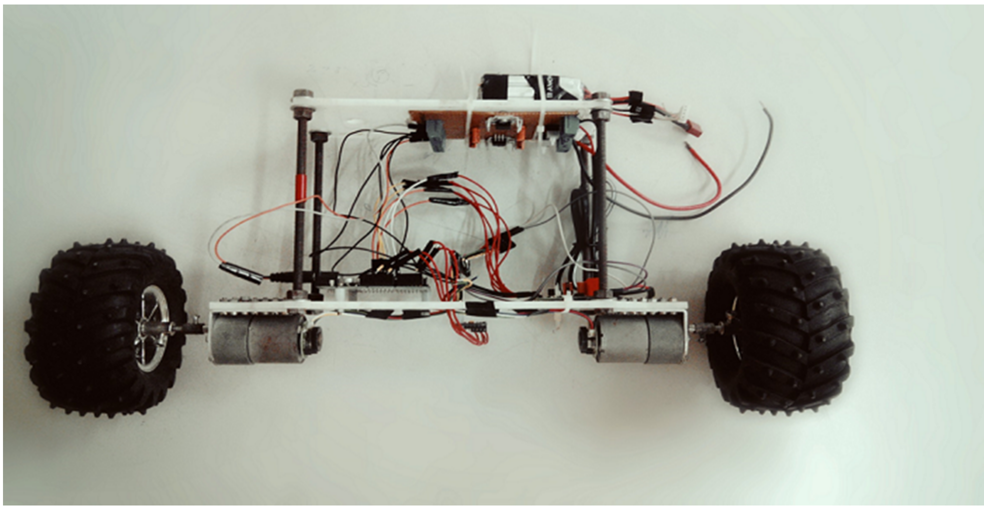

Figure 6.1: Hardware of self balancing robot

of the self balancing robot,this model is the group of different parts like MPU6050 gyro and acceleration sensor,20A motor driver circuit,12v two DC motor,12v power supply and Arduino controller,

6.1 Parts description

Here in this chapter different parts of the self balancing system has to be describe shortly, and these parts are MPU6050 gyro and acceleration sensor,20A motor driver circuit.

6.1.1 MPU6050 Sensor calibration

Here we used MPU6050 gyro and acceleration sensor which shows in fig,this is 8 pin sensor and it gives $acc_x, acc_y, acc_z, gyro_x, gyro_y, gyro_z$ and temperature value,

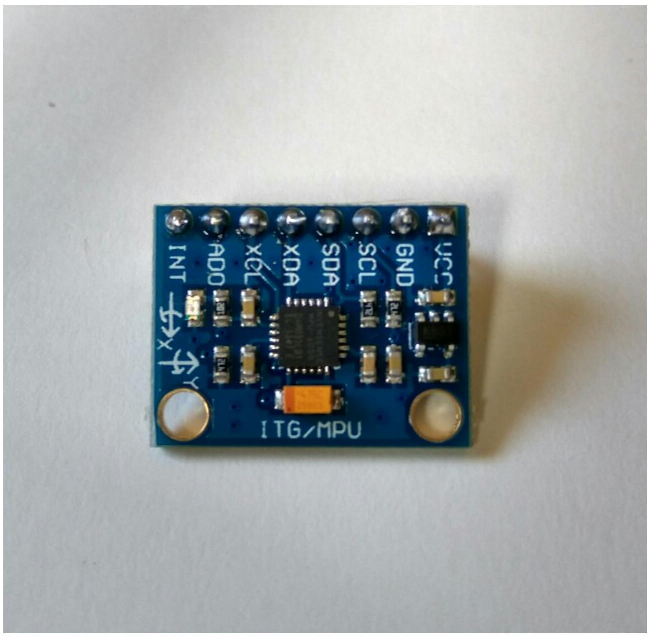

Figure 6.2: MPU6050 gyro sensor

These 8 pins are VCC,GND,SCL,SDA,XDA,XCL,AD0 and INT,we have to gives 3.3v to the pin VCC and GND is connect with the ground,SCL gives serial clock for I2C

interfacing, SDA is a serial data line for I2C interface and used to fetch the data stored into the internal FIFO register,XDA is used for external data interface for I2C bus, we can connect any external device with MPU6050 sensor,XCL gives the serial clock for externally connected device, I2C address is depend on the AD0 pin of the sensor,if AD0 connected to the ground the address is 0x68 and if it connected with Vlogic(3.3v) then address is 0x69 and INT pin is for the interupt.
Here the MPU6050 sensor output value is very noisy so for that we have to remove this noise and oscillation from the sensor output data, so for that complementary filter is the best way to remove the noise. here Fig 6.3 is shows the block diagram of the complementary filter

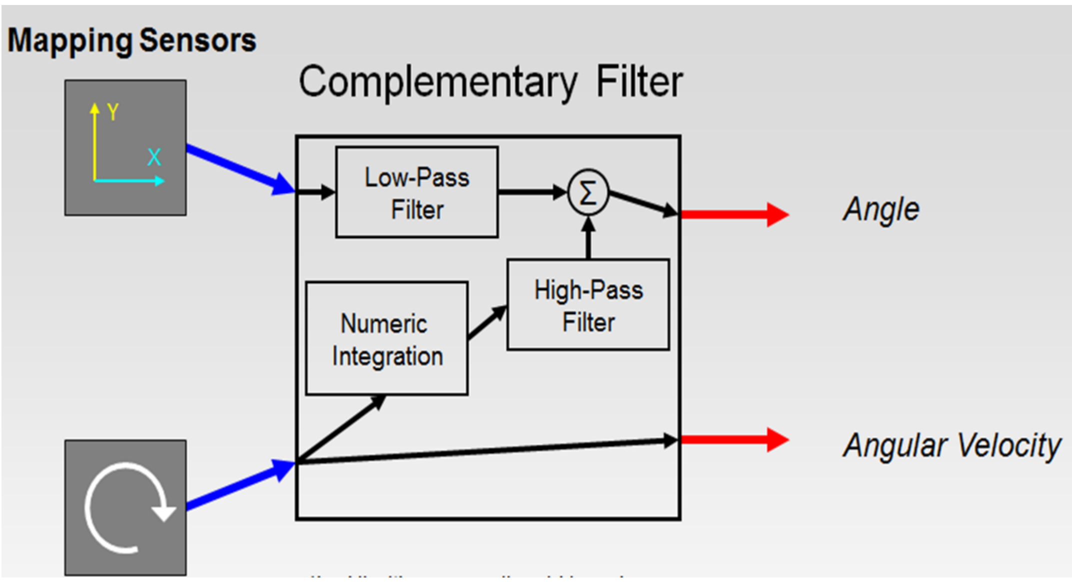

Figure 6.3: Block diagram of complementary filter

As per the block diagram of complementary filter we have to make the simulation model of the complementary filter, for that we have to implement these two equation which are shown below,

$$Pitch = tan^{-1}\frac{-accX}{accZ}$$

$$CompangleY = 0.93(CompangleY + gyrorate * dt) + 0.07 * Pitch$$

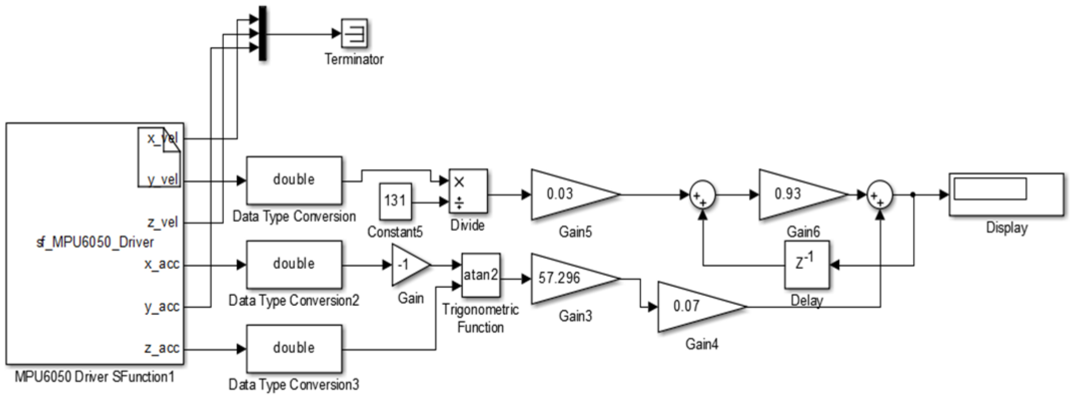

Figure 6.4: Simulink model of the complementary filter

Fig. 6.4 shows the simulation model of the complementary filter. when we apply the complementary filter and get response between gyro angle Y and compangleY which shown in Fig. 6.5

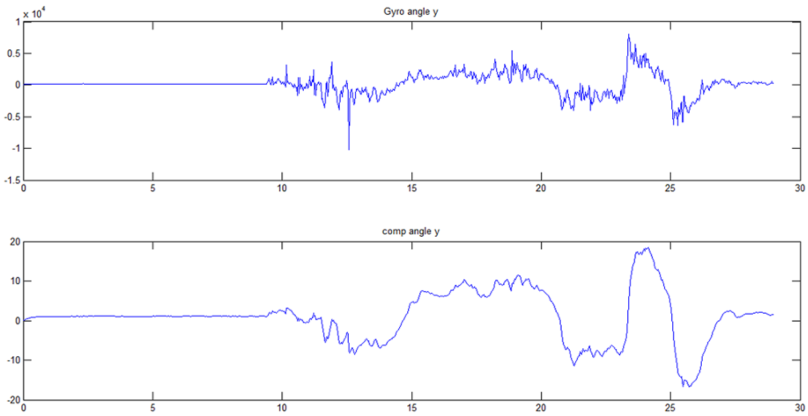

Figure 6.5: Response of the complementary filter

6.1.2 20A motor driver circuit

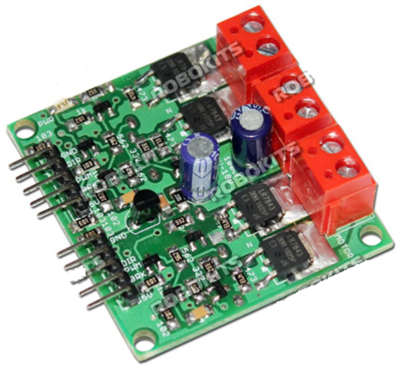

Figure 6.6: 20A motor driver circuit

Here in fig 6.6 shows the 20A motor driver circuit,it's working with 6 to 18 V and 20A current. This motor driver circuit is ideal for two motor where work with 20A current during normal operation and startup operation,it have breaking feature and we can work with the motor direction like forward and reverse, it has protection circuit for avoid any current and voltage fluctuation.

6.2 Connection diagram of hardware model

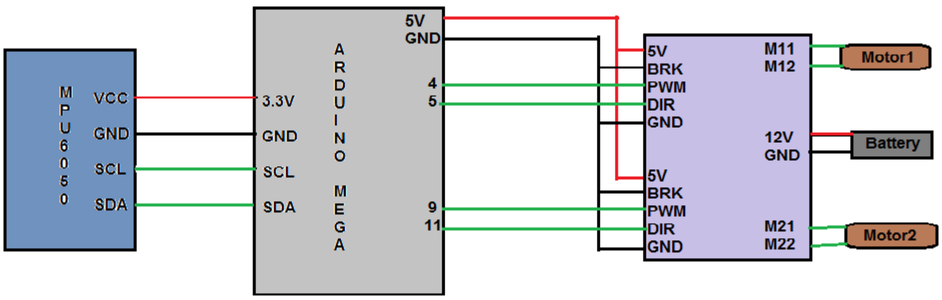

Figure 6.7: system connection diagram

Here in fig 6.7 shows the connection diagram of the hardware model, here MPU6050 gyro sensor is connected with the arduino mega 2560, and arduino controller connected with the 20A motor driver circuit, 12V two DC motor and battery is connected with the 20A motor driver circuit.

6.3 Implement PID and FOPID on hardware model

For applying PID and FOPID control algorithm on the hardware model we have to make a simulation model into the MATLAB,fig 6.8 shows the simulink model of the system,

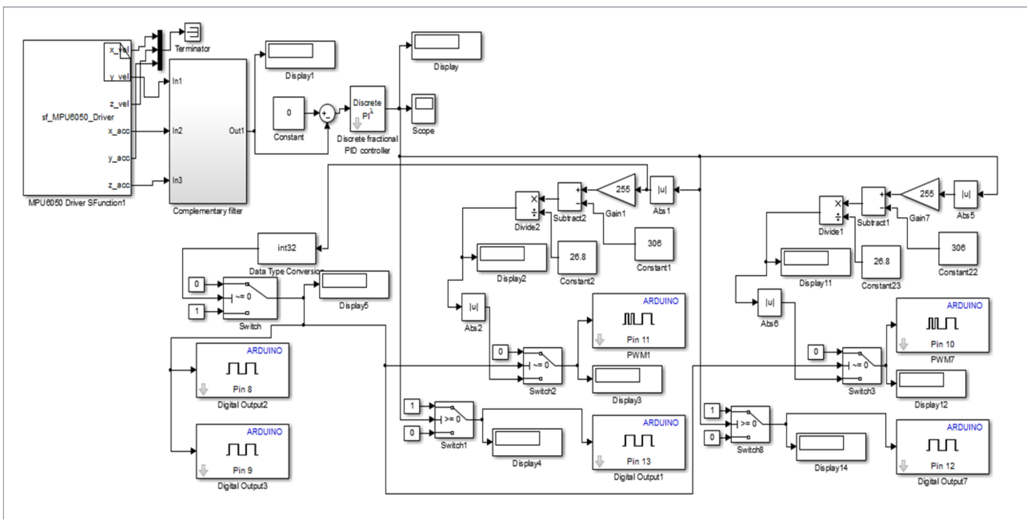

Figure 6.8: Simulink model of the system

Here we plot the response of the PID and FOPID control algorithm for different KP value,

From these PID response we can observe that when KP value is 4 that time response is good compare to other KP value and that time system can softly move.

From this FOPID response we can observe that when alpha value is samll like near to 0 that time response of the system is good and system become faster.

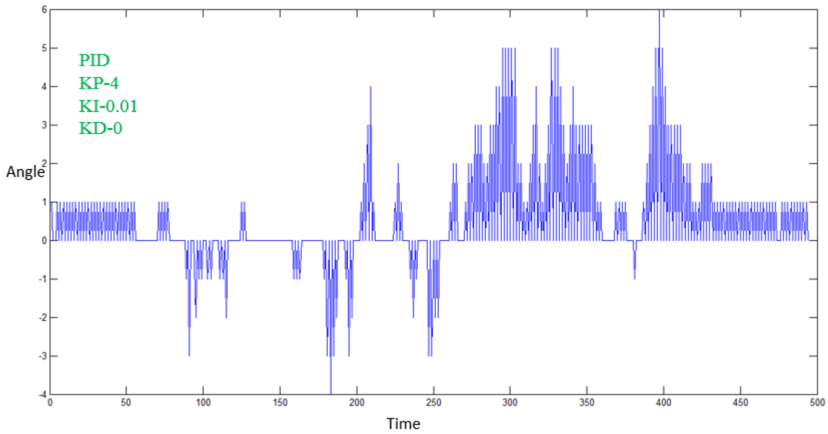

Figure 6.9: PID response for the KP=4,KI=0.01,KD=0

Fig. 6.15 shows the comparison graph between PID and FOPID control algorithm, and from this response we can observe that FOPID response is good.

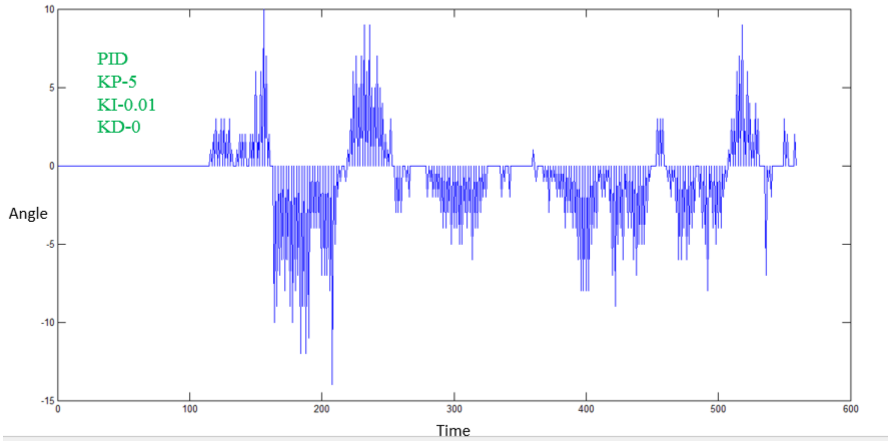

Figure 6.10: PID response for the KP=5,KI=0.01,KD=0

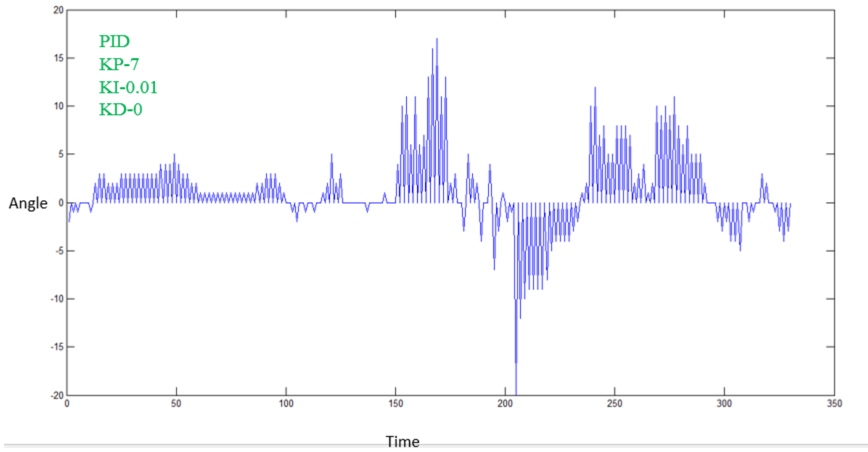

Figure 6.11: PID response for the KP=7,KI=0.01,KD=0

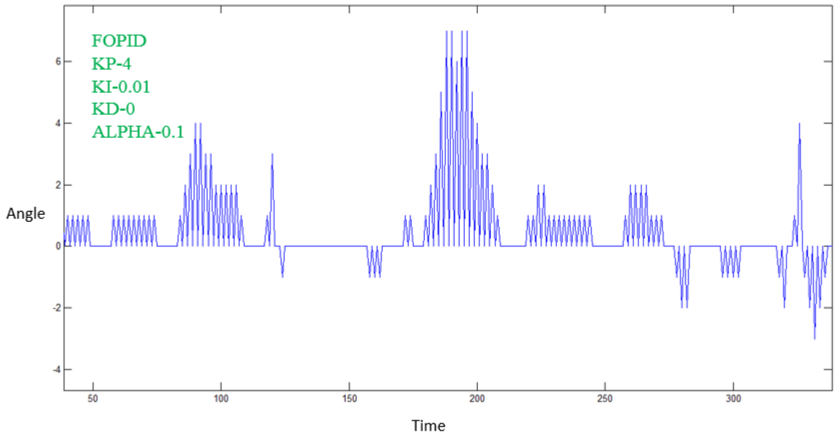

Figure 6.12: FOPID response for the KP=4,KI=0.01,KD=0,alpha=0.1

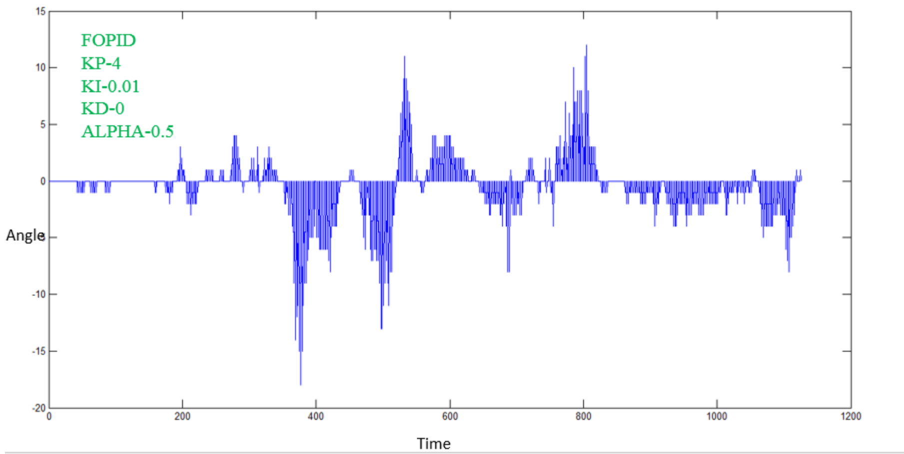

Figure 6.13: FOPID response for the KP=4,KI=0.01,KD=0,alpha=0.5

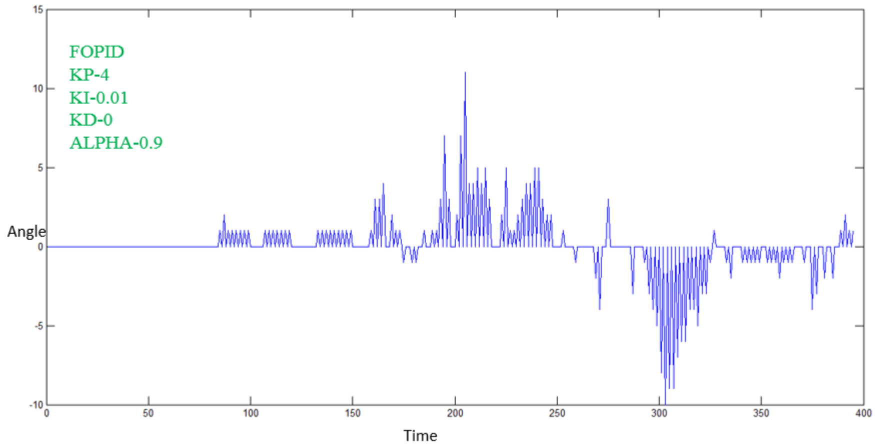

Figure 6.14: FOPID response for the KP=4,KI=0.01,KD=0,alpha=0.9

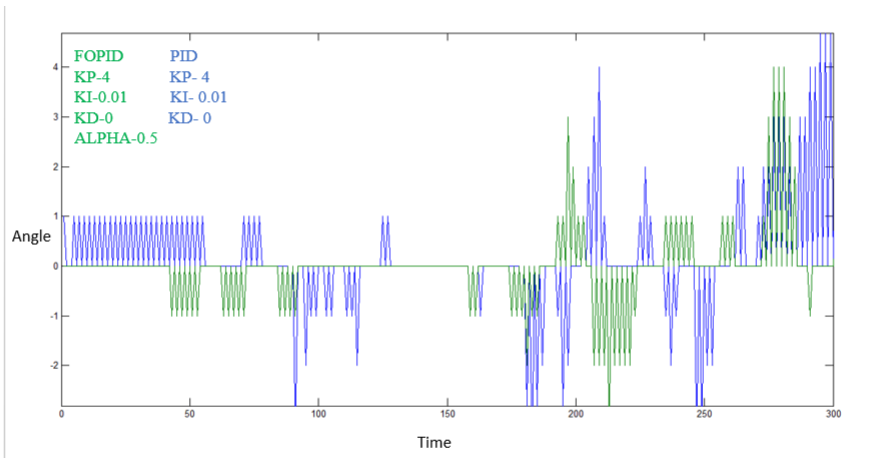

Figure 6.15: Comparison of PID and FOPID control response

Chapter 7

Conclusion and Future work

Conclusion

After the apply PID and FOPID control algorithm on the self balancing system we can conclude that the FOPID controller response is more better than PID control algorithm,FOPID have more stability, less oscillation and gives faster response compare to PID control algorithm.

Future work

Furthermore,Fractional optimal control can be implemented on aforementioned system in order to improvise system response.

References

[1] juri Belikov Aleksei Tepljakov, Eduard Petlenkov. Digital fractional-order of a position servo. *Department of Computer.*

[2] Billur Hugh F. Durrant-Whyte Barshan.

[3] A.Hernandez B.M Vinagre, I.Podlubny.

[4] Roy ABI ZEID DAOU. comparison between integer order and fractional order controllers. *IEEE Electronical Conference*, Vol. 9, No. 3, May 1994.

[5] E Komoriya, K. Oyama.

[6] Carnal C.L. A.T. Alouani Lahdhiri, T.

[7] dingyu Xue yangQuan Chen, blas manuel vingre.

[8] dingyu Xue Yangquan chen, Ivo. Fractional order control-a tutorial. *American Control Conference.*

[9] Ivo Petras YangQuan Chen and Dingyu Xue. Fractional order control - a tutorial.

[10] SHAN Liang ZHAO Yuanzheng. Application of fractional order controller in servo control system. *Chinese control conference*, 2004.

Index

20A motor driver circuit, 37

Apply PID and FOPID control algorithm, 27

Applying LQR controller on fractional order system, 25

Applying LQR controller on integer order system, 19

Block diagram, 5

Conclusion and Future work, 43

Connection diagram of hardware model, 37

Convert integer order to fractional order system, 16

Design different controller for self balancing system, 18

Hardware implementation of self balancing system, 33

Implement PID and FOPID on hardware model, 38

Linear Quadratic Regulator (LQR) control, 18

Mathematical model of self balancing robots chassis, 12

Mathematical model of two wheeled, 9

modelling of a DC motor , 7

Motivation, 4

MPU6050 Sensor calibration, 34

Objective , 5

Objective And Block diagram , 5

Parts description, 34

System modelliing, 7

system modelling , 7