Proceedings of the
International Conference on Mechanical Engineering and Renewable Energy 2017
(ICMERE2017) 18 – 20 December, 2017, Chittagong, Bangladesh

ICMERE2017-PI-294

DESIGN AND IMPLEMENTATION OF A SELF-BALANCING ROBOT

A.K.M Ashfaque Shakil Shaon [1], Subrata Bhowmik [2], Bikash Kumar Bhawmick [3],
Piyas Das [4] and Nipu Kumar Das [5]

[1234]B.Sc. in EEE, Chittagong University of Engineering and Technology, Bangladesh,
[5]Assistant Professor, Dept. of EEE, Chittagong University of Engineering and Technology, Bangladesh
[1]ashfaqueshaon@gmail.com, [2]psbhowmik12@gmail.com, [3]anjan.13@outlook.com, [4]pnb.cuet@gmail.com,
[5]nipu16@gmail.com

Abstract- *The field of robotics is the playground of the creative minds of modern age. Dreams turned into reality with the development in this field. Two wheel self-balancing robot is also an example of advanced development in the field of robotics. The concept of two wheel self-balancing robot is based on Inverted pendulum theory. This type of robot has earned interest and fame among researchers and engineers of worldwide as it based on such a control system that is used to stabilize an unstable system using efficient micro controllers and sensors. These robots provide exceptional robustness and capability due to their smaller size and power requirements. These types of implementations find applications in several purposes such as surveillance & transportation. This project is based on development of a self balanced two wheeled robot. In particular, the focus is on the electro-mechanical mechanisms & control algorithms required to enable the robot to perceive and act in real time. Similar concept can be applied in various control system with complex implementation such as humanoid robot, industrial robots, etc.*

Keywords: Robotics, Accelerometer & Gyro, Self Balance, Microcontroller, Control System

1. INTRODUCTION

The self balancing robot, concept of two-wheel inverted pendulum has gained momentum in research over the last few years. Inherently self balancing robot is unstable and it would roll around the wheels' rotation axis without external control and eventually fall. The robot returns to right position if motor driving occurs in right direction. The robot is naturally unstable although, it has many favors over the statically stable multi wheeled robots. A special electromechanical system in which the robot has to be based on balances itself onto a pair of wheels while standing tall. If the base on which the robot stand is not stable or the platform is not balanced, the robot tend to falling off from the vertical axis. This time a gyro chip is needed to provide the PID controller about the angular position of the base of the self balancing robot [1]. A self balancing algorithm is programmed into the controller and the controller drives the motors either clockwise or anticlockwise to balance the basement by a pulse width modulation (PWM) control signal. The robot has to be work upon any type of surface based on two motors constructed with wheel one for each [2].

2. DESIGN PROCESS

The self balancing robot gets balanced on a pair of wheels having the required grip providing sufficient friction. For maintaining vertical axis two things must be done, one is measuring the inclination angle and other is controlling of motors to move forward or backwards to maintain 0˚angle with vertical axis. For measuring the angle, two sensors, accelerometer and gyroscope are used. Accelerometer can sense either static or dynamic forces of acceleration and Gyroscope measures the angular velocity.

The outputs of the sensors are fused using a Complementary filter. Sensors measure the process output say α which gets subtracted from the reference set-point value to produce an error. Error is then fed into the PID where the error gets managed in three ways. After the PID algorithm processes the error, the controller produces a control signal μ. PID control signal then gets fed into the process under control. Process under PID control is two wheeled robot. PID control signal will try to drive the process to the desired set-point value that is 0˚ in vertical position by driving the motors in such a way that the robot is balanced. The simplified block diagram is shown in Figure 1.

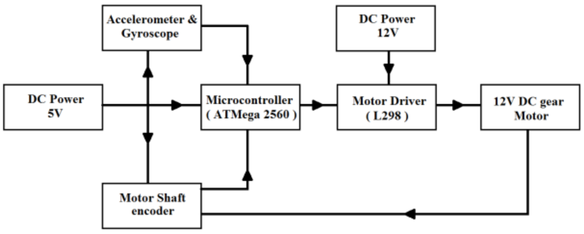

Fig.1: Block diagram of self - balancing robot

3. CONTROL TECHNIQUE

It is a challenging work to balance an inverted pendulum, because naturally it is unstable. The small error or disturbance from equilibrium position takes away from equilibrium forcefully that further destabilizes the system. So, keeping balance at a slightly non-equilibrium requires precise control to immediately correct any errors in tilt the instant they happen [3].

To deal with this problem, a PID controller is employed that uses tilt feedback to control the torque of the motors and keep the robot balanced. A PID controller continuously measures a process variable and calculates an error value (angle from the vertical), which is the deviation of the process variable from some desired value (0 degrees from the vertical) [4]. The PID controller try to minimize this type of error over time by continuously adjusting a control variable (motor torque) according to the following equation, where u(t) is the control variable, e(t) is the current error in the process variable, and K_p, K_i, and K_d are coefficients that must be tuned to achieve the desired behavior of the controller:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}$$

4. DATA ACQUISITION AND FILTERING

Both the accelerometer and gyroscope data are used to obtain the angular position of the object. Gyroscope does this by integrating the angular velocity over time. From the position of the gravity vector (g-force) angular position is obtained using the accelerometer. In both these cases, it is very hard to use without a filter.

Accelerometer measures all forces that are working on an object, as well as it will also see a lot more than just the gravity vector. Every small force working on the object disturbs measurements in a great amount. While working on an actuated system the forces that drive the system will also be visible on the sensor as well. The accelerometer data is reliable in case of a long term, for that reason a "low pass" filter is required.

In the case of Gyro, it is very easy to obtain an accurate data which is not susceptible to external forces. But because of the integration over time, the measurement has a tendency to drift, not returning to level zero when the system went back to its original position. This data is reliable only for a short term; it shows drift on the long term.

The complementary filter provides data both for short and long term from gyroscope and accelerometer respectively. The filter is shown in Figure 2.

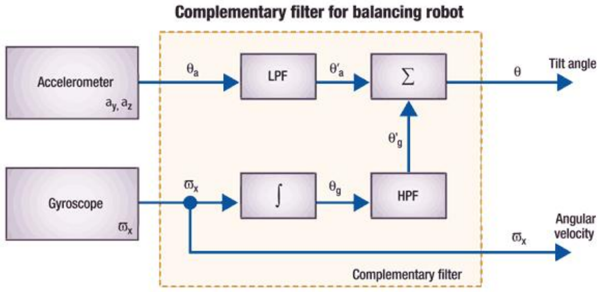

Fig.2: Complementary filter block

Arduino Mega and MPU 6050 are used to acquire data and filter. Basic connection diagram is shown in Figure 3

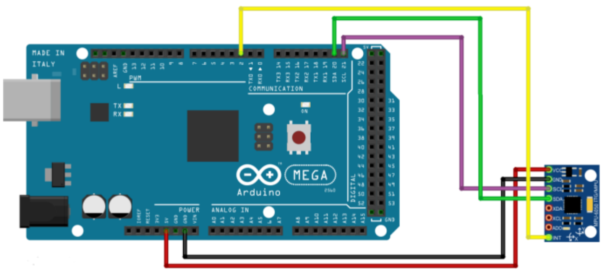

Fig.3: ARDUINO and MPU-6050 connection

5. CONTROL SYSTEM AND ALGORITHM

The control system of the two wheels balancing robot in this project is illustrated by the block diagram in Figure 4.

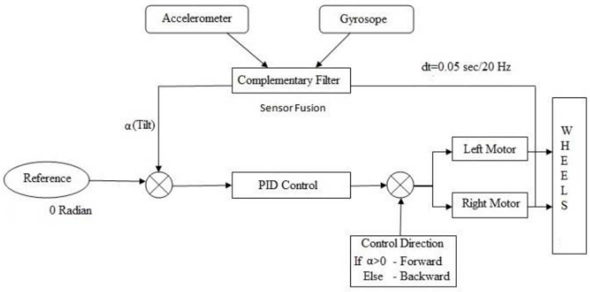

Fig.4: Control system of the robot

The robot control software is inside the microcontroller module and the program flowchart is shown in Figure 5.

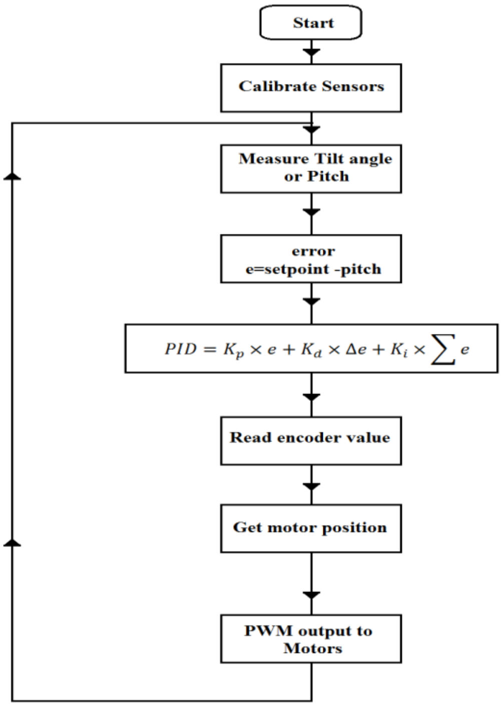

Fig.5: Program flowchart

6. MOTOR CONTROL

The speed of the motor can be able to adjust by the PWM by adjusting the duty cycle and hence the voltage supplied to the motor. Using the PWM method saves the cost of acquiring a Digital to Analogue Converter. Another benefit of using the PWM is that the signal remains digital and no digital-to-analogue conversion is necessary, by doing so the noise effects are minimized.

Changing the set point for the PID controller individually for two motors can control the translational motion of the robot. The motor power increases by the proportional term as the system leans further over and decreases the motor power as the system approaches the upright position [5] [6]. A gain factor, K_p, determines how much power to apply to the motor for any given lean, as follows:

$$Proportional\ Term\ =\ Kp\ *\ Error$$

The differential term of the PID algorithm acts as a damper reducing oscillation. Another gain factor, Kd, determines how much power is applied to the motor according to the following equation:

$$Differential\ Term = Kd\ *\ (Error - Last\ Error)$$

Finally, neither the proportional nor differential terms of the algorithm will remove all of the lean because both terms go to zero as the orientation of the system settles near vertical. The integral term sums the accumulated error tries to drive the lean to zero as follows:

$$Output\ Integral\ Term\ =\ Ki\ *\ (Sum\ of\ Error)$$

The output of the PID controller is

$$\begin{aligned} PWM\ =\ Proportional\ Term + Integral\ Term \\ + Differential\ Term \end{aligned}$$

The output of the Motor PWM as above will be used as the set-point for the motor.

$$\begin{aligned} &Error\ of\ Motor\ Speed \\ &= Motor\ setpoint - Current\ Speed\ Reading\ of\ Motor \end{aligned}$$

$$\begin{aligned} &Output\ Proportional\ Term\ of\ Motor \\ &\qquad = Kp\ *\ Error\ of\ Motor\ Speed \end{aligned}$$

$$\begin{aligned} &Output\ Differential\ Term\ of\ Motor\ =\ Kd\ * \\ &(Error\ of\ Motor\ Speed - Last\ Error\ of\ Motor\ Speed) \end{aligned}$$

$$\begin{aligned} &Output\ Integral\ Term\ of\ Motor \\ &\qquad = Ki \\ &\qquad * Sum\ of\ Error\ for\ Motor\ Speed \end{aligned}$$

And finally,

$$\begin{aligned} Motor\ Speed\ =\ &Proportional\ Term\ of\ Motor \\ &+ Differential\ Term\ of\ Motor \\ &+ Integral\ Term\ of\ Motor \end{aligned}$$

For tuning the PID control of motor speed, the value of K_p, K_i and K_d is get by trial and error method.

The H-Bridge amplifier amplifies the signal from the pulse-width modulator channel to produce voltage that is sufficient enough to drive the motor. Under the command of the software the H-Bridge swaps the motor terminals to drive the motor in a different direction. Using the H-Bridge saves the cost of using two voltage sources for bidirectional control of the motor. The H-Bridge has got two logic inputs for the direction control of the motor.

7. EXPERIMENTAL DATA

7.1 Variation in Tilt Angle

For analyzing the response of the experimental setup, real-time data of the sensor was imported to MATLAB in various conditions. A single PID computation for both the motors rather than having separate computations for two motors solves issues regarding a delay due to serial communication which has a considerable effect on the response of the setup [7]. This implementation, however, doesn’t enable any rotational move along the vertical axis to be taken but to balance in only one translational direction. Variation in tilt angle is shown in Figure 6.

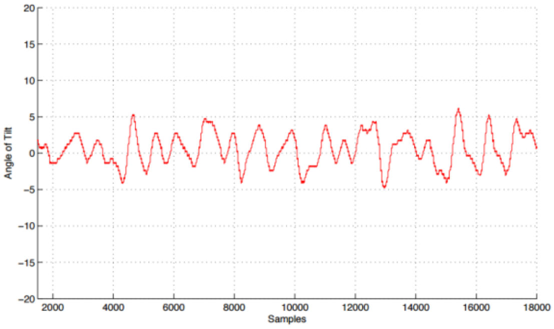

Fig.6: Variation in tilt angle K_p= 0.85, K_i= 3.2, K_d= 0.1(Balancing Condition)

As seen from figure 6, there is a variation of ±5° while it tries to balance about equilibrium position for K_p= 0.85, K_i= 3.2, K_d= 0.1.

7.2 Maximum Angle of Tilt

To determine the maximum angle of tilt beyond which the system will not be able to come back to stable position, Impulsive external forces were provided various times and plotted. As a result of these experiments, it balances many a times, but sometimes not. The maximum angle in any of the plots would determine the desired threshold angle. The maximum angle is approximately 12 degrees. This is shown in Figure 7.

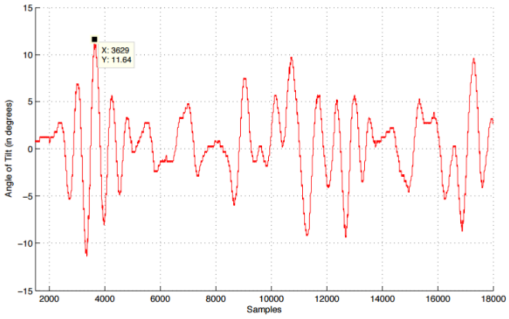

Fig.7: Maximum angle of tilt to remain in stable position.

7.3 Position Drift

In order to quantify the position drift while the setup tries to balance, RPM that is sent to motors in the form of PWM was acquired onto the computer software and integrated. However, there is an inherent assumption here that PWM signals sent to motors are directly proportional to the RPM of the motor. Position drift varies every time the experiment is repeated.

7.4 PID Controller Tuning

For tuning the PID control of motor speed, the value of K_p, K_i and K_d assumed by trial and error method. Although this method is not efficient enough, it can control the speed well. The robot starts to oscillate (move back and forth) about the balance position by setting K_p and K_d term as zero initially and then adjusting K_p. K_p set large enough for the robot to move. With K_p set, the robot accelerates faster when off balance by increasing K_i. The robot is able to self-balance for a few seconds if Kp and ki properly tuned. Finally, the robot would move about its balanced position more gentle by increasing K_d. In this condition no significant overshoots found. PID parameters affect the system dynamics and Effect of parameters over four major characteristics of the closed-loop step response are given in Table 1.

Table 1: Effects of increasing each of the controller parameters K_p, K_i and K_d

Response	Rise Time	Overshoot	Setting Time	Steady State Error
Kp	Decrease	Increase	Minor change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Minor change	Decrease	Decrease	Minor change

Steps for designing a PID controller are:

• Determine what characteristics of the system needs to be improved

• Use K_p, K_d, K_i to decrease the rise time, reduce the overshoot and settling time, eliminate the steady-state error respectively

7.5 TRANSLATIONAL MOTION CONTROL

Changing the set-point for the PID controller individually for two motors can control the translational motion of the robot. If we change one while keeping the other constant, the robot will go into rotational motion as well. Higher the difference between the set point for it to vertically balance and actual set point, faster will the translational motion be.

8. FINAL IMPLEMENTATION

The final implementation of the self balancing robot is figured bellow. As mentioned earlier it is a static linear model. It can handle tilt angle maximum 12 degree. The robot moves forward and backward continuously within a range of about -3 cm and +3cm around the balancing spot in order to balance itself on the flat surfaces. Compared with the smooth flat surfaces, the stability of the robot is better when balancing on the rough flat surfaces like carpet. This can be explained by the relatively small moment of inertia of the wheels compared to the moment of inertia of the robot’s body and the extra damping effect produced by the carpet. The final implementation is shown in Figure 8.

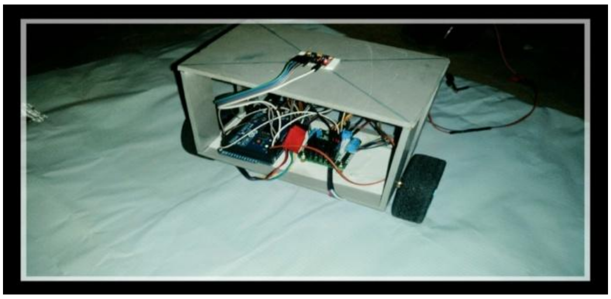

Fig.8: Real time model of the self balancing robot

9. CONCLUSION

A self balancing robot was designed and established as desired with limited resources possible. It was able to balance smoothly with a maximum tilt error of 5 degrees. The robot is capable of taking payloads around 0.3 Kg. Maximum angle of tilt for balancing was also determined through various experiments. However there are some limitations. The robot is unable to balance itself at upright position without significant movement. It means the robot has to move forward and backward continuously within a range of -3 cm and +3cm around the balancing spot in order to balance itself on the flat surfaces. Compared with the smooth flat surfaces, the stability of the robot is better when balancing on the rough flat surfaces like carpet. This can be explained by the relatively small moment of inertia of the wheels compared to the moment of inertia of the robot’s body and the extra damping effect produced by the carpet. Besides, the height of the robot in this project is limited by the torque of the DC motor. The height of the robot should be higher so as to reduce the angular rate of the robot. To overcome this problem, the DC motor with higher torque can be used as the robot actuator. Such technology is suitable only for flat ground. In order to make it work on slant surface, the angle of slant needs to be fed into the system manually or using intelligence.

10. REFERENCES

[1] Liu Kun, Bai Ming, Ni Yuhua, "Two-wheel self-balanced car based on Kalman filtering and PID algorithm" 2011 IEEE 18Th International Conference on Industrial Engineering and Engineering Management (IE&EM), vol.Part 1, Sept. 2011.

[2] N. M. Abdul Ghani, N. I. Mat Yatim, N. A. Azmi: "Comparative Assessment for Two Wheels Inverted Pendulum Mobile Robot Using Robust Control", International Conference on Control, Automation and System 2010, KINTEX, Gyeonggi-do, Korea, Oct 27-30, 2010.

[3] Nawawi, S. W., Ahmad, M. N. and Osman, J. H. S (2008) "Real-Time Control of Two-Wheeled Inverted Pendulum Mobile Robot" World Academy of Science, Engineering and Technology 29, 214 - 220.

[4] Lei Guo , Hong Wang , “PID controller design for output PDFs of stochastic systems using linear matrix inequalities”, IEEE Transactions on Systems, Man, and Cybernetics, Part B 01/2005; 35:65-71,pp.65-71

[5] Mrs.Lekshmy.S, Aleesha George, Athira C.V, “Self Balancing Robot” International Journal of Computer Engineering in Research Trends, Volume 2, Issue 12, December-2015, pp.1091-1095

[6] Sulabh Kumra, Shipa Mehta, “ Singular Axis Self Balancing Robot”, International Journal of Image Processing and Vision Sciences (IJIPVS) ISSN (Print): 2278 – 1110, Vol-1 Iss-3,4, 2012

[7] R.S.Meena,Vikas Kumawat, “Controller Design For Servo Motor Using MATLAB,” .In Proceeding of National Conferences on Advances & Research in Electrical System Technology (AREST 2011)