*MINI PROJECT REPORT*

# SELF BALANCING ROBOT

**ECD334 B.Tech : MINI PROJECT**

*Submitted By*

**NIKHIL K J (PKD19EC040)**

**JISHA ALBERT (PKD19EC031)**

**JINEESH P (PKD19EC030)**

**JAISON JOSHI (PKD19EC029)**

*under the guidance of*

**Prof. Vishnuprasad K**

*Asst.Professor,Dept.of Electronics and Communication Engineering*

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**GOVERNMENT ENGINEERING COLLEGE SREEKRISHNAPURAM**

**AUGUST 2022**

# CERTIFICATE

*This is to certify that the report entitled* **SELF BALACING ROBOT** *submitted by* **NIKHIL KJ (PKD19EC040), JISHA ALBERT (PKD19EC031), JINEESH P (PKD19EC030) and JAISON JOSHI (PKD19EC029)***, to the Department of Electronics and Communication Engineering, Government Engineering College Sreekrishnapuram,Palakkad-678633, in fulfilment of the requirement for the award of B-Tech Degree in Electronics and Communication Engineering, is a bonafide record of the project carried out by them under our guidance and supervision.*

**Prof.Vishnuprasad K**

**Guide** **Coordinator**

**Dr.Bindu P**

**Head Of the Department**

Place: Sreekrishnapuram

Date:

# ACKNOWLEDGEMENT

First of all we thank the Almighty, for giving us the strength ,courage and knowledge to do this project successfully.

We express our heartfelt gratitude to the Principal Dr. P C Reghu Raj, Govt. Engineering College, Sreekrishnapuram, for his inspiration throughout the project.

We would also like to thank Dr. Bindu P , Head Of the Department, Electronics and Communication Engineering, Govt. Engineering College, Sreekrishnapuram for providing permission and availing all required facilities for undertaking the project in a systematic way.

We are extremely grateful to Prof Vishnuprasad K , Govt. Engineering College, Sreekrishnapuram, who guided us with kind, ordinal, valuable suggestions. We would like to appreciate the guidance given by Prof. Hariprasad B and Prof. Vishnuprasad K our project Coordinators, Department of Electronics and Communication Engineering, Govt. Engineering College, Sreekrishnapuram, for providing sincere guidance.

We would also like to express our sincere gratitude to all teaching and non-teaching staffs of Department of Electronics and Communication Engineering, Govt. Engineering College, Sreekrishnapuram, for the sincere directions imparted and the cooperation in connection with the project.

We are also thankful to our parents for the support given in connection with the project. Gratitude may be extended to all well-wishers and our friends who supported us for the project.

# ABSTRACT

A two-wheeled self-balancing robot is a classic example of inverted pendulum system; i.e., it is a naturally unstable system. In recent years, it has become a hot area of research due to vast possibilities of its use in various areas like Segway vehicles, better wheelchairs, efficient robots, etc. In this project, an attempt has been made to address the inherent instability property of two-wheeled self-balancing robot using LQR control system. The goal is to construct a low-cost prototype which can successfully balance itself, and can be used to practically analyze the responses and stability of the system for different variables. Keywords: Segway, LQR Controller, Arduino

# ORGANISATION OF REPORT

The main body of the report is preceeded by detailed contents including abstract. This is followed by system description.

- Chapter 1 gives a brief introduction about the aim of the paper.

- Chapter 2 provides the literature review.

- Chapter 3 gives the details of requirements.

- Chapter 4 gives the details about construction of the robot and its working

- Chapter 5 gives the details results and disscusion about challenges we faced and future scope

- Chapter 6 is about the conclusion

The main report is then followed by references.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# INTRODUCTION

Two wheeled self balancing Robot is a foremost resarch topic in the area of robotics and control engineering.The principal of operation behind a self balancing robot is an inverted pendulum cocept.The major focus on this project is the hardware development of two wheeled self balancing robot for an application to carry object from one place to another.The modelling of self balancing robot is done in terms of inverted pendulum.Since the two wheeled self balancing robot has some inherent characteristics like instability,non linearity type of controller like LQR is used.A LQR controller is incorporated in the hardware section to improve and ensure a good performance for the system

## 1.1 AIM

To develop a self balancing robot

## 1.2 MOTIVATION

In today's era, whenever was talk about technological advancements and the resulted automation in our lives, the first thing that comes in our minds is the recent developments in field of electronics and the better, faster, efficient and smart machines and robots it has created. We desire for machines with smart thinking,self-sustaining and decision making capabilities. We want better robots which can automate the entire work chain in a company, which can replace a human workforce. Such is the desire for optimized systems that today we prefer wheeled robot over two-legged humanoid robot for having faster navigation capability, and among that we prefer two-wheeled robot over three/four-wheeled robot for having compact design and faster movement speed. Also, since the introduction of commercial Segway vehicles, two-wheeled selfbalancing robot has become a popular area of interest for researchers for its vast possibilities to be used for Means of accident free, environment friendly local transportation. Self-balancing wheelchairs for disabled patients. Compact and efficient robots to be used in hotels, warehouses, etc. And many more areas of application. All these offered challenges and the vast area of possible applications by a twowheeled self-balancing robot were the major driving force behind we were choosing this project.

# Chapter 2

# LITERATURE SURVEY

## 2.1 Mathematical Modelling

The principle of balancing a vertically standing robot with two wheels placed parallel to each other is llustrated in figure 2.1. The system is represented with one upward-standing inhomogeneous pendulum connected to the wheels on an axis. Firstly a tilt $\theta bfromitsuprightequilibrium, causedbytheforceofgravity, isdetected$
$torqueintheoppositedirection, largerthanthetorquecausedbythegravitationalforce, hast$

The robot platform can be divided into two parts:

A) The Main Body, which consist of the body and the two stators of the electric motor.

B) The Wheel Assembly, which consist of the wheel and the rotor of the electric motor.

The Wheel Assemblies and Main Body in Figure 2.2 are linked together by the magnetic forces generated between the rotor and stator of the electric motors, and also the mechanical forces on the bearings within the motors and gearbox.

Now we can apply Newtonian mechanics on the Main Body and Wheel Assembly separately, using the concept of Free-Body Diagram(Figure 2.3).and finally arrive at the equations for position and tilt angle.

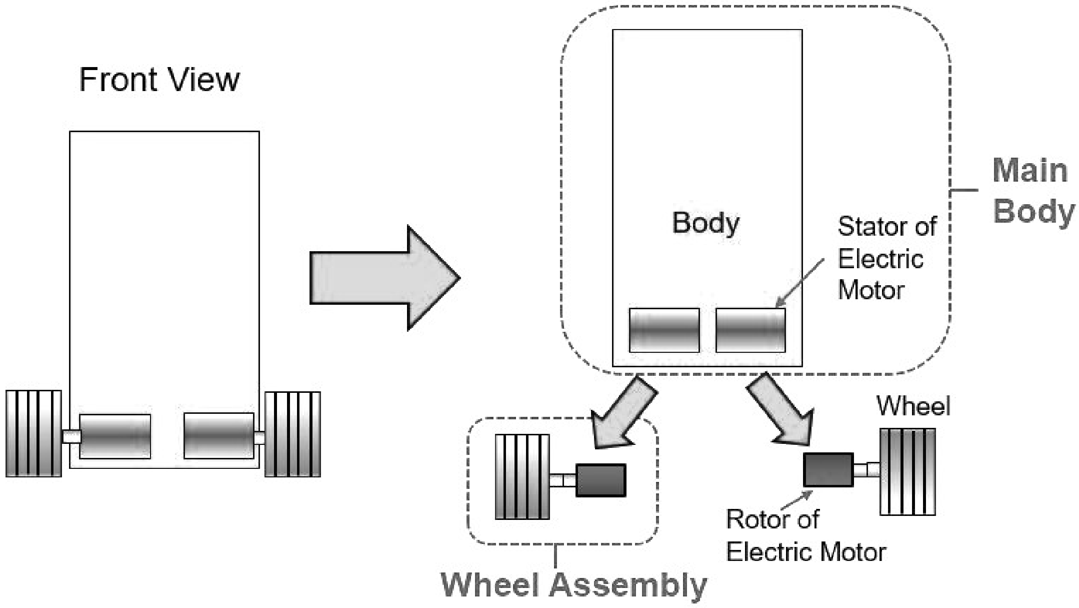



**Figure 2.2:** Dividing the TWSB robot platform into the Main Body and Wheel Assemblies

$$\ddot{x} = \left[m_b + 2\left(m_w + \frac{J_w}{R^2}\right) - \frac{(m_b l\cos\theta_b)^2}{m_b l^2 + J_b}\right]^{-1} \left\{ \begin{array}{l} (m_b l \sin\theta_b)\dot{\theta}_b^{\,2} - \frac{(m_b l)^2 g\cos\theta_b\sin\theta_b}{m_b l^2+J_b} + \\ 2\left(\frac{1}{R} + \frac{m_b l\cos\theta_b}{m_b l^2+J_b}\right)T_w \end{array} \right\}$$

$$\ddot{\theta}_b = \left[J_b + m_b l^2 - \frac{(m_b l\cos\theta_b)^2}{m_b + 2\left(m_w + \frac{J_w}{R^2}\right)}\right]^{-1} \left\{ \begin{array}{l} (m_b l\sin\theta_b)\left[g + \frac{m_b l\cos\theta_b\,(\dot{\theta}_b)^2}{m_b + 2\left(m_w + \frac{J_w}{R^2}\right)}\right] - \\ 2\left[R + \frac{m_b l\cos\theta_b}{m_b + 2\left(m_w + \frac{J_w}{R^2}\right)}\right]\frac{T_w}{R} \end{array} \right\}$$

Since we are using stepper motor, this equations for position and tilt angle can simplified further[6] to get the state space model of the system.

$$s\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{C_2C_3 - C_1C_4}{C_2} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -sR\frac{C_4}{C_2} \end{bmatrix}(s\theta_w)$$

$$x_1 = \theta_b \qquad x_2 = s\theta_b$$

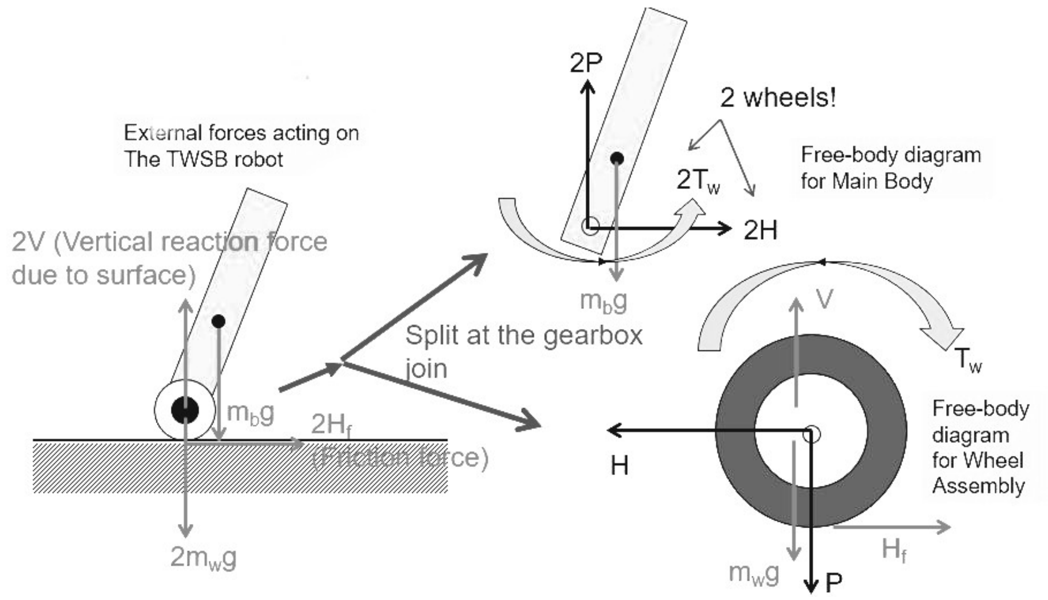



**Figure 2.3:** Free-Body Diagram

## 2.2 Controllers

Balancing of robot requires application of closed-loop (Feedback) control systems. Researchers have used different kinds of control methods to study balancing of robots. Some of these controllers/methods are as follows:

- LQR (Linear Quadratic Regulator)
- PID (Proportional, Integral, Derivative)
- LQG (Linear Quadratic Gaussian)
- FSF (Full State Feedback), or, Pole Placement
- Adaptive control
- SMC (Sliding Mode Control)
- Intelligent controls like NN (Neural Network) and Fuzzy logic

Each of these methods have their own advantages and shortcomings in terms of precision, responsiveness and reliability. So, researchers try to combine two or more control systems so that they can compensate each other's

disadvantages and we can get a better performance. [4] derived mathematical models for PID and LQR control systems, simulated them in MATLAB, and concluded that LQR had better balancing performance than PID.
[5] developed a mathematical model with PID and LQR control systems, simulated them in MATLAB and Simulink, and concluded that PID controller is sufficient if robot has to only balance itself vertically and not move around; but if we want a robot which can balance itself vertically as well as move around, then we better go for LQR for best results.

## 2.3 LQR controller

Stability of a system is determined by the placement of the poles, they have to be carefully chosen. For a stable system the poles must be in the left half-plane.If they are too far away the input voltage u will be too high. The poles' imaginary part should not be too large relative to the real part, otherwise the system might oscillate.There are few ways of placing poles but firstly the poles need to be chosen. One method is to choose some arbitrary poles in the left half-plane and see if it yields the desired rise time and overshoot. This method might require a lot of guessing before obtaining a desirable result. Another method is called Linear-Quadratic Regulator which is based on the theory of optimal control of a dynamic system. This is the method of choice for this project since it is more systematic.To control a linear system, LQR-control can be applied.

The states of the dynamical system are described by a state space model including the matrices A, B and C as seen in Figure 2.4.The systems poles can placed with the help of the gain K. In LQR the poles are chosen by

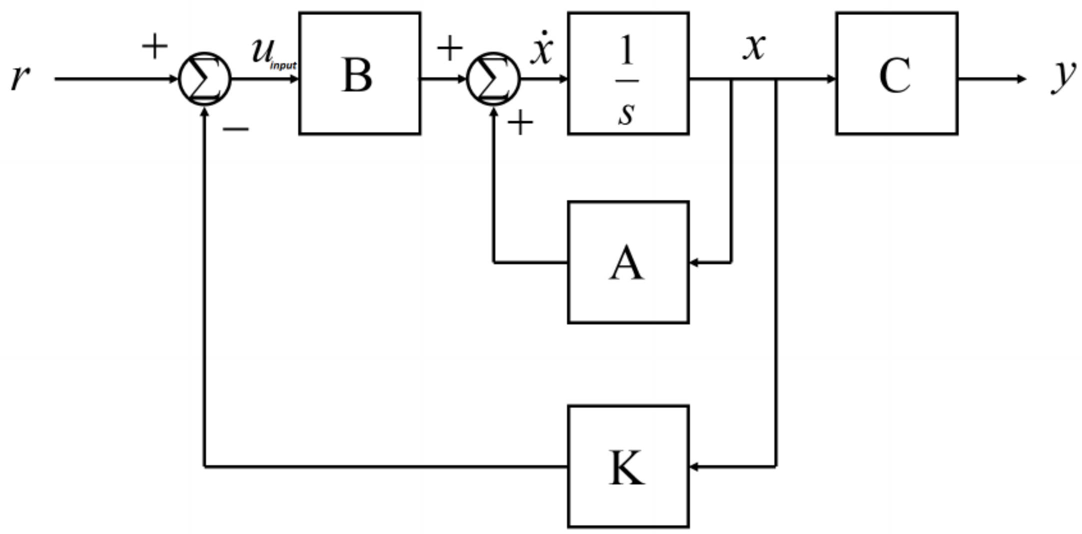



**Figure 2.4.** Block representation of state space system showing the system matrices $A$, $B$ and $C$ as well as the gain matrix $K$.

minimizing the cost function defined as

$$\mathbf{J} = \int_0^\infty x^T Q x + u^2 \, dt$$

where the Q matrix define the weighted importance of each state in x . It can be shown that the optimal control law is

$$u = -Kx$$

where K is

$$K = R^{-1} B^T P$$

And P is the solution to the Ricatti equation:

$$Q + ATP + PA - PBB^T P = 0$$

Solution to this Equation and the value of the **K** can be easily calculated

using the matlab function **lqr()**.
The values of Q and R are choosen using Trial and Error method. For example,if Q is large enough then LQR will focus only on reaching x = 0. In order to achieve the desired results the Q matrix should be chosen while iteratively simulating the entire system. The simulations with different Q and R would give a better understanding of how they affects the system's performance.

## 2.4 Complementary Filter

We have two measurements of the angle from two different sources (Acclerometer and Gyroscope). The measurement from accelerometer gets affected by sudden horizontal movements and the measurement from gyroscope gradually drifts away from actual value. In other words, the accelerometer reading gets affected by short duration signals and the gyroscope reading by long duration signals. These readings are, in a way, complementary to each other. Combine them both using a Complementary Filter and we get a stable, accurate measurement of the angle.[7]The complementary filter is essentially a high pass filter acting on the gyroscope and a low pass filter acting on the accelerometer to filter out the drift and noise from the measurement.

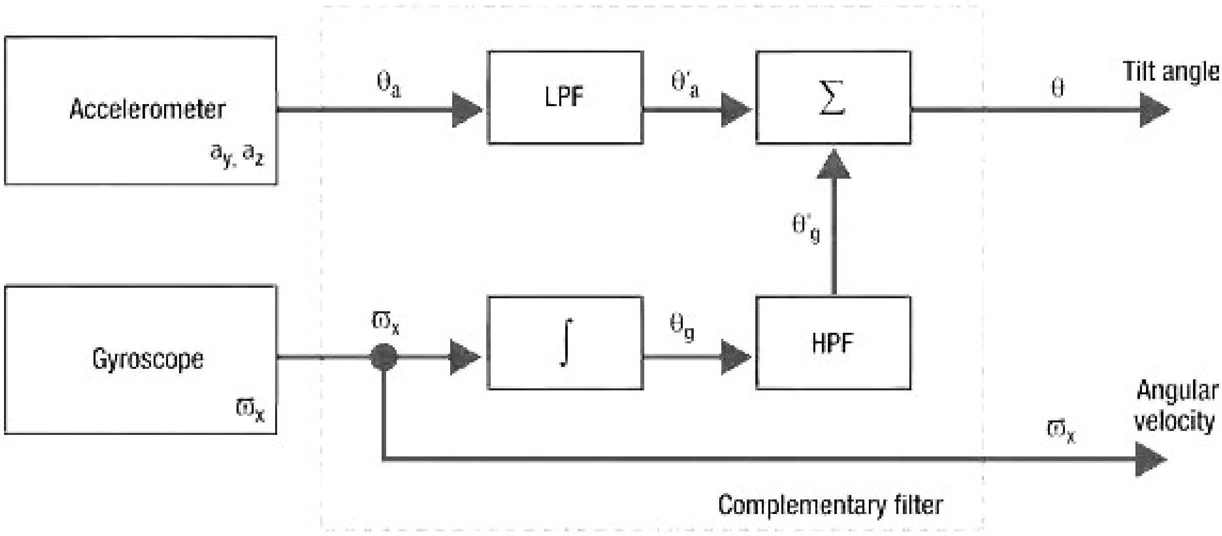



**Figure 2.4:** Complementar Filter

## Chapter 3

# Overview of the Proposed System

### 3.1 Project Conceptualisation

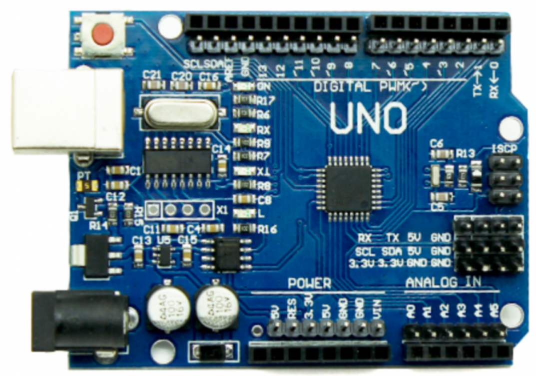
The complete project has divided into different steps and the first step is the project conceptualisation. Here the basic idea of working of the projects and theoretical concepts are referred and solid understanding about the projet is formulated. Next step is designing and fabrication of robot hardware by selecting all the necessary electronic components. This process includes building complete frame of the robot and embedding components over it with care by keeping its center of mass in mind.This step is followed by software algorithm implementation and its integration with the hardware.

### 3.2 Hardware Development

The hardware system designing is one of the most important phase in development of the project. There are many constraints that must be considered while developing the hardware system. As the first step 3d modelling and design is done with the help of suitable softwares. As per the design, the structure of the body is fabricated. The selection of materials for building the body is an important factor in the hardware development. We used transparent acrylic sheet for bulding the body and the structure is cut out using laser cutting. We have selected appropriate well gripped wheels and used stepper motors since it provides constant torque. Then circuit is designed and components are placed as per the modelled design.

## 3.3 Hardware Components

### 3.3.1 Aurdino UNO




The main microcontroller used to balance the robot is Arduino UNO. It can be considered as the brain of the SBR. There are many microcontrollers like PIC series, AVR series, 8051, raspberry, etc. available in the market, but We have selected the Arduino because it is comes with a complete package which includes 5V regulator, an oscillator, a microcontroller, serial communication interface, LED, a burner and headers for the connection, which makes it very easy in use and very simple to assemble components with it. It is an open source prototyping platform i.e. the board can be easily modified and optimized for better functionality. It easily scales between different members of the family. Its software has many available libraries and functions which makes it easy to do its coding even for beginners. Here it shows an Arduino UNO. The Arduino UNO is mainly based on AVR microcontroller Atmega328 and it comes with 6 analog input pins, USB interface and 14 I/O digital ports out of which 6 pins can be used for PWM output. These I/O digital pins can be used to connect it with external electronics circuits. In the SBR it is connected with the IMU sensor to get the tilt angle information and after processing the data it communicates with the motor drivers with the instructions of speed and direction of the motor adjustments.

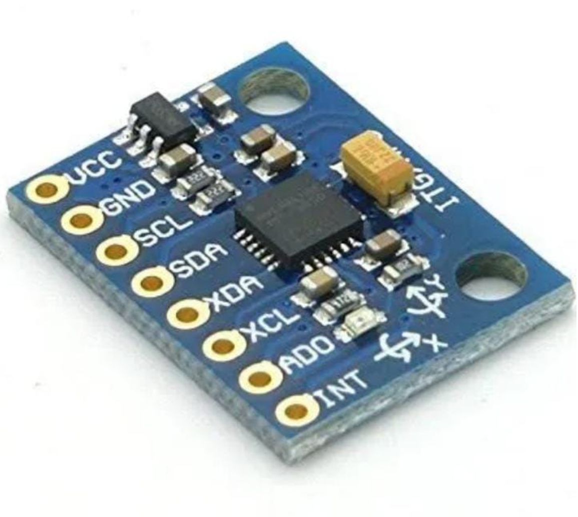
### 3.3.2 Stepper Motor


This is the view of stepper motor which is used as an actuator in the self-balancing robot. There are different motors like DC geared motor, servo motor, etc. are available for use in the market but We preferred to use stepper motor in the project because it is a better choice whenever we need any controlled movement. It is precise and has no any performance loss when the battery voltage drops. There is no lag in performance due to moment of inertia of the motor. Stepper motor is widely used in robotics and in other fields where precise and accurate movement is needed. The motor's rotation has a direct relationship with the applied input pulses. We have used hybrid bipolar stepper motor . This motor has 1.8o step angle which means 200 steps per revolution. It has two coils and there are two wires connected with each coil. It needs a motor driver module to control it. The use of these stepper motors is the main difference with respect to the previous design based on DC motors. In the new version, bipolar stepper motors are being utilized. This motor is a brushless DC motor in which the complete turn is divided into several steps. Working in the correct operating range, they allow to maintain the position without using encoders

### 3.3.3 Wheels

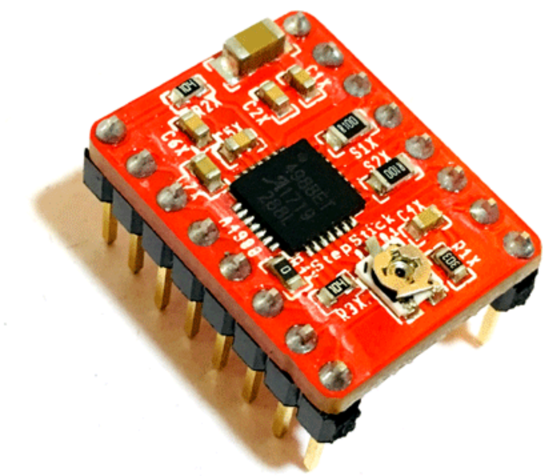
It is a view of wheel used in the robot. The selection of wheels is the important part in development of a proper balanced robot. The wheels should have a good grip over a floor which we are using so that it can efficiently stand with the weight of robot and can reduce the chance of falling down of the robot. I have used two wheels of 6.5cm diameter (with tyre its diameter is 7cm). There is a rubber tyre on the outer circumference of wheel which helps it to get good grip. It was preferred to use moderate size wheel in order to get satisfactory center of mass of the robot and also to properly notice the robot's balancing movement. It is very important that wheels should not skit the floor, otherwise it will be very hard to reach balanced condition of the robot.

### 3.3.4 IMU Sensor




This is an electronic device that measures the tilt angle and the tilt angular speed using a combination of accelerometers and gyroscopes. The communication between this sensor and the microcontroller is done by I2C

### 3.3.5 A4988 - Motor Driver



Working in the correct operating range, they allow to maintain the position without using encoders. The speed of the motor can be controlled by means of a PWM signal y the direction with a digital signal using the microstepping motor driver A4988

## 3.4 Software

### 3.4.1 Aurdino IDE

Arduino IDE (Integrated Development Environment) is an official Arduino software which is mainly used for writing, compiling and uploading the sketch or code into the Arduino modules. It is an open source software and easily available for various operating systems like Windows, MAC, Linux. It runs on the Java platform which comes with many inbuilt commands and functions that play an important role in editing, compiling and debugging the code in the environment. Arduino IDE supports both C and C++ languages and the main code written in this software is known as a sketch. IDE generates a HEX File from the sketch and transfer and upload it into the controller of the Arduino board. The IDE mainly consists of two basic parts: Editor which is used for writing the required code and Compiler which is used for compiling and uploading the sketch into the given Arduino device.

### 3.4.2 MATLAB

Matlab is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.
Although MATLAB is intended primarily for numeric computing, an optional toolbox uses the Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.Here MATLAB is used to calculate K matrix required for the LQR controller

### 3.4.3 3D Designing and simulating tools

**SketchUp**

For developing a 3D design of self balancing robot we used a software known as sketchup. SketchUp is an intuitive 3D modeling application that lets you create and edit 2D and 3D models with a patented “Push and Pull” method. We created basic designs in SketchUp and evaluated each one and selected the best approach.

**Fusion 360**

Fusion 360 is a software used to design and create prototypes from scratch, and finalize models to finished products. We have used Fusion 360 to simulate the robot and find its properties such as center of mass, moment of inertia etc. which were essential for the modelling of the robot.

**CorelDraw**

CorelDRAW is a vector graphic editor software and we have used it for designing the base and side planes with accurate dimensions for the purpose of laser cutting the platforms.

# Chapter 4

# IMPLEMENTATION

This chapter discusses about the construction of the Self balancing robot and the steps involved in its fabrication and working.

## 4.1 3D modelling

The first step in the construction process is designing the robot. As per the constraints and requiremnts from the mathematical modelling of the self balanced robot, a basic 3D desgin of the robot is prepared. We have used a software called SketchUp for this purpose. Figure 4.1 shows the basic structure of the body designed in SketchUp.

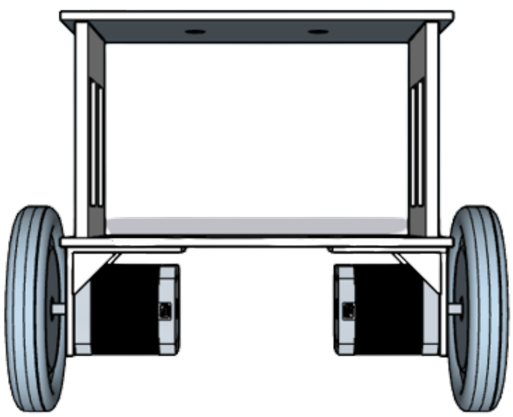


**Figure 4.1:** Body structure

The body is designed in such a way that it has two layers as shown in the figure. Each layer is of dimensions 18x10x0.4 cm. The top and bottom layer is separated by a distance of 10 cm. Side walls are of dimensions 10x8.60x0.6 cm.

## 4.2 Structure Building

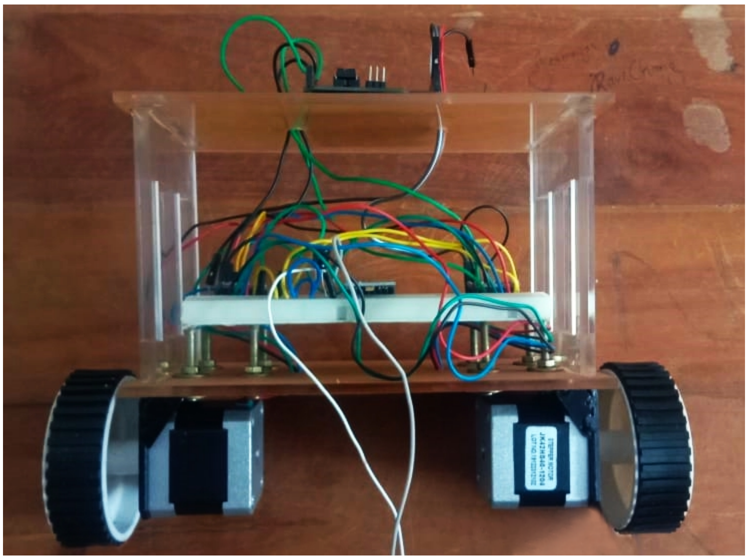


**Figure 4.2:** Sel balancing robot - structure

The main point behind selection of the frame of the robot was its availability, versatility and sufficient strength to handle all components weight used . Other frames like plywood, cardboard, etc. are not sufficient to handle the weight of all components used here and also there can be seen some bends or change in size of holes when it falls while doing some trial and error experiments on the robot. So, acrylic sheet is the better option than other frames. We have used transparent acrylic sheet . Other frames like plywood, cardboard, etc. are not sufficient to handle the weight of all components used here and also there can be seen some bends or change in size of holes when it falls while doing some trial and error experiments on the robot. So, acrylic sheet is the better option than other frames.The advantage of using acrylic sheet is it is easy to cut, drill, install and very cost effective due to having less chance of break. The base and side platforms of the robot are fabricated using acryilic sheet as per the dimenions by laser cutting. Then stepper motors are mounted and other components are placed as per the design. Figure 4.2 shows the complete structre of the robot after hardware assembling.

## 4.3 Circuit Diagram

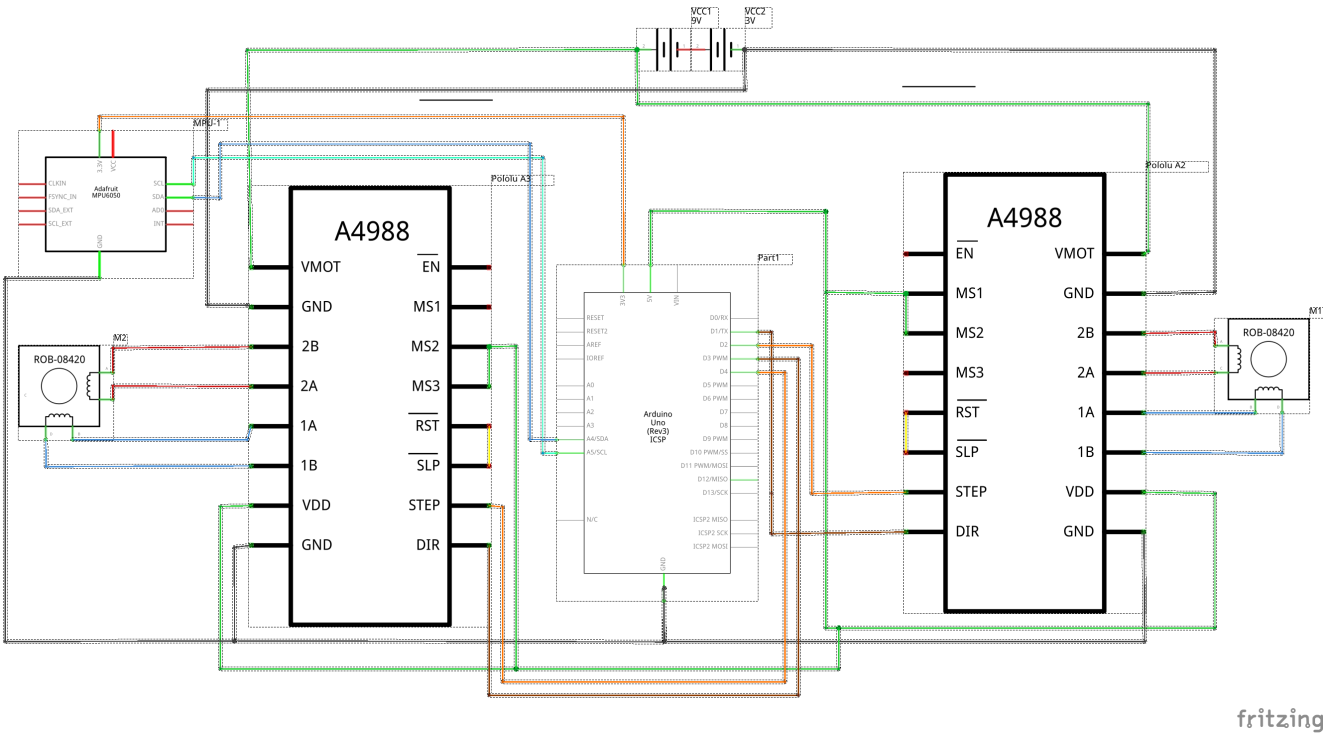



**Figure 4.3:** Circuit Diagram

A 12V power supply is given here for the working of the Stepper Motors.The I2C communication serial data lines SCL and SDA of IMU sensor is connected with the I2C communication serial pins A4 and A5. Digital output pins 2, 3, 4 and 5 of the Arduino UNO is joined with the direction and step control pins of the stepper controllers. A1, A2, B1 and B2 pins of stepper motor controller is wired with the stepper motors to give it power supply and instruction of movement. VMOT and GND pins of motor controller are directly connected with the the power supply. M1 and M2 pins of the A4988 is kept high in order to achieve 1/8th step control of the motor. And finally the motors are connected with the wheels for the movement in order to balancing the robot. Figure 4.3 shows these hardware connections between components of robot.

### 4.3.1 A4988 Current limiting

We needed to limit the maximum amount of current flowing through the stepper coils and prevent it from exceeding the rated current of the motor(1.2A).To set the current limit, a small trimmer potentiometer is provided on the A4988 driver.The current limit is determined by measuring the current running through the coil as shown in Figure 4.5.

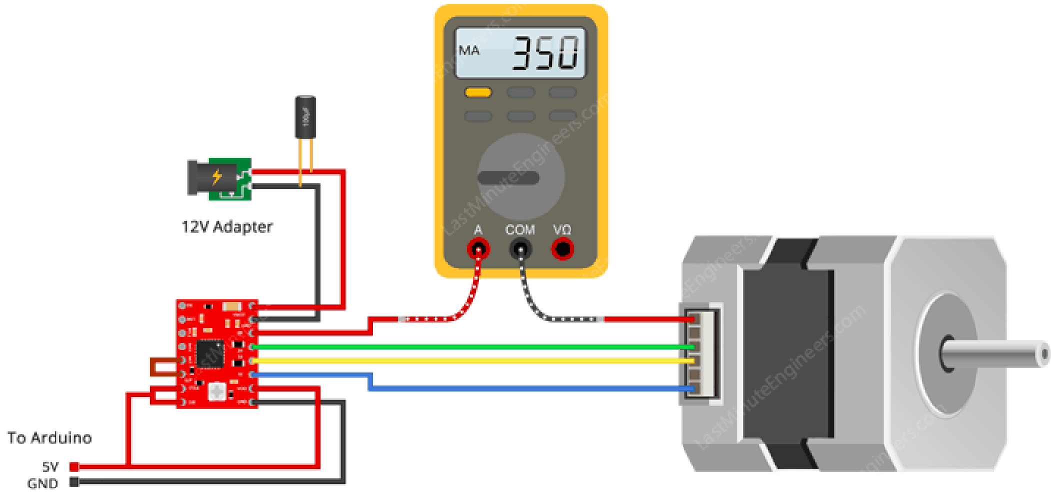



**Figure 4.4:** Current Limiting

## 4.4 Working

Figure 5.1 shows the block diagram of electronic system of the self balancing robot.

The robot functionality starts with the serial data transmission initialization between the Arduino and IMU sensor. All components of the robot wake up from its inactive or sleep mode when the power supply is switched on. Now IMU sensor starts reading data along side-to-side axis (pitch). MPU6050 IMU contains inbuilt DMP that fuses accelerometer and gyroscope data together to reduce error effect of sensor and provides us accurate data. Just after the

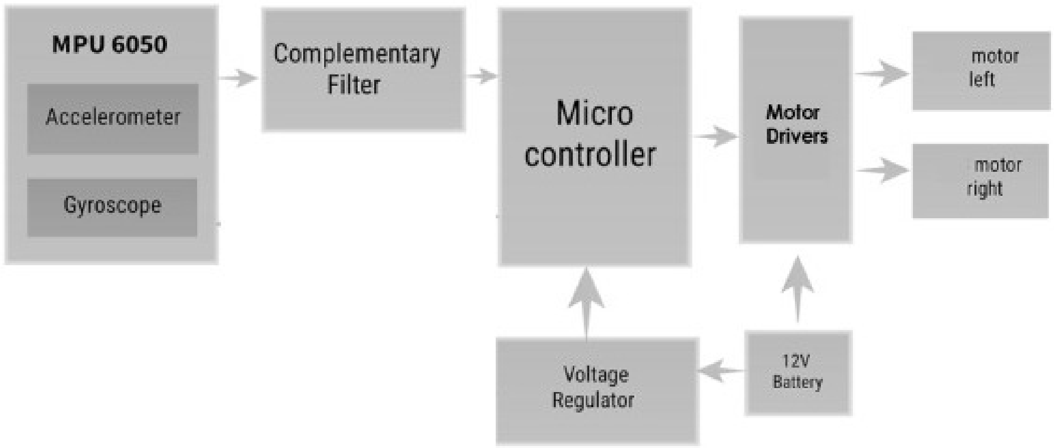



**Figure 4.5:** Block Diagram

start robot gyro takes some milliseconds for calibration of offset value. Next step is to set a desired set point in the system. There is an inbuilt encoder in the MPU6050 which continuously reads the position and velocity of the robot, and depending on that it calculates tilt angle of the robot. We just compare these continuously received raw data bytes with the set point and get the error which we have to eliminate by including a closed loop control system, i.e., LQR to the device. The calculated K matrix of LQR multliplied by tis error gives a control signal i.e Angular Velocity required by the wheels to eliminate that error. The robot is based on two stepper motors which are controlled by the Arduino. Since the stepper motor is a pulse driven device, the calculated angular velocity is mapped to the pulse rate .
if the pulse frequency is increased in the motor then the speed of the motor also gets increased. Since in order to balance the robot, it is moved in the direction of fall so that it can come in upright position. For that the lower part of robot needs to move faster than upper part so that it can come again in balanced vertical position. In order to achieve balance in robot LQR controller helps the Arduino to produce PWM output to make robot move with different required speed to reach set point or to come to upright position.

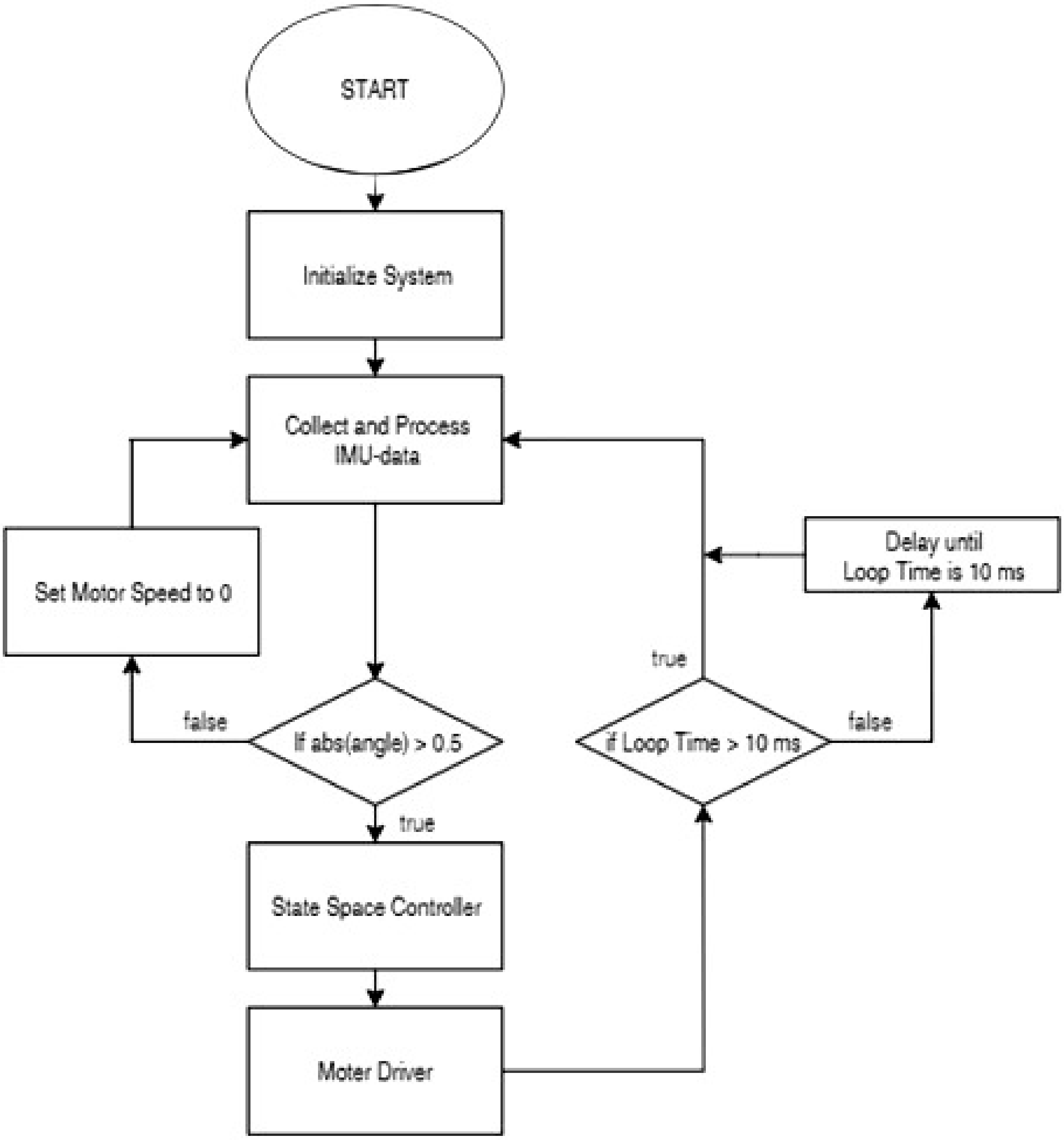



**Figure 4.6:** Flowchart

# Chapter 5

# Results and Disussions

This chapter contains the results from the test performed with the system and discussions about the challenges faced,future scope of this project

## 5.1 Mechanical Properties

The values of the systems mass and its center of gravity is shown in table 5.1.

| | |
|---|---|
| mbody [Kg] | 1.5 |
| mwheels [Kg] | 0.060 |
| Ibody [Kgm2] | 0.011 |
| Iwheels [Kgm2] | 1.47e-05 |
| l [m] | 0.0285 |

**Table 5.1:** Table showing the robots mass,moment of inertia and the length to the center of gravity.

These parameters are found by simulating the self balancing robot in Fusion 360. The robot is modelled in Fusion 360 with the appropriate physical textures and coresponding parameters are analysed.

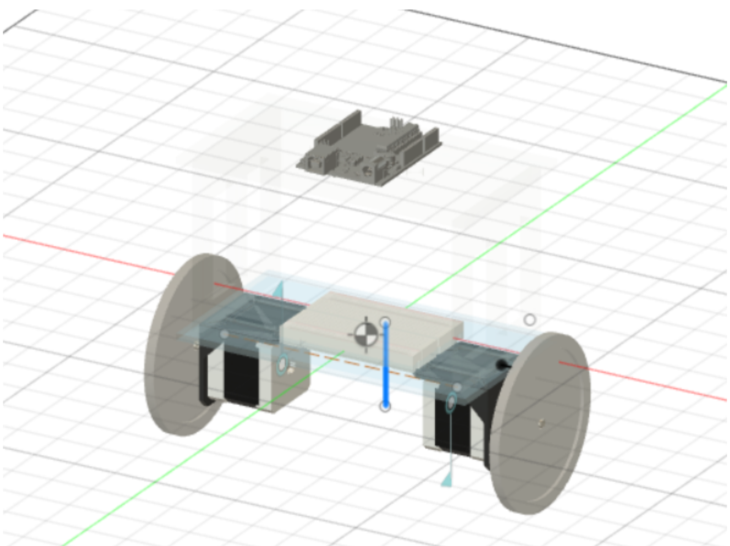


**Figure 5.1:** Center of mass of the system

## 5.2 MATLAB-SIMULINK Model

The LQR gain **K= [28.8275 415.4115]** is calculated using MATLAB. This gain K is validated using the SIMULINK model shown in Figure 5.3

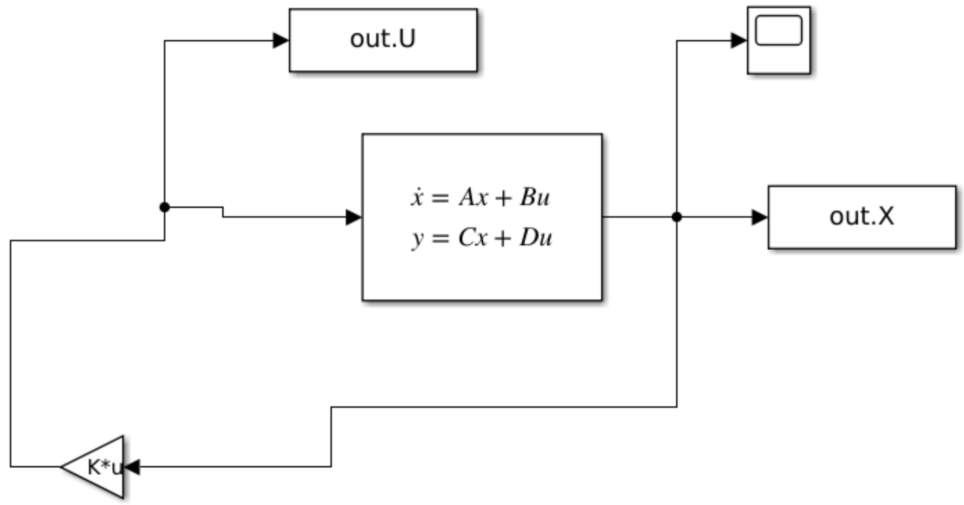



**Figure 5.2:** SIMULINK Model

Here intially the angle of the system is set to 6 degree and the simulated result is observed in Figure 5.4

Here We observe that the system regain balance after 0.5 second. Also the control signal increases and returns to 0 when the system is balanced.

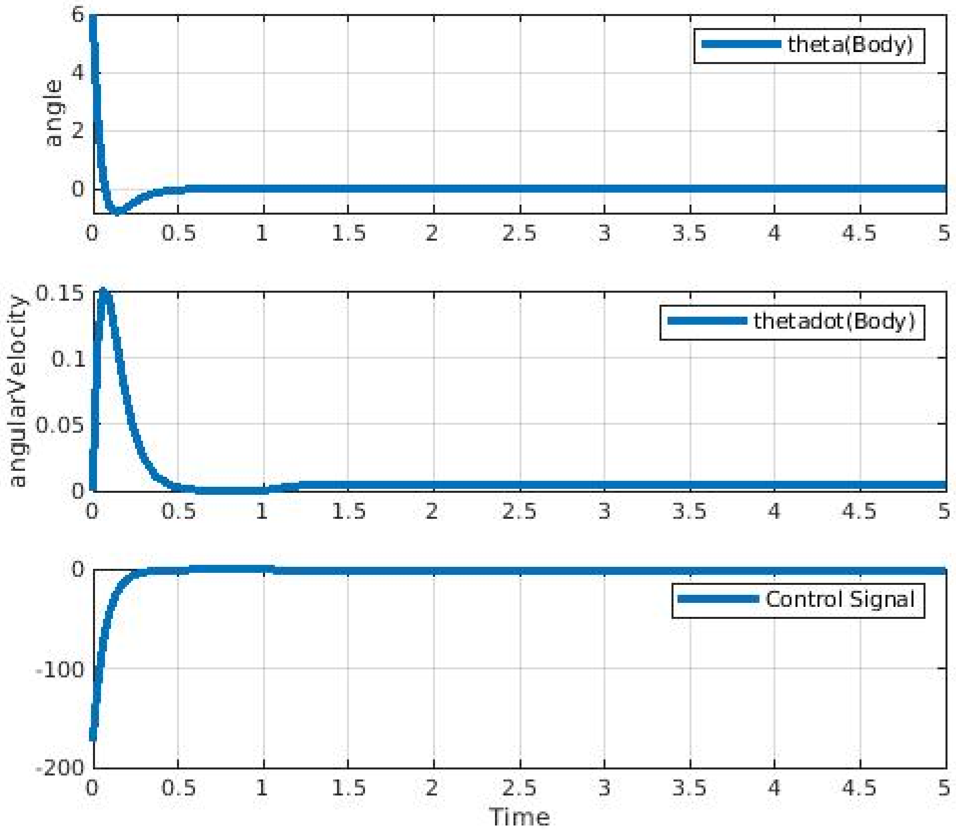



**Figure 5.3:** Simulation Result

## 5.3 CHALLENGES

The biggest challenge that We faced during fabrication and testing of this robot was selection of the motor. Our hardware design follows a minimalist approach and onboard 12V battery was not used primarily due to budget constraints. Ideally, we wanted to use a motor with enconders and high RPM. Encoders allow direct measurement of the speed of the robot, which is necessary to avoid motor speed saturation that may cause the robot to be unable to keep its balance. However, we ultimately settled with stepper motors.That added some necessary complexity to our software implementation, but our robot was able to balance without any serious issues. Another challenge that We faced was to find a wheel with good grip. It is very important to use a good griped wheel, otherwise robot can slip over or fall on the ground even in smaller tilt angles

## 5.4 FUTURE WORK

It can be further enhanced to work with vehicle mechanics. For example, it can be added as a safety feature in two-wheelers. These vehicles can balance themselves sideways whenever the rider loses his balance, therefore can significantly decrease the road accidents rate. The robotic arms can be attached with it for pick and place purposes in the warehouses, hotels and malls. It can be used to develop self-balancing wheelchairs for disabled patients. In future, it can be modified in a legged robot where each movable layers are balanced by using different control systems. Some modification and analysis can be further done in this robot in order to increase the balancing zone. Future work can be done on this robot to make it move and balance itself in different terrains.

## Chapter 6

# CONCLUSION

The project was aimed to fabricate and to test working of the self balancing robot. The design and fabrication of the robot is corrected with fixed position components even at maximum amplitude of vibration.The robot body is able to tolerate the sudden force to good extent in the case of its falling down horizontally on the ground when it goes out of limited balancing zone. The power transmission is sufficient enough to supply the required amount of power to the robot The maximum zone (-20 to 38 degree tilt angle)in which the robot is able to balance itself without any external help, is analysed properly

# BIBLIOGRAPHY

[1] Laubli, David; Garabedian, Thomas; Paul, Jordan; Patel, Nikheel; and Redle, Michael, "Self-Balancing Two Wheeled Robot" (2015). Honors Research Projects.

[2] Modelling  Simulation for Optimal Control of Nonlinear Inverted Pendulum Dynamical System using PID Controller  LQR

[3] Sun, C., Lu, T., Yuan, K., "Balance control of two-wheeled self-balancing robotbased on Linear Quadratic Regulator and Neural Network", 2013. IEEE, 4th International Conference, Intelligent Control and Information Processing(ICICIP), China.

[4] An, W., Li, Y., "Simulation and control of a two-wheeled self-balancing robot",2013. Proceeding of IEEE, International Conference, Robotics and Biometrics(ROBIO), China.

[5] Jamil, O., Jamil, M., Ayaz, Y., Ahmad, K., "Modeling, Control of a two-wheeled self-balancing robot", 2014. IEEE, International Conference, Roboticsand Emerging Allied Technologies in Engineering (iC,REATE), Pakistan.

[6] J.A.Borja*,I.Alvarado*,D. Munoz de la Pena*Low cost two wheels self balancing robot for control education powered by stepper motors

[7] Comparison of complementry and Kalaman filter based data fusion for attitude heading refernce system

[8] C.Gonzalez,... Low Cost two-wheels self-balancing robot for control educatiom