Self balancing Robot using Complementary filter

Implementation and analysis of Complementary filter on SBR

[1]Kartik Madhira, [2]Ammar Gandhi, [3]Aneesha Gujral
Instrumentation and Control Engineering department,
Institute of Technology, Nirma University
Ahmedabad, India
[1]13bic020@nirmauni.ac.in,[2]13bic014@nirmauni.ac.in, [3]13bic005@nirmauni.ac.in

***Abstract*—The Self balancing robot is based on the inverted pendulum concept, wherein an inverted pendulum is positioned on a cart and the cart is allowed to move on the horizontal axis so as to keep the pendulum in the upright position. This is a classic case of an unstable system. The angle measurement with the help of a fusion of gyroscope and accelerometer requires filtering mechanism as both provide erroneous angle results. Kalman filter is one such filter, but the design and implementation of such a filter is lengthy, tiresome and difficult to implement on smaller 8-bit micro controllers. Thus, this paper intends to design and implement a Self balancing robot with the help of a complementary filter and its analysis using different filter coefficients using PID algorithm as the control strategy. The robot is powered with a lithium-polymer battery to drive the motors.**

Keywords—Self balancing robot; inverted pendulum; complementary filter; PID

I. INTRODUCTION

The self balancing robot (SBR), is a classical example of the inverted pendulum concept. The inverted pendulum in the simplest form, consists of a cart driven by two DC motors, dynamically, to control the position of the inverted pendulum tending to rotate about a fixed position on the cart. This system is a non-linear unstable system as we will see in the later topics. The mathematical analysis is well researched in [1] and [2].

The primary requirement for the control of the system requires angle measurement of the pendulum with respect to the vertical axis. A stable angle position is fixed as a set point and the most basic strategy is to drive the DC motors backward or forward so as to keep the inverted pendulum in the stable angle position, this system can be easily be seen in Segway system[13]

The angle measurement sensor deployed is an Inertial Measurement Unit (IMU), MPU-6050. The angle can be measured individually each by deploying an accelerometer by finding the angle between different forces acting on the pendulum or, alternatively, by using a gyroscope and integrating the angular velocity over time to get the angular measure of pendulum.

This method contains erroneous data as, Firstly, the accelerometer is sensitive to all the 'unwanted' forces acting on the system like vibrations from the DC motors, friction on the wheels and all the other forces except gravitational force. The resulting angle thus measured contains error and the error fades only after a certain period of time after the vibrations and other forces have stabilized or come to a halt. Secondly, the angle measurement from the gyroscope requires integration of the angular velocity to get angular data but even a small error in the measurement of the angular velocity will cause the measurement to drift away from the actual measurement in the long run as error gets integrated over time. Thus, the accelerometer is error free only after a certain period of time and the gyroscope reading is error free only in the initial period [3].

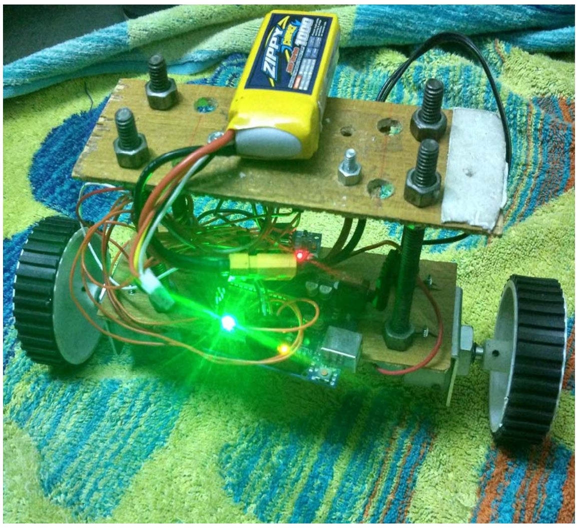


Figure 1: SBR in action

The error in both these strategies paves way for the deployment of a filter so as to remove or 'filter' away the errors by combining the measurement from both the accelerometer and gyroscope. One such strategy is of the Kalman filter, which uses mathematical estimation from the two measurement to produce an error free output [4]. This method is very cumbersome and difficult to understand as seen in [14].

978-1-4673-9939-5/16/$31.00 ©2016 IEEE

Though mathematically proven, it is best available filter to use as seen in the research paper [14], it is very difficult to integrate this technique on smaller microcontrollers (8-bit). Thus, a more simpler but effective filtering strategy is of the Complementary filter and employs low pass and high pass filter to arrive at a nearly error free angle measure. The primary focus of this paper is for the implementation of this filter in the SBR and the effect the coefficients have on the working of the SBR.

Finally, the control strategy for obtaining the stable angle position is the Proportional-Integral-Derivative (PID) control method was applied after getting inputs from researches in [5, 6, 15].

II. INVERTED PENDULUM MODEL

In the following analysis of the pendulum cart system by Siebert [6], we mathematically obtain the transfer function of the pendulum-cart system.

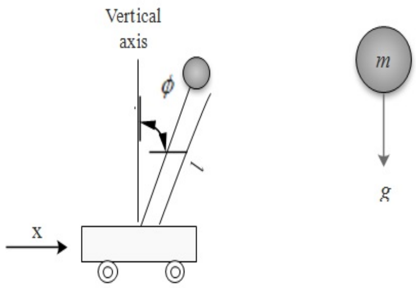


Figure 1: A cart and pendulum system

The torque produced about the pivoting point at an angle ϕ due to gravitational acceleration, *g*.

$$\mathrm{T}=\mathrm{mglsin}(\phi) \tag{1}$$

Based on (1), we derive the following two equations for the vertical and horizontal rotational accelerations

$$\ddot{\phi}_g = (\frac{g}{l})\sin(\phi) \tag{2}$$

$$\ddot{\phi}_x = (\frac{\ddot{x}}{l})\cos(\phi) \tag{3}$$

From (2) and (3), we combine them to get the final rotational acceleration acting on the system.

$$\ddot{\phi} = \ddot{\phi}_g + \ddot{\phi}_x = (\frac{g}{l})\sin(\phi) - (\frac{\ddot{x}}{l})\cos(\phi) \tag{4}$$

$$l\ddot{\phi} - g\phi = -\ddot{x} \tag{5}$$

Finally applying Laplace operator to (5), get,

$$\frac{\phi(s)}{X(s)} = \frac{-s^2}{ls^2 - g} = \frac{\frac{-s^2}{g}}{(\tau s + 1)(\tau s - 1)}$$

III. MECHANICAL STRUCTURE

The robot comprises of two wooden platforms kept one above the other and supported by four bolts. The upper side of the top floor platform contains the Li-ion battery pack while the lower side contains the motor driver circuit.

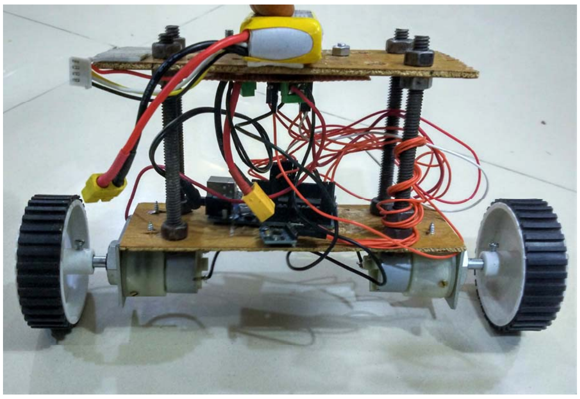


Figure 2: front view-structure

The lower floor consists of the 8-bit microcontroller Arduino Uno, the IMU sensor and two 60 rpm DC motor attached to the bottom of the platform. The IMU sensor was kept at lower platform so as to keep it as immune to vibrations as possible. The wheels used were generic robot wheels.

IV. ELECTRONICS

The following electronic items were deployed in the robot

1. An Arduino Uno microcontroller was used for faster deployment of algorithm and ease of debugging capabilities and sensor interfacing is easier to implement with less code footprint.
2. The IMU sensor, MPU-6050, MEMS based, manufactured by InvenSense technologies was used. The module consists of 3-accelerometers and 3-Gyroscopes making the unit a 6 DOF (Degrees of freedom) sensor. The sensor works at a voltage level of 3.3v and I2C protocol as means of communication with the microcontroller.

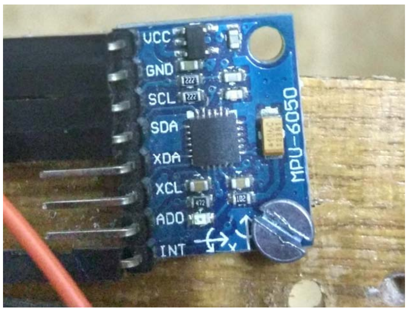


Figure 2: MPU6050 sensor on lower deck

3. To drive the motors, L293 motor driver was used with PWM input from Arduino being given to the enable pins of both channels.
4. A 100 rpm DC geared motor with a max torque of 1.2kg-cm was used to counter the rotational torque of the robot.
5. Finally, a Li-Po battery capable of providing 11.1v with 1000 mAh as used to power the overall system.

V. BLOCK DIAGRAM

From the block diagram in figure 4 as we can see, the stable angle of 180 degrees as the reference set point to the controller. But, due to a dead zone present in the wheels in the motor, practically 177 degrees was taken as the set point. At the summation point is the filtered angle from both accelerometer and gyroscope. This is the real time angle that the system is tilted to with respect to vertical axis. The difference in the actual angle and the output angle is fed into the controller. The practical implementation occurs on taking the robot angle from the IMU sensor, and giving respective output from the PID controller to the motor in terms of the PWM signals by the controller.

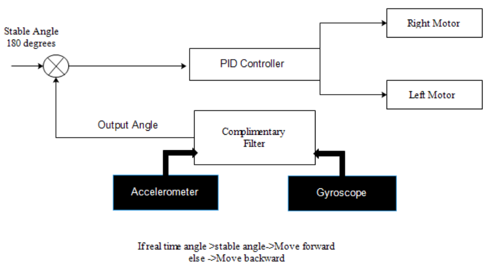


Figure 3: Block Diagram SBR.

The control strategy chosen was PID algorithm. The reason for applying PID algorithm was in context with the simplicity in implementation yet most accurate in comparison, in getting the desired set points.

To move the robot in forward direction, which is triggered by a dedicated pushbutton on the robot, the robot is given a set point greater than the stable angle (178.5°), and thus the overall PID control work with this angle as a set-point and as a result, the robot tends to move forward. Same strategy is applied in moving the robot backward by giving it a set point of 176.2°.

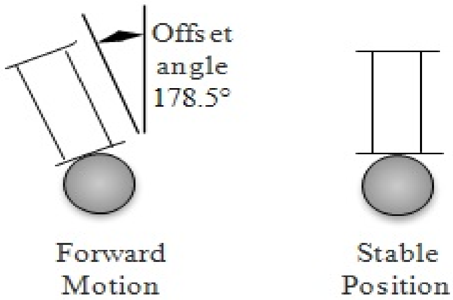


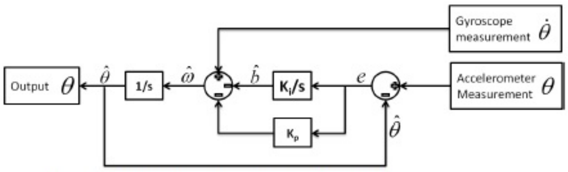
Figure 4: Forward motion process

VI. COMPLEMENTARY FILTER

The need for an alternative to Kalman filter arises from the fact that the Kalman filter is very cumbersome, difficult to understand and challenging to implement on a smaller 8-bit microcontroller. Thus a complementary filter serves this purpose of simplifying the difficulties faced while implementing simple first order high pass and low pass filters.

Complementary filters do not act on the signals. They are in fact filters to be associated with noise of the signal present. Thus the task of the filter is to estimate a stable angle from multiple sources containing erroneous data and which exhibit noise with different frequency content [8].

The complementary filter is a frequency domain filter. Mathematically, it can be seen as the use of two or more transfer functions which are complements of each other. Thus, if one data from the first sensor is represented by the transfer function G(s), then the other data from the second sensor is I-G(s) with the sum being the identity matrix I [8]. If the high frequency noise from one sensor and the low frequency from the second one are compliments of each other then, the complementary filter is applicable. The signal is passed through both high and low pass filter to remove the noise and final reconstruction takes place.



θ = measured angle of accelerometer
$\dot{\theta}$ = measured angular velocity of gyroscope
$\hat{\theta}$ = estimated angle of balancing robot
e = estimation error between $\hat{\theta}$ and θ measurement
$\hat{b}$ = estimated gyro bias
$\hat{\omega}$ = estimated angular velocity
k_p = proportional gain
k_i = integral gain

Figure 5: Complementary filter block diagram [3]

In theory the Kalman filter should provide the best output and does not work in the frequency domain. But on the other hand the practical implementation of complementary filter does not vary much or rather is almost same as the kalman filter. What makes the difference in the two is the implementation.

A. *Complementary filter implementation on Arduino*

The software implementation of the complementary is very easy with the use of just a digital high pass and a low pass filter. The filter consists of two filter coefficients and the following is the complementary filter equation

FAngle = c*(FAngle+GyroA*dt) + (1-c)*AccA

Where,
FAngle= Final filtered angle
c= High pass filter coefficient
GyroA=Angle obtained from the gyroscope
AccA=Angle obtained from acceleration
dt=loop time since last iteration

Thus, a simple one line code is enough for complementary filter to be implemented and won't exacerbate the controller efficiency which is limited to only 8- bit processing power.

```
accXangle = (atan2(accY,accZ)+PI)*RAD_TO_DEG;
accYangle = (atan2(accX,accZ)+PI)*RAD_TO_DEG;

double gyroXrate = (double)gyroX/131.0;
double gyroYrate = -((double)gyroY/131.0);
gyroXangle += gyroXrate*((double)(micros()-timer)/1000000);
gyroYangle += gyroYrate*((double)(micros()-timer)/1000000);

angle=0.98*(angle+gyroXrate*dt)+0.02*(accXangle);
```

Figure 5: Code snippet from Arduino IDE

VII. PRACTICAL RESULTS

The following results were obtained for the PWM output from the PID algorithm and the corresponding angle measurement from the Arduino IDE serial monitoring window. The PWM signal for the angles were obtained from a Digital Oscilloscope.

1) *For High pass filter coefficient c=0.98*

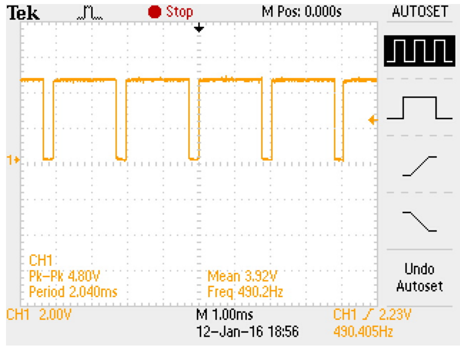


Figure 7: PWM output

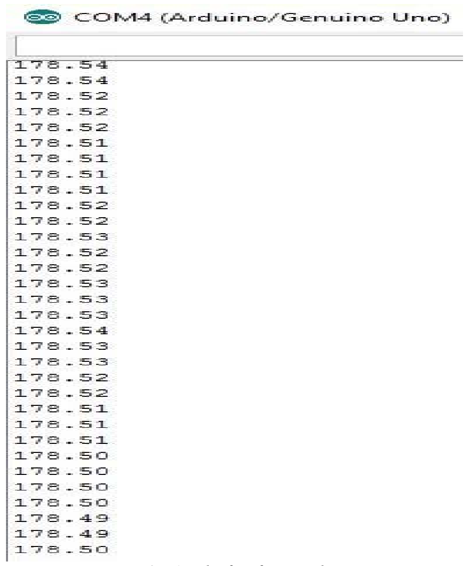


Figure 8: Angle for forward motion

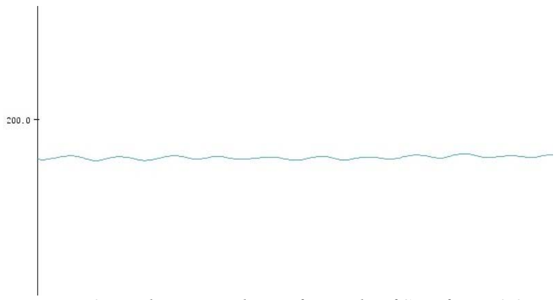


Figure 9: Real time simulation for angle of SBR for c=0.94

2) *For High pass filter coefficient, c=0.94*

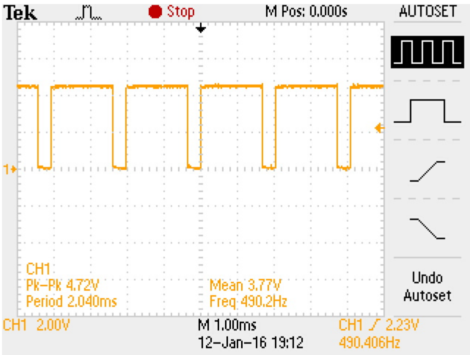


Figure 10: PWM output for c=0.94

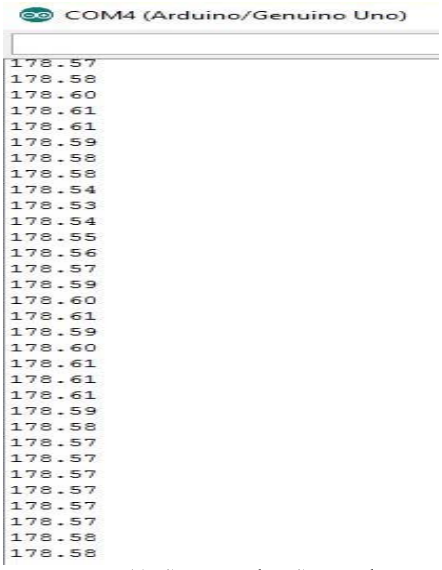


Figure 11: Corresponding SBR angle

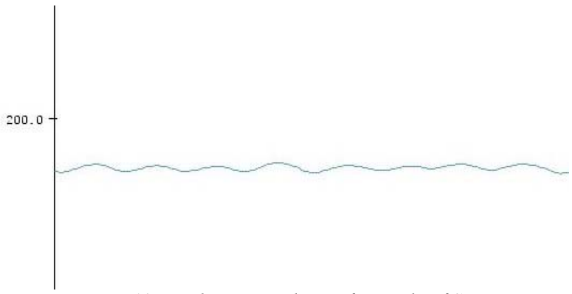


Figure 12: Real time simulation for angle of SBR for c=0.94

VIII. CONCLUSION

This research work was conducted with low cost resources and was aimed at using less computation power by devising software implementation of complementary filter and thereby developing a working model of the same. This research can be extended with future works where we intend to use NEMA17 3.6kgcm torque Stepper Motor and Image Processing Capabilities.

REFERENCES

[1] James K. Roberge. The mechanical seal. Bachelor's thesis, Massachusetts Institute of Technology, May 1960.

[2] Katsuhiko Ogata. Modern Control Engineering, pages 277–279. Prentice-Hall, Englewood Cliffs, New Jersey, 1970

[3] Hau-Shiue Juang and Kai-Yew Lum, "Design and Control of a Two-Wheel Self Balancing Robot using the Arduino Microcontroller Board", 2013 10th IEEE International Conference on Control and Automation (ICCA),June 12-14,2013.

[4] J. Alberto Bonales Valencia, J. Juan Rinc´on Pasaye and Renato Gonzalez Bernal, "Instrumentation and Wireless Control for the Self-Balancing Mobile robots on Two Wheels", IEEE online journal,2014

[5] Liang Sun and Jiafei Gan , "Researching Of Two-Wheeled Self-Balancing Robot Base On LQR Combined With PID", IEEE online journal,2010

[6] Tim Wescott," PID without a PhD", Embedded Systems Programming, 2000, 86-108.

[7] William McC. Siebert. Circuits, Signals, and Systems, pages 177–182. MIT Press, Cambridge, Massachusetts, 1986.

[8] Walter T Higgins,JR, "A Comparison of Complementary and Kalman Filtering", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES- 11, No.3 , 1975

[9] Hyung Gi Min and Eun Tae Jeung, " Complementary Filter Design for Angle Estimation using MEMS Accelerometer and Gyroscope"

[10] Brisilla R.M, Sankaranarayan V. "Nonlinear control of mobile inverted pendulum", Robotics and Autonomous Systems Elsevier,Volume 70, August 2015, Pages 145-155

[11] Fast PWM on Arduino - Arduino Playground

[12] InvenSense, MPU-6050 Datasheet, Nov. 2010 [Revised Sept 2013]

[13] Segway Inc, [http://www.segway.com]

[14] Fengxin Sun, Zhen Yu, Haijiao Yang, " A Design for Two-Wheeled Self-Balancing Robot Based on Kalman Filter and LQR" , International Conference on Mechatronics and Control, July 2014

[15] Yulei Gong, Xiao Wu, Huijaio Ma, "Research on Control Stratergy of Two-Wheeled Self-Balancing Robot", International Conference on Computer Science and Mechanical Automation, 2015