Project Report

**Project Title: Self Balancing robot**.

Md. Juwel Rana (IT-14052)

Md. Asaduzzaman (IT-14053)

Md. Yasin Arafat (IT-13049)

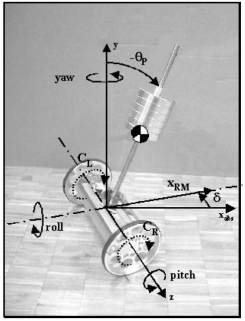
Dept of ICT, MBSTU

**Executive Summary**

The objective of this project is to design and implement a self- balancing algorithm to help physically disabled people. The implementation utilized both a 6 dimension free (6-DOF) accelerometer and a rate-gyroscope built into the micro-controller in order to achieve a vertical balance.

The fusion of both sensor data into a single usable value is achieved through a complementary filter (Kalman filter). Consequently, the output of the complementary filter is designed to be primarily dependent on the gyroscope data, to which a fraction of the accelerometer data is added to compensate for the gyroscopic drift.

A high-current H-bridge circuit is connected to the control loop, which included both the software  
implementation of the complementary filter and the PID controller, was measured to run at 530 Hz (±20Hz). Additionally, a PWM signal generator is implemented in software using the interrupt service routines of the micro-controller. Consequently, this resulted in a robust code-base which was able to achieve a self-balance with an oscillatory amplitude of 1 cm (±0.3cm) and a balance time of about 15 seconds.



**FIGURE: THE 6 DEGREES OF FREEDOM (6-DOF) OF THE SYSTEM.**

**Instruments:**  
(1) Arduino Mega 2560.  
(2) Motor driver L298N.  
(3) MPU-6050 sensor 6-DOF IMU (3-AXIS Accelerometer ADXL345 Gyroscope Gyro L3G4200D)

(4) 9 volts Li-ion Battery.  
(5) Two gear motors and wheels.

(6). Jumper wires.  
(7) Chassis and screws.

**Mathematical Calculation:**

The z-axis (pitch) by an angle = 

Corresponding angular velocity = .

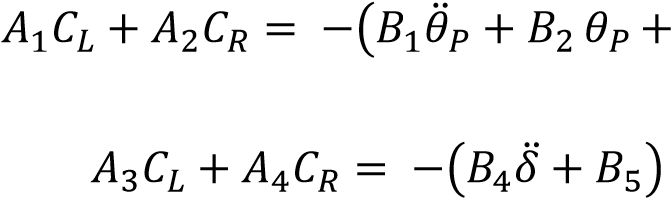
The linear movement of the robot is characterized by the position = xRM

and the velocity = vRM.

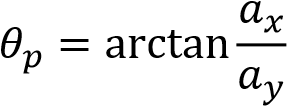
The vertical axis (yaw) by an angle = δ with a corresponding angular velocity = .

The robot’s motion would be controlled by applying torques CL and CR to the corresponding wheels.

So,

  **(1)**

**(2)**

**(3)**

** (i) =  (i-1) + 1⁄6 (vali-3 + 2 vali-2 + 2 vali-1 + vali)** **(4)**

**(5)**

Where R1 models the robot when it is balanced at a stationary position and R2 models the robot when it is turning about its y-axis while maintaining a balanced posture.

Control Algorithm



Calculate Pitch

Angle



PID Controller



Motors



Accelerometer



Gyroscope

Control Algorithm



Integrate Angle



PID Controller



Motors



Accelerometer

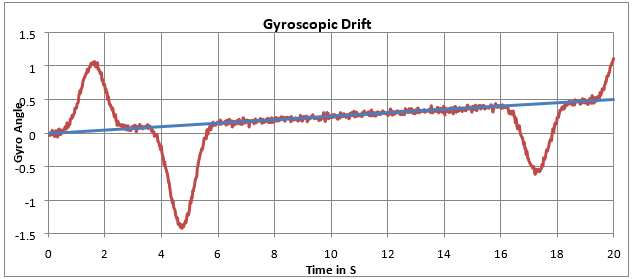


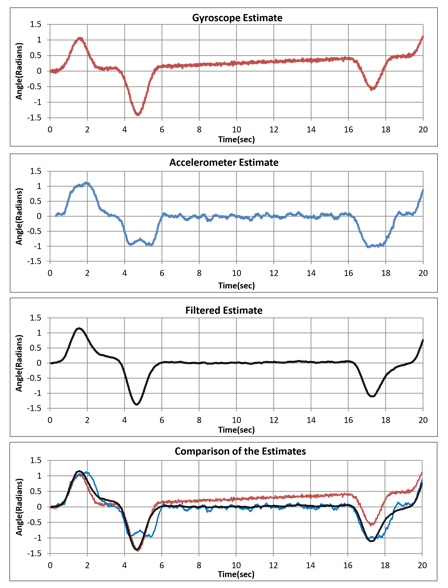
Gyroscope

**FIGURE: CONTROL SYSTEM FOR ACCELEROMETER AND GYROSCOPE**

**Data Analysis of Gyroscope and Accelerometer**

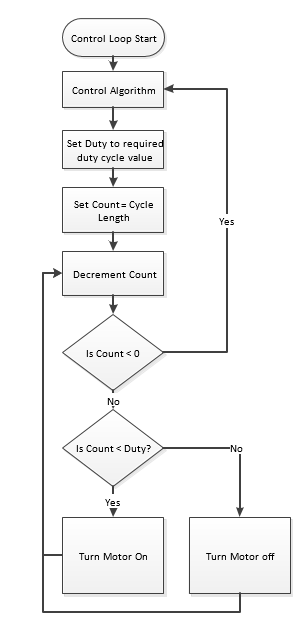
A common drawback of gyroscopes is that there exists a small DC bias which, upon integration, would cause the zero point to drift overtime. Hence, a balancing robot based solely on a gyroscope would be vertical for a few seconds and eventually fall over due to the drift of the zero point. The effect of gyroscopic drift can be clearly seen in the Figure .





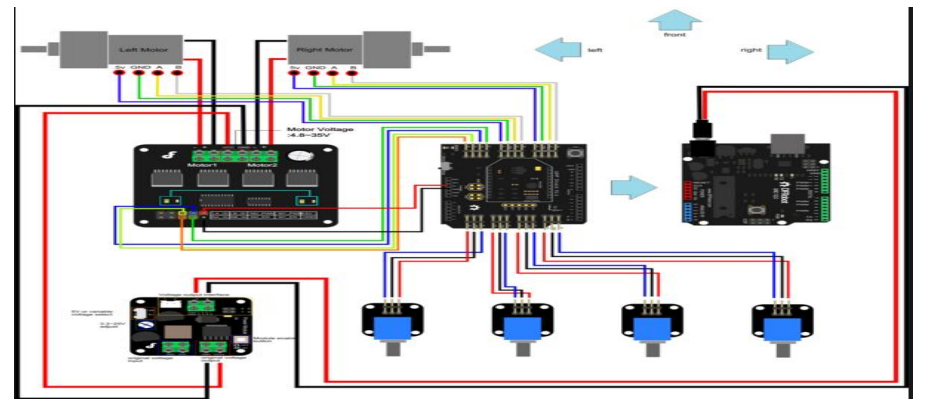
**FIGURE: GRAPH SHOWING THE EFFECTS OF GYROSCOPIC AND ACCEROMETER DRIFT**

**Working Principle:**

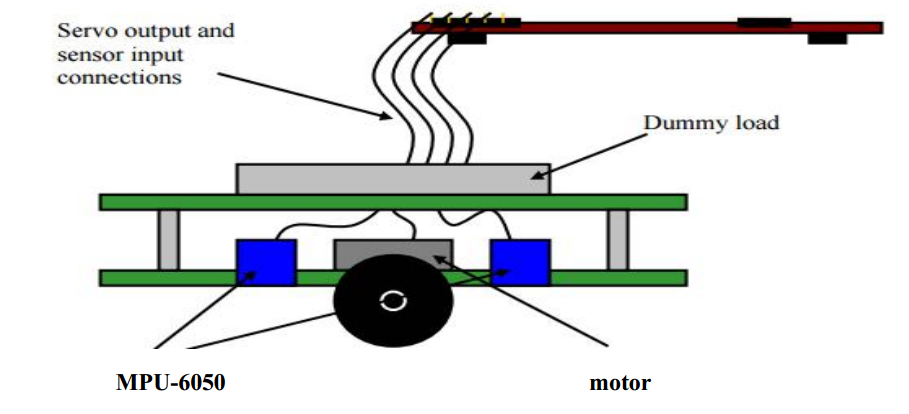


**FIGURE: LOGICAL WORKING PROCEDURE**

**Circuit Diagram:**



**FIGURE: Circuit Diagram**



**FIGURE: Demo Model of self Balancing Robot**

**Implementation Plan:**  
The most critical part of the project has progressed from a software issue (i.e. concerns about  
programming the PID controller) to a mechanical issue – obtaining a suitable balancing platform:  
(a) Obtain the circuit diagram with Arduino MEGA 2560 microcontroller, bot platform, tilt (MPU-6050) sensor, battery pack (or other mobile power source), and servo motor control board.  
(b) Construct scooter bot and ensure that it is susceptible to tipping when an appropriately weighted load is attached. Make any modifications to the base platform as necessary.  
(c) Experiment with tilt sensor and stepper motor to understand its behavior and characteristics.  
(d) Assemble the components in either the “self-contained” or “external controller” configuration  
as illustrated above.

**Future Implementation:**

(a) In future, we will upgrade this robot with voice controlling and Self -Lifting capability.

(b) We also can upgrade this robot as a Maze solver.

**References:**  
(1) www.google.com , www.instructables.com , www.youtube.com.  
(2) Robot building for beginners –David Cook.