International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Image Processing Based Self Balancing Robot

Rasika Kantute[1], Naishadh Nimbark[2], AnkurTyagi[3]

[1]School of Sense, VIT University, Vellore, India

Abstract: ***Now-a-days, self-balancing robots are becoming popular because of their ability to balance itself on two wheels. The project is designed to implement the self-balancing robot which balances itself on two wheels and navigate its path around and perform image processing. Balancing unit and the image processing unit forms the entire system. The balancing unit performs the function of keeping the robot stable and the image processing unit is used to navigate the path. The System has been designed using IMU6050, Arduinoboard, two DC motors and the raspberry pi. The PID controller is used to improve the stability of the system and the image processing system is developed to detect the variety of signs such as start, stop, and right, left and navigate the robot accordingly.***

Keywords: MPU6050, Arduino board, Raspberry Pi,PID controller

1. Introduction

Self-balancing robot balances itself on two wheels and provides the unique stability which differentiates it from other ordinary robots. This ability of robot allows it to navigate on various terrains, sharp corners etc. These abilities of robot solves the number of challenges in industry and the society. The basic idea of self-balancing is to drive the wheels in the direction in which the robot tilts. The general design of the robot is the simple rectangular body on two wheels .The robot body comprises of the three layers placed on one another. The bottom layer consists of the two wheels and MPU6050 sensor. The first layer consists of the Arduino board, L298N the motor driver circuit and the MPU6050 sensor. The second layer consists of the raspberry pi which is used for image processing. Balancing robot is designed using 6DOF MPU6050, Arduino board and the two DC motors. Here the MPU6050 consists of accelerometer which gives the component's accelerationn and the Gyroscope gives the component's angular velocity along its three axis. The sensors communicates over I2C protocol. The microcontroller (Arduino) continuously reads the data from the IMU sensor and calculates the angle of the tilt of the robot with respect to the setpoint, with the help of the data.

The controller sends the appropriate signals to the motor driver circuit to drive the motors. Raspberry pi is used for Image processing which operates at 700MHz.

1.1 Block Diagram

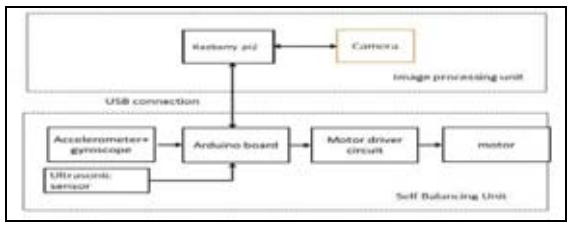

Figure1: Block Diagram

2. Working

In order to keep the robot stable, we have to continuously monitor the angle of the tilt of the robot and drive the wheels accordingly.

2.1 Angle estimation

The IMU sensor consisting of accelerometer and the gyroscope is used to find the angle of the tilt. Accelerometer and gyroscope are individually unstable and cannot give accurate angle so we need to combine the data to get the accurate angle. The accelerometer is used to measure the linear acceleration and the Gyroscope measures angular velocity which is integrated to find the angle of tilt. For a small interval of time, the gyroscope value remains accurate, but since it experiences the drift and integration gives the error, after some time the gyroscope reading becomes unreliable. So we require some way to combine these two values.[1] This is done with the help of the Kalman filter. The filter is used to remove the noise that can affect the performance of the system under control. Kalman Filter is not easy to implement and requires heavy effort in programming. [2] Kalman filter will predict the future state from current and previous states using weighted averaging.

2.2 Balancing control:

The PID control algorithm used to maintain the balance on two wheel robot. PID stands for proportional, integral, and derivative terms. [3] These terms have their own gain constants kp,ki and kd known as the proportional gain, integral gain and the derivative gain constants. PID controller uses the concept of negative feedback control.The measured position will be subtracted from the reference set point value to produce an error. [4] The error is given back to the PID controller. The PID algorithm processes this error and tries to adjust the input signal provided to PID. This control signal will help to drive the motor through motor driver circuit to correct the position.The set-point value is defined as 'zero degree'. PID can be represented in mathematical form as follows:

$$\text{Correction}=Kp*\text{error}+Ki*\int\text{error}+Kd*d/dt(\text{error}); \quad (1)$$

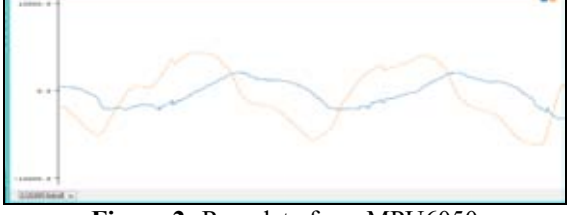

Figure 2: Raw data from MPU6050

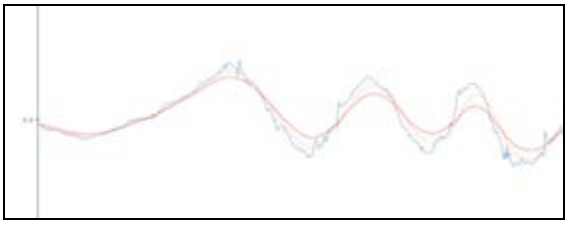

Figure 3: Comparison of Kalman filtering and complementary filtering

The gain values Kp, Ki and Kd can be set through various algorithms like Ziegler-Nichols, Twiddle or Auto tuning. The comparison of P, PI and PID controller is given in figure 2.We have used a practical approach to set these values. The P is responsible for oscillations of the system, theI is responsible for response time of the system and D is used to increase the sensitivity of the system. The values of these gain constants are set experimentally. First we make the Ki and Kd values as zero and vary the Kp value till we get sustained oscillations. [5] Once the Kp value is set we vary the Ki value and afterwards the Kd value to increase the response time and sensitivity of the system.

Table 1: ziegler – nichols

Control Type	K_p	K_i	K_d
P	$0.5K_u$	-	-
PI	$0.45K_u$	$1.2K_p/T_u$	-
PD	$0.8K_u$	-	$K_pT_u/8$
PID	$0.6K_u$	$2K_p/T_u$	$K_pT_u/8$

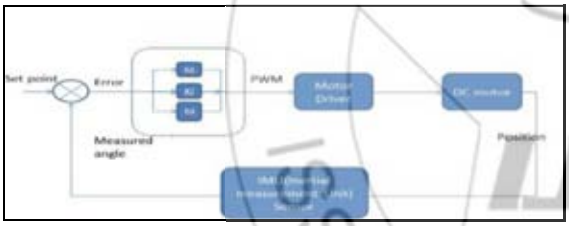

Figure 4: Modeled system

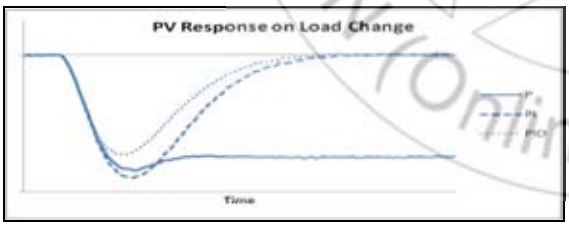

Figure 5: Comparisons of P, PI and PID

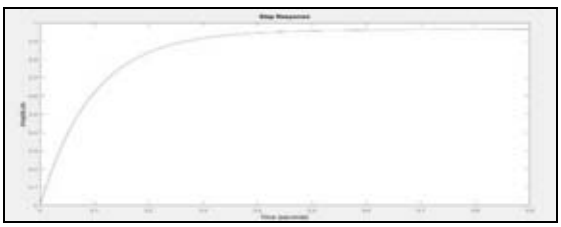

Figure 6: PID response for values of Kp, Ki and Kd using MATLAB simulation.

3. Image Detection

The image detection and the navigation of the robot can be achieved by using the ultrasonic sensor and the image processing system. With the help of this feature the robot is able to navigate its path and decides what to be done when it detects the obstacle. For this purpose, raspberry pi is interfaced with the digital camera. The ultrasonic sensor is interfaced to the Arduino board.[6] The ultrasonic sensor uses the concept of sonar to detect the obstacles and measure the distance between them. The sensor transmits a high frequency sound wave on applying the appropriate trigger signal which is not audible to humans. The wave is transmitted and gets reflected by the obstacle and it is captured by the receiver, for calculating the distance between the sensor and the obstacle , the elapsed time is calculated between sending and the receiving waves.[7] So the distance in meters can be calculates as ,

$$\text{Distance} = 343 * \text{elapsed time} / 2 \quad (2)$$

The Arduino board continuously measures the distance of the robot to the nearest obstacle in front of it. This distance is then compared with the reference value for e.g. 30 cm in this case and if this distance is less than the reference value, the Raspberry pi triggers the camera and captures the image. The image is then processed to obtain some useful information. Based on the information obtained from the image, the microcontroller navigates the path of the robot.[9] The colour image captured by the camera is converted to black and white image. The image is then cropped to remove any unwanted part of the frame. The size of the image can be reduced since the captured image is of reasonably high resolution. The size gets reduced and this image can be scanned for recognizing any symbol on it. For recognizing the symbol template matching has been used. The raspberry pi sends the appropriate signals to the microcontroller to turn the robot in the appropriate direction. This shows that using the image processing system the robot can easily navigates its path and move around accordingly.

4. Conclusion

We have designed and implemented the self-balancing robot which balances itself on two wheels and navigate its path around with the help of image processing system.Here by varying the various values of kp,ki, and kd of the PID controller and by using the Kalman filter system is balanced and is capable of navigating along the smooth surface as shown.

References

[1] Obstacle Detection and Map Building with a Rotating Ultrasonic Range Sensor using Bayesian Combination S. Kodagoda, E.A.S.M. Hemachandra, P.G. Jayasekara, R.L. Peiris, A.C. De Silva and Rohan Munasinghe (Corresponding author) Department ofElectronic and Telecommunication Engineering, +Department of Electrical Engineering University ofMoratuwa Colombo, Sri Lanka.
[2] Automatic Detection and Recognition ofCircular Road Sign Hua Huang, Chao Chen, Yulan Jia, and ShumingTang
[3] Dynamic Model and Balancing Control for Two-Wheeled Self-Balancing Mobile Robot on the Slopes Kui PengXiao gang Ruan, and Guoyu Zuo School of Electronic Information and Control Engineering Beijing

University of Technology, Beijing.
[4] A PID Backstepping Controller For Two-Wheeled Self-Balancing Robot Nguyen Gia Minh Thao(l), Duong Hoai Nghia(2), Nguyen Huu Phuc(3) Faculty of Electrical and Electronics Engineering HoChiMinh city University of Technology (HCMUT) HoChiMinh city, VietNam.
[5] Two-wheel self-balanced car based on Kalman filtering and PID algorithm LIU Kun, BAI Ming, NI Yuhua College of Information Technology Beijing Normal University at Zhuhai Zhuhai, China
[6] PID Control Behavior and Sensor Filtering for a Self Balancing Personal Vehicle M. Abdullah Bin Azhar, Waseem Hassan, and Usman Rahim Electronics & Power Engineering Department, PN Engineering College, National University of Sciences & Technology Karac
[7] Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino Microcontroller Board Haut-Shiue Juang1 and Kai-Yew Lum2
[8] Servo State Feedback Control of the Self Balancing Robot using MATLAB V. Kongratana1, S. Gulphanich1, V. Tipsuwanporn1 and P. Huantham1 1Department of Instrumentation and Control Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand.
[9] Simulation and Control of a Two-wheeled Self-balancing RobotWeiAn[1] andYangminLi[12], SeniorMember,IEEE
[10] Ensemble Kalman Filter and PID Controller Implementation On Self Balancing Robot Barlian Henryranu PrasetioComputerEngineeringandRobotics Lab Faculty of Computer Science, University of Brawijaya Malang, Indonesia
[11] Two wheeled self-balancing robot for autonomous navigation Jisha Kuruvilla1, Jithin Abraham2, Midhun S2, Ranjini Kunnath2,Rohin Reji Paul2 Asst. Prof., Dept. of EEE, Mar Athanasius College of Engineering, Kothamangalam, India UG Student, Dept. of EEE, Mar Athanasius College of Engineering, Kothamangalam, India

Author Profile

Rasika Kantute received the B.E degree in Electronics and Tele-communication from SGBAU University, Amravati and currently pursuing M. Tech in embedded system from VIT University Vellore. Studied RTOS and embedded system lab during Masters to study the basics of embedded c and RTOS and is now with TATA MOTORS LTD asan intern.

Naishadh Nimbark received the B.E degree in Electronics and Communication from Dharmsinh Desai University, Nadiad, Gujarat and currently pursuing Master's degree in the field of Embedded Systems. As a part of the academic curriculum pursuing an internship at Intel Corporation, Bangalore.

Ankur Tyagi received Btech degree in Electronics and communication from UPTU, Lucknow and currently pursuing M tech in Embedded systems from VIT University, Vellore. Currently pursuing internship at Ingersoll Rand, Bangalore.