Basic Electronics End Semester Project
First Report

Saravan Sriram G.(IMT2018521)
Shanthan Reddy(IMT2018522)
Shubhayu Das(IMT2018523)
Veerendra S. Devaraddi(IMT2018529)

2nd November, 2019

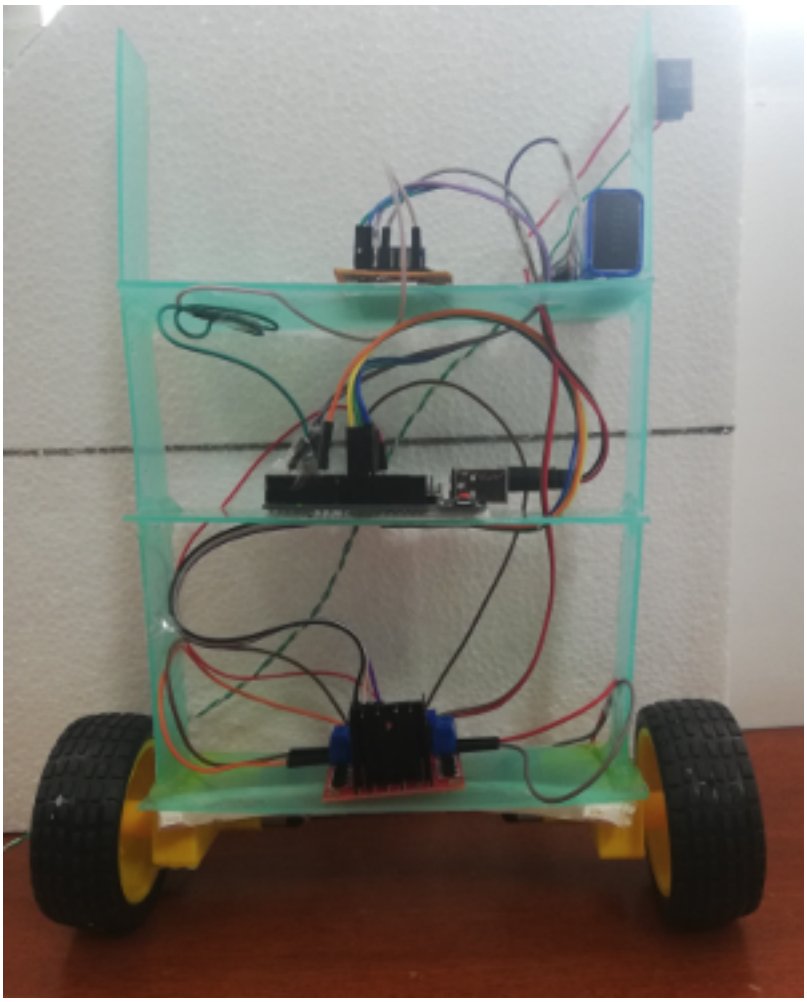
Contents

1 Abstract

This is the report to our end semester project of the Basic Electronics course(EC-201P).The aim of this project is to make a *self-balancing robot*. The idea is to make a two-wheeled robot balance itself, at a vertical normal to the ground plane. The sensor was first tested by us on a breadboard to get a general idea of its working. Then the sensor was attached to our model and after some PID tuning, made to work alongside a motor driver.

2 Introduction

A self balancing robot is an application of control theory. We have made a robot that can continuously maintain a vertical orientation, even when an external force is applied on it.



3 Concepts

We have learnt and used the following concepts in the course of this project:

1. **I2C communication** - We have set up and used the I2C protocol to communicate with the MPU6050 module.

2. **Fundamentals of PID control** - The model works on the basis of control theory: take in data from the accelerometer and power the motor accordingly to maintain its position. We have essentially set up a feedback mechanism for the motor driver, via an Arduino Uno for processing the data and generating a control PWM signal.

3. **Basic Arduino library management** - We have gone through open sourced libraries to figure out the working of the sensor and data gathering through I2C. This gave a high level insight into how custom libraries can be written and worked with.

4. **Physics involved** - If the robot gets tilted by an angle, wrt frame of the wheel, the center of mass of the robot will experience a pseudo-force which will apply a torque opposite to the direction of tilt. This pseudo force is due to the acceleration of the wheels in the direction of the tilt. This acceleration will be be picked up by the sensor.

4 Results

We have successfully made a model, which can balance itself(along with some load) in a vertical position. Due to imperfections in our PID tuning and some errors in the code libraries, the robot may need occasional nudges.

The robot can absorb small disturbances($<5^\circ$ tilts at the sensor). For larger angles, the motors are unable to respond fast enough and/or end up overshooting the balance point, resulting in larger oscillations. This effectively causes the robot to fall.

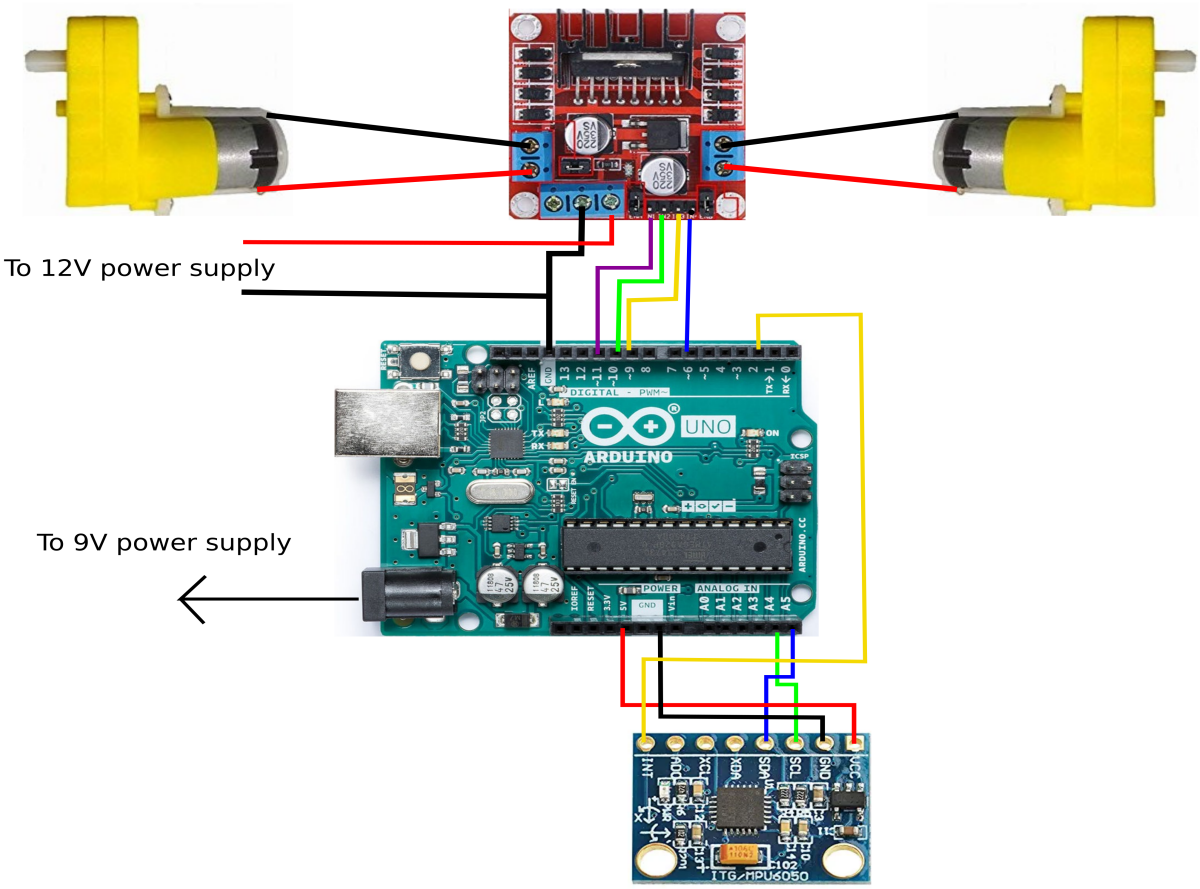
Part of the reason for this problem is improper PID tuning. We have a general idea as to how PID control works, but not how to implement it perfectly in a model.

5 Applications

1. The idea is already utilized in Segways.

2. We can use this to keep storage containers in a stable vertical orientation.

6 Circuit diagram



7 Components used

*prices are from the time of purchase and liable to change.

Item name	Quantity	Unit Price	Total price
B.O. motors	2	₹82	₹164
2A L298N motor drivers	1	₹148	₹148
Wheels	2	₹63	₹126
MPU6050 module	1	₹70	-
Plastic exam pad	1	₹70	₹70
Arduino Uno	1	₹400	-
Wires, solder, etc	-	<₹20	-
Delivery charges	-	₹99	₹99
		Total money spent:	**₹607**

8 Future improvements

Hardware based:

1. Better motors such as stepper motors with encoders for finer control over the angle turned.
2. Rigid material(e.g. Acrylic plastic, wood etc) for the frame.
3. PCB for securing the gyroscope/accelerometer sensor, Arduino and motor driver in a static configuration.

Software based:

1. Better libraries for the sensor and PID control.

9 References and sources

Link to code: Github

1. PID control theory
2. I2C communication protocol
3. MPU6050 library
4. MPU6050 datasheet