Self Balancing Robot Using MPU6050

Anshu De,Giridhar Indurkar,Kaustubh Kshirsagar*

A B S T R A C T

Self-balancing robots are a topic of curiosity amongst students, roboticists, and hobbyists around the world. The fascinating aspect is the fact that it is a naturally unstable system. This project presents an attempt on developing an autonomous self- balancing robot. A key element in maintaining the robot in the upright position is estimation of the tilt angle. For this, the Kalman Filter has been implemented and tested to fuse data from a gyroscope and an accelerometer. In addition, the method- ology in which the hardware was chosen and put together has been justified. Then the software development and challenges in the implementation of the PID Control have also been explained. Lastly the control of the robot has been explored, with characterization of the robot in an attempt to implement an Bluetooth Control Robot.

Keywords: Self-balancing robots, PID Control, Arduinonano

1. Introduction

Self-balancing robots have been a topic of interest of many researchers, students and hobbyists worldwide. In essence, it is an inverted pendulum on wheels, a derivative of the inverted pendulum on a cart. Unlike traditional robots, which are in a constant state of equilibrium, the robot is a naturally unstable system. Its design is more complex, as it needs to be actively controlled to maintain it supright position, however, it benets from being able to turn on the spot. The primary practical application of a self-balancing robot is human transportation, which was popularised by the release of the Segway PT (Personal Transporter). It is used in many industries such as inside factory Floors or for tourism in the park. It is more attractive compared to four or three wheeled vehicles as they can take sharp turn sand navigate in tighter spaces.

2. Problem Definition

The primary incentive of the project is to develop general understanding of control theory. For the last few decades, 0nverted pendulum has been the most popular bench- mark, among others, for teaching and research in control theory and robotics. Hence, developing a self-balancing robot is the ideal platform to put into practice what has been covered in Control Systems lectures. It would also be interesting to see the differences between the behaviour in practice compared to simulations. Furthermore, the material and methods learnt have a wide array of applications; for example, inverted pendulums have been used to model human locomotion, which then was used to de- velop bipedal robots. Besides learning about the theoretical aspects, the project also incorporated a practical side.

3. Project Design

To determine the appropriate motors for the robot, the rest consideration was the minimum torque required. To estimate the torque, the model to the below is considered.

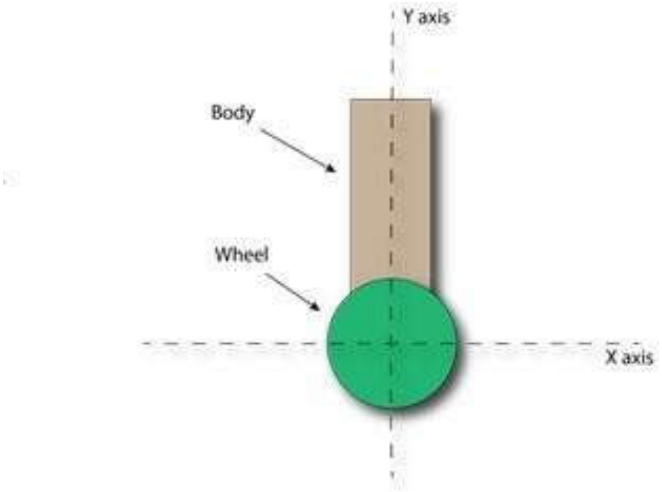

Fig. 1. Cross section of the robot

Where is the position vector and is the angle between force and position vectors. Assuming the distance between the pivot point and the centre of mass () is 12cm, the maximum tilt angle (max) is 40 and the mass of the robot () is 0.7kg, the minimum torque required is =0.120.79.81sin(40)=0.530
(35) ven that there are two motors, the torque required per motor is 0.265Nm. In this scenario, the inertia is not taken into account and the robot is assume to start moving at max. Due to the crude assumptions, the robot by Gornicki was looked at [16]. In which the stall torque of the motors were 4.5kgcm and a gear with a 3:1 ratio was used. With the assumption of 15 Looking solely at the parameters mentioned, the chosen motor was the Pololu medium power 47:1 Metal Gearmotor with 48 CPR Encoder. The motor has a stall torque of 4.5kgm andtheencoderoutputs200countsperrevolution.
The microcontroller used in the robot is the Arduino Nano. It is a board based on the ATmega328P from Atmel's AVR family. It is an 8-bit microcontroller running with a 16 MHz clock speed. The board has a built-in voltage regulator allowing it to be powered by any input voltage in the range of 5-20V. On-board is also an FTDI chip, making only a USB cable required to program it. The board has a relatively small size,maintaining the robotas small as possible. The main advantage of the Arduino is the IDE and the large community. Its IDE enables fast software development due to the extensive collection of libraries and sample code. The large community is helpful in the case where a problem is encountered, there is a higher chance that someone else has found a solution and it is visible in one of the many forums available.

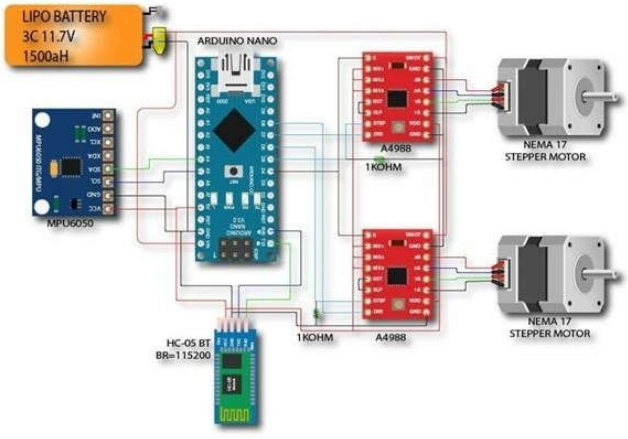

Fig. 2.electrical schematics

This section provides in detail the key aspects of programming the robot. The code is composed of various sections found online, this maybe in the form of forums, blogs or tutorials. Throughout the project various versions of code developed, each for a specic purpose, for example just printing the output of the Kalman Filter and the accelerometer angle to view the performance of the Kalman Filter or code to characterize the motors. Part of the code is the same as it would be in an equation from, thus only the non-trivial aspects are considered here and the reasons behind are mentioned. To begin with, the important sections of the code are mentioned. Later, the block diagram of the code is given to provide a higher level view of the code. Given more time and resources, better motors could be purchased to make the robot balance. The development of a self-balancing robot may extended in many directions for future work, the yaw can be considered, PID controller can be implemented or remote control using Bluetooth communication.
.

A PID controller was chosen because of its three levels of control each of which has a particular situation it is best at. PID is an acronym for these three control methods the P standing for proportional, I for integral and D for derivative. The MPU6050 supplied the necessary rotational and orientation data to the arduinonano for the PID controller. The P term is what has the fastest and most drastic impact on the output to the motors. The I term is for more fine control of the drones stability and the D term is only implemented to fix sudden drastic changes in stability caused by things like a gust of wind.This meant that the arduino needed to be programmed to receive the information from the MPU6050 and send the necessary signals to the driver to maintain the stability.

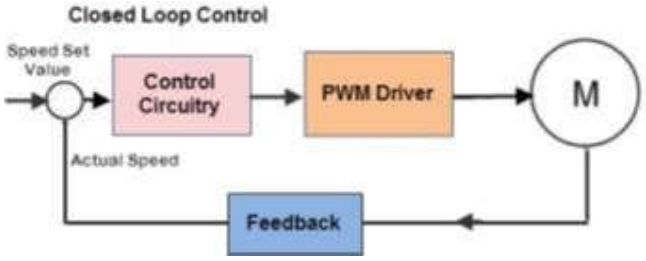

Fig. 3. PID controller block diagram

In terms of the above diagram, the arduinonano served as the control circuitry and the motor driver acted as final control element and feedback with motor being represented as M. Setting of the desired destination coordinate is done using a telemetry. Figure below shows the final bot.

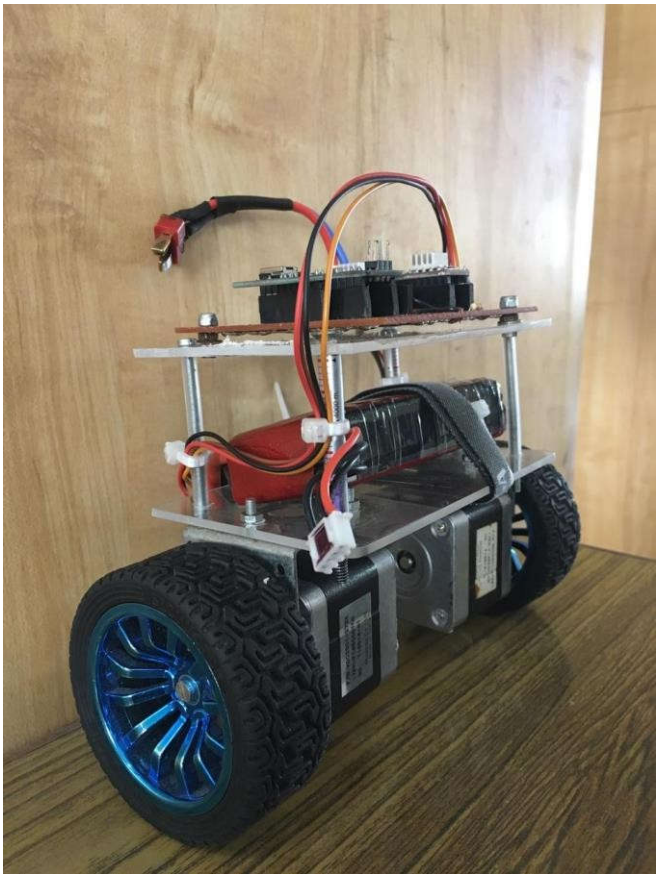

Fig. 4. Finalbot

4. Conclusion

Objectives set in the beginning of the project were met, with the exception of making the robot balance by itself using PID. The project as whole was a steep learning due to the wide array of disciplines involved from construction and design, to control and software implementation

REFERENCES

V. Georgitzikis and I. Akribopoulos, O.and Chatzigiannakis, "Controlling physical objects via the internet using the Arduino platform over 802.15.4 networks," IEEE Latin America Transactions, vol. 10, no. 3, pp. 1686–9, 2012

G. M. T. Nguyen, H. N. Duong, and H. P. Ngyuen, "A PID backstepping controller for two-wheeled self-balancing robot," in Proceedings fo the 2010 IFOST, 2010

D. P. Anderson. (2003, Aug.) nBot balancing robot. Online. [Online]. Available: http://www.geology.smu.edu/ dpa-www/robo/nbot

T. Nomura, Y. Kitsuka, H. Suemistu, and T. Matsuo, "Adaptive backstepping control for a two-wheeled autonomous robot," in ICROSSICE International Joint Conference, Aug. 2009, pp. 4687–92

Thao, N. G. M., Nghia, D. H., & Phuc, N. H. (2010, October). A PID backstepping controller for two-wheeled self-balancing robot. In International Forum on Strategic Technology 2010 (pp. 76-81). IEEE.

M. J. A. Arizaga, J. de la Calleja, R. Hernandez, and A. Benitez, "Automatic control for laboratory sterilization process rsd on Arduino hardware," in 22nd International Conference on Electrical Communications and Computers, 130-3, Ed., 2012