Design and Implementation of a 3-Dimensional Attitudes Estimator Device using Low Cost Accelerometer & Gyroscope with Microcontroller IDE

M. RAJA*[,1,a], Ugur GUVEN[1,b], Kartikay SINGH[1,c]

*Corresponding author
[1]Aerospace Department, University of Petroleum and Energy Studies,
Bidholi, Dehradun -248007, Uttarakhand, India,
mraja@ddn.upes.ac.in*, drguven@live.com, kartikaysingh1297@gmail.com

DOI: 10.13111/2066-8201.2020.12.2.13

Received: 18 October 2019/ Accepted: 30 March 2020/ Published: June 2020

Abstract: *Navigation and guidance systems for most automobile as well as aerospace applications require a coupled chip setup known as Inertial Measurement Units (IMU) which, depending on the degree of freedoms, contains a Gyroscope (for maintaining orientation and angular velocity), Accelerometers (to determine acceleration in the respective direction) and a Magnetometer (to determine the respective magnetic fields). In the three-dimensional space, any required rotation analysis is limited to the coordinate systems and all subtended angles in either direction must be defined by a fixed axis to effectively estimate the stability and to define all the attitude estimates needed to compile different rotations and orientations. The Quaternions are mathematical notations used for defining rotations and orientation in three-dimensional space. The simplest terms Quaternions are impossible to visualize in a three-dimensional space; the first three terms will be identical to the coordinate system, but through Quaternions another vector quantity is added into the equations, which may in fact underline how we can account for all rotational quantities. The fundamental analysis of these components different applications for various fields is proposed.*

Key Words: *Quaternions, Inertial Measurement Units (IMU), Gyroscope, Accelerometers*

1. INTRODUCTION

In recent years, the use of Inertial Sensors is less expensive and due to their compact dimensions they are highly used for obtaining high measurement values for position and orientation. Even though these values are not sturdy as they experience the integration drift over time due to large time scales, on the other hand they are highly accurate over a short time scale. These errors due to gyroscopic drift can be approximately of deg/sec and may cause system instability.

Thus, to overcome these high drifts, the Inertial sensors are combined with some additional sensors and by using different set of algorithms to correct the error from one input to another we can reduce propagation errors [1]. In Aerospace applications, accelerometers and gyroscopes are often coupled into an IMU (Inertial Measurement Unit). Figure (1) shows the

[a]Assistant Professor Dr
[b]Professor Dr
[c]Student

coordinate frames such as navigation frame (x_n, y_n, z_n), earth frame (x_e, y_e, z_e), and inertial frame (x_i, y_i, z_i). Figure (2) shows the attitude variation in aircraft [2]. The reference frame is referred as coordinate frame mainly used to find the position and orientations of the body. In the proposed research work, an inertial sensor is used to combine the results of 3-axis gyroscope and accelerometer [3]. The gyroscope is used to measure the rate of change of angles (Pitch rate, Yaw, Roll) p, q, r and the accelerometer is used to measure the specific force with adding the gravity, and then get the linear acceleration of the vehicle.

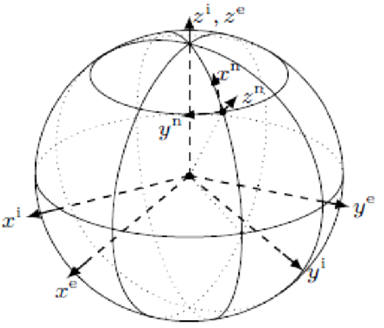


Fig. 1 Coordinate Frame Reference

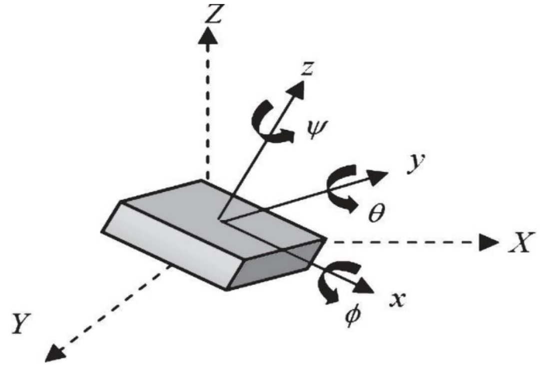


Fig. 2 Yaw Pitch Roll Variation w.r.t to any object

The inertial measurement unit measures the angular velocity with respect to the body frame. The coordinate frame is shown in figure (1); the inertial frame is referred as 'i' the earth frame as 'e' and the navigation frame as 'n' [4].

2. LITERATURE REVIEW

In the research paper, "An efficient orientation filter for inertial /magnetic sensor arrays", the author Sebastian O. H. discusses the fundamental approach to this project covering topics ranging from quaternions and algorithms merging to offsetting the gain acting on the sensor itself; everything has been taken into account. The comparative algorithm between the Kalman filters was measured in terms of sampling rate performance and filter gain performance. An improvement of the device undergoes higher computational loads due to static rate errors. John L. Crassidis, discussed the reference frames and different filters that have been deployed to more stressful situations such as satellite orientation problems [3].

Harris Teague explains a relatively simple complementary filter that combines on-board accelerometers, gyroscopes, and magnetometer sensing [4]. This paper explores the limits of performance of such attitude estimation, with a focus on performance in highly dynamic maneuvers.

This paper focuses more on comparing estimator performance for a baseline set of algorithms in challenging conditions rather than finding the overall limits of performance. Thus, additional sensing sources such as computer vision may significantly change the overall performance of all algorithms. Further, it is expected that a more sophisticated algorithm for adjusting filter gains/ weights than the simple dynamics detector of Section, also provides significant benefits in performance. A goal is to provide a concise development of these nonlinear attitude estimation techniques such that the reader can implement working algorithms [5], [6]. The UKF estimates the state covariance and state-measurement cross covariance during the estimation process and does not rely on gradients that as assumed to be known.

3. METHODOLOGY

The unit vectors are considered $\widehat{e_{x1}}$; $\widehat{e_{y1}}$; $\widehat{e_{z1}}$ the Earth centered and Earth fixes coordinate frame. Let

$$\boldsymbol{\omega} = \boldsymbol{\omega} BB/E = [p, q, r] \quad (1)$$

The angular velocity is measured in the aircraft body frame to the Euler angles. The Euler angles rotation matrix is [1 sin ϕ tan θ cos ϕ tan θ; 0 cos ϕ − sin ϕ; 0 sin ϕ/cos θ cos ϕ/cos θ]. The rate of change of unit vector is

$$d\boldsymbol{e}\hat{}idt = \boldsymbol{\omega} E/B \times \boldsymbol{e}\hat{}\boldsymbol{i} = -\boldsymbol{\omega} \times \boldsymbol{e}\hat{}\boldsymbol{i} = [\boldsymbol{\omega}]\, T \times \boldsymbol{e}\hat{} \quad (2)$$

where, the angular velocity is [0 −ω3 ω2; ω3 0 – ω1; −ω2 ω1 0].
The attitudes of vehicle is =

$$\begin{Vmatrix} c(pitch)\, c(yaw) & c(pitch)\, s(yaw) & -s(pitch) \\ s(roll)s(pitch)c(yaw) & s(roll)s(pitch)s(yaw) + c(roll)c(yaw) & s(yaw)c(pitch) \\ c(roll)s(pitch)c(yaw) & c(roll)s(pitch)s(yaw) & c(roll)c(pitch \end{Vmatrix}$$

c – cosine & s – sine terms

The following algorithms are used:

To calibrate the sensor itself aligning it to the factory setting itself.

- The I^2C bus is used to attain the values from the sensor itself and manipulating the data to find out our required method of representation [7], [8].

A. Data Acquisition Algorithm: Code Breakdown

1. Start
2. Include the I^2C bus header file (inter integrated circuit protocol allows slave DIC circuit to communicate with the master circuit.
3. Creating MPU object to develop an interface between the master slave and serial monitor.
4. Choosing an output measurable function.
5. Interrupt pin definition so that processing of different source codes can be implemented (internal).
6. Defining a digital interrupt for a readable form and defining Boolean values for these readable form via LEDs.
7. Using unsigned integer values to define different status, counting values, and defining the default value of the FIFO storage buffer.
8. Defining container values for quaternions and vector quantities for different frames of references and Euler angles and yaw, pitch and roll conversion sets.
9. Detect interrupt state via the digital interrupt on the board itself to verify interrupt control connections.
10. Interrupt pin gets high so that the program can be run through.
11. Calling wire. h to begin communication.
12. Setting up a baud rate since accuracy is the foremost priority thus setting up a high baud rate for maximum bit rate per second is essential.
13. Calling in the MPU protocols to ensure proper connection.
14. Using the function pin-mode to set the interrupt as the primary input.
15. Tell the user that the connection is set.
16. Using the MPU library to test connection and display it to the user [9].
17. Adding a default value to the empty buffer to initialize the whole data set.

Fig. 3 Data Acquisition Algorithm

18. Calling wire. h to begin communication.
19. Setting up a baud rate since accuracy is the foremost priority thus setting up a high baud rate for maximum bit rate per second is essential.
20. Calling in the MPU protocols to ensure proper connection.
21. Using the function pin-mode to set the interrupt as the primary input.
22. Tell the user that the connection is set.
23. Using the MPU library to test connection and display it to the user [9].
24. Adding a default value to the empty buffer to initialize the whole data set.
25. Adding offsets to understand the calibration needed to supply accurate measurements.
26. Receiving different status either sets on or off to understand activity and proper working of the chip and connecting external and internal interrupts.
27. Use the particular output readable form to perform conversion into each of the output display forms.
21. End.

B. Calibration Sketch Algorithm: Code Breakdown

1. Start
2. Import the Required Header files for the I2C, wire communication and the respective measurement unit.
3. Create buffers to accommodate the required elements to take an average and calculate.
4. Display these calibrated values and add a small delay to compensate to make sure there are no repeated measures in the system.

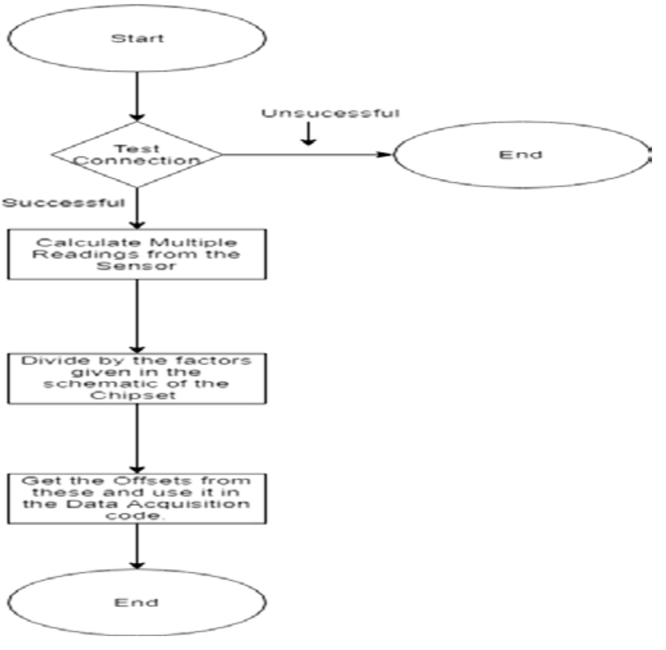


Fig. 4 Data calibration Algorithms

5. Now use the factory defined numbers to find the offsets in each components i.e. the numbers [10].
6. Use these calibrated values to implement simulation sketches, which has called the teapot analysis.
7. End.

C. Compare the Kalman and complimentary filter Algorithm

1. Start
2. Include wire. h header file, kalman. h header file [11].
3. Creating MPU object to develop an interface between the master slave and serial monitor.
4. Create Kalman instances in X and Y directions.
5. Convert raw data into output readable form and set the sampling rate.
6. Measure quality factor biases between the angle and the measurements [12].
7. Determine data from the accelerometer by asking for the addresses and input in an array form.
8. Define rate after getting rate from the next iteration and find the common rate by subtracting from the user-defined bias.
9. Update the error covariance and predict the error covariance ahead.
10. Estimate the next angle and bias and update the measurement.

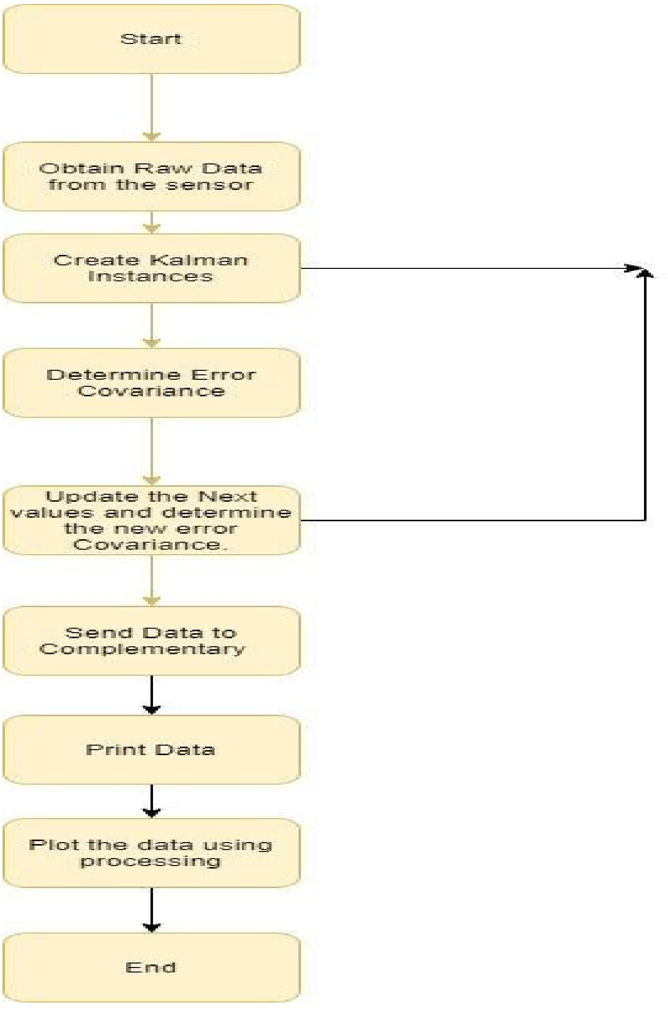


Fig. 5 Kalman Filter Algorithm

11. Return the angle bias and rate to the microcontroller and update in the form of values of X and Y.
12. Use the complimentary filter, mathematical equations defined and update in the form of values of X and Y.
13. Print the data.
14. Plot the graph between amplitude and time for each of these extracted values.
15. End.

D. Algorithm for PID

1. Start
2. Include Wire.h for serial communication and Servo.h for motor operations.
3. Import data from the MPU and convert every data to degrees.
4. Define PID constants obtained from practical run of the model.
5. State an initial value for throttle and begin serial communication.
6. Request for 6 data values from the MPU using the present address. [13]

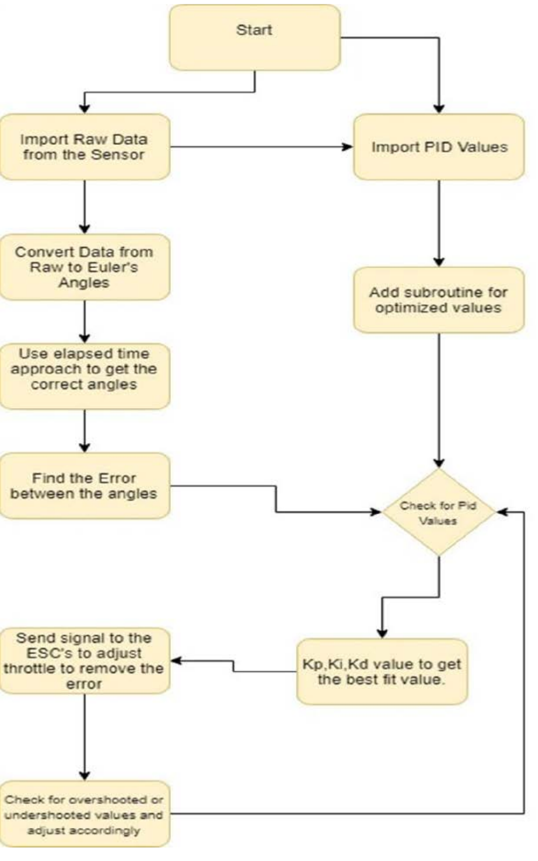


Fig. 6 PID Controller Algorithms

7. Convert every acceleration value into Euler's angles.
8. Find the total angles by using gyro angle, acceleration angle and elapsed time.
9. Find the error after subtracting the total angle from the desired angle.
10. Multiply error into the proportionality constant.
11. Check for the magnitude of the error; if under the permissible limit do not apply integral tuning.
12. Multiply derivative constant into the previous state errors divided by the elapsed time.
13. Add all the PID values.
14. Continually check for the threshold value for the motor at 2000 microseconds and change accordingly to the PWM signal.
15. Send the data to the motor and return the value of previous error back to the loop.
16. End.

4. RESULTS & DISCUSSIONS

The simulations of the instruments have been performed on respective software to provide figures and graphs and the following results have been obtained by selecting the respective output readable form without the use of the barometer. Each of these tests were done in the presence of a meek vibration system attached to the sensor itself to emulate the gyro drift with presence of minimal movements, which thus portray the overall noise present in the chipset installation.

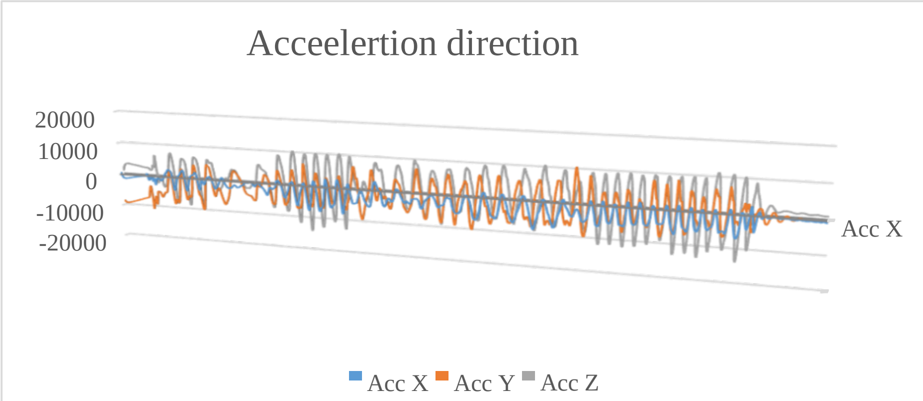


Fig. 7 Acceleration Vs Time with the Scale Factor of 350

The comparative study shows the data variation in X and Y direction.

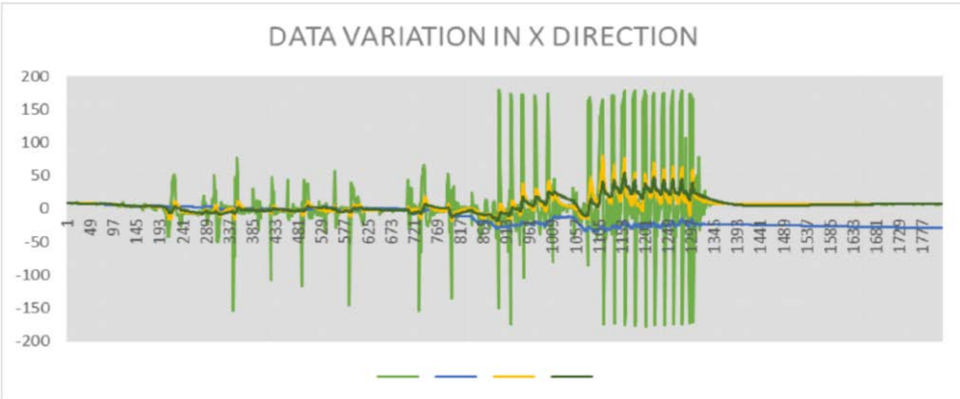


Fig. 8 Graph depicts the data variation in X direction

- Complimentary filter
- Kalman filter
- gyro
- Pitch

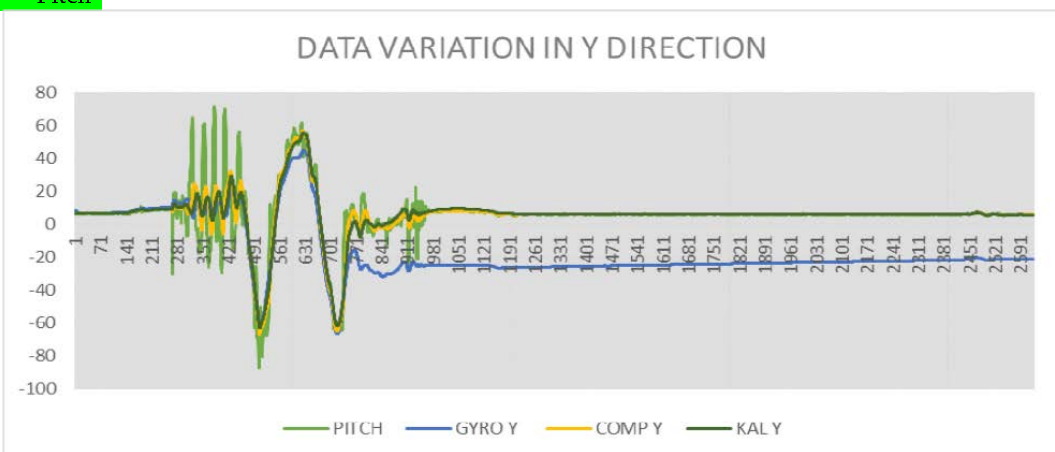


Fig. 9 The graph represents data variation in y direction

PID Controller

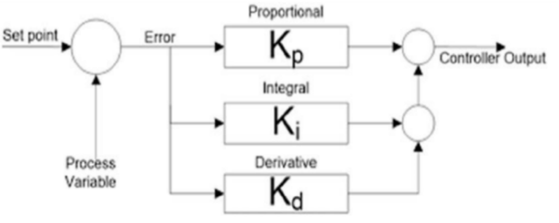
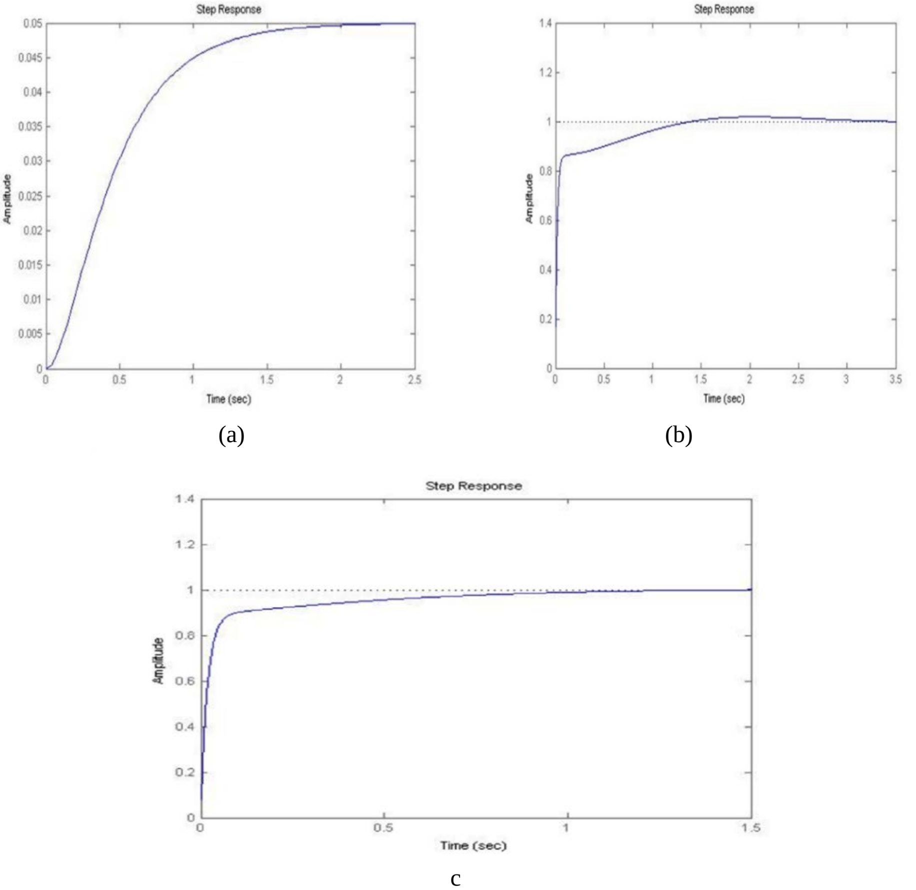


Fig. 10 PID Block Diagram

$$u = k_p e + K_i \int_0^t e dt + K_d \frac{de}{dt} \tag{3}$$

(u- Controlled input; e- Error signal; K_p-Proportional gain; K_i-Integral gain; K_d-Derivative gain)
Figure 10 shows the PID block diagram; it compares the desired input and process variables to provide the error signal. This error signal is to convert the controlled signal with the help of PID controller. The transfer function of PID controller is given below equation (4). The various step responses of the system with controller and without PID controller can be seen from various plots shown in Fig. 11 (a), (b), (c).



(a) (b)

c

Fig. 11 (a) Step responses for ideal case; (b) Step responses with 3.5 seconds; (c) Step responses with 1.5 seconds

In Figure 12 (b), the simulation results show the transient response of the system with large steady state errors compared to figure (c).

5. CONCLUSIONS

The overall drift and vibration due to the complex mechanism of attitude estimation is still far from being complete now and need to use filtration processes such as the Kalman Filter and their types. The following graphs show the noise generated in the chipset just by being in the presence of an actual vibrating source present at a distance of 20cms to eradicate any magnetic field deviations.

Fig. 12 Attitudes vs. Time

We used filtering process to simulate the noise generation and drift produced.

- Higher mathematical models are used for a better plotting efficiency and more values to define the curve.

1) Comparisons of Each of the Filters are used to understand the best algorithm for our application.
2) PID controller is used for reducing the deviation error.

A comparison between the calculated and generated value is shown in the table below (1).

Table 1-Comparative Results between the Calculated Values and the Generated Values

CALCULATED GAIN VALUES			IDEAL GENERATED VALUES			NUMBER OF ITERATIONS
KP	KI	KD	KP	KI	KD	N
0.2	0.0002	0.1	1.2105	0.00032	0.7152	5
0.4	0.0004	0.2	1.5732	0.0004	0.8152	10
0.6	0.0006	0.3	1.9364	0.00048	0.9125	20
0.8	0.0008	0.4	2.1625	0.00049	1.1015	30
1	0.001	0.5	2.3824	0.0005	1.2045	40
1.2	0.0012	0.6	2.9989	0.00052	1.2426	50
1.4	0.0014	0.7	4.2328	0.00075	1.2112	60
1.6	0.0016	0.8	3.9959	0.00084	1.4151	70
1.8	0.0018	0.9	5.5989	0.00156	1.4562	80
2	0.002	1	4.2685	0.00224	1.5246	90
2.2	0.0022	1.1	4.8596	0.00248	1.5987	100
2.4	0.0024	1.2	5.8976	0.00287	1.8452	120
2.6	0.0026	1.3	5.2689	0.00291	1.6789	140
2.8	0.0028	1.4	4.5932	0.00329	1.7485	160
3	0.003	1.5	4.4589	0.00358	1.7526	180
3.2	0.0032	1.6	5.2592	0.00389	1.8965	200
3.4	0.0034	1.7	4.1703	0.00418	1.9125	240
3.6	0.0036	1.8	4.3626	0.0042	1.9265	280
3.8	0.004	2	5.4568	0.00442	1.9762	300

REFERENCES

[1] G. Wahba, A Least Squares Estimate of Spacecraft Attitude, *SIAM Review 1966*, **8**:3, 384-386, *Journal of the Astronautical Sciences*, **38**(3), November 1987.

[2] F. L. Markley, Attitude Determination Using Vector Observations and the Singular Value Decomposition, *The Journal of the Astronautical Sciences*, vol. **36**, no. 3, pp. 245-258, July-September 1988.

[3] S. O. H. Madgwick, An efficient orientation filter for inertial and inertial/ magnetic sensor arrays, *Rehabilitation Robotics (ICORR)*, 2011 IEEE International Conference.

[4] J. L. Crassidis, F. Landis Markley, Yang Cheng, Survey of Nonlinear Attitude Estimation Methods, *Journal of Guidance, Control, and Dynamics*, vol. **30**, No. 1, pp. 12-28, 2007.

[5] H. Rehbindera, X. Hub, Drift-free attitude estimation for accelerated rigid bodies, *Automatica (Journal of IFAC)*, Volume **40**, Issue 4, Pages 653-659, April 2004.

[6] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, *Journal of Process Control*, Volume **14**, Issue 4, Pages 465, June 2004.

[7] H. M. Hasanien, S. M. Muyeen, Affine projection algorithm based adaptive control scheme for operation of variable-speed wind generator, *GTD IET*, vol. **9**, no. 16, pp. 2611-2616, 2015.

[8] S. Agrawal, V. Kumar, K. P. S. Rana, P. Mishra, Optimization of PID controller with first order noise filter, *Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE) 2015 International Conference*, pp. 226-231, 2015.

[9] * * * http://mrpt.sourceforge.net/reference/0.9.6/classmrpt_1_1poses_1_1_c_pose3_d.html (Image yaw pitch roll on an object).

[10]* * * http://www.chrobotics.com/library/understandingeuler-angles (Image yaw pitch roll on an aircraft).

[11]* * * http://www.perfectlogic.com/articles/Avionics/FlightDynamics/FlightPart2.html (Image coordinate frame reference).

[12] * * * https://arxiv.org/pdf/1704.06053.pdf (Introduction: Gyroscope and accelerometer).

[13] * * * https://www.robotshop.com/community/forum/t/kalman-filter-vs-complementary filter/13079 (PID and Filter).