

Mathematical Modelling of a Two Degree of Freedom Platform Using Accelerometers and Gyro Sensors

Vinícius Nunes Lage, Alan Kardek Rêgo Segundo and Thomás Vargas Barsante e Pinto

Control and Automation Engineering and Fundamental Technics Department, Federal University of Ouro Preto, Ouro Preto 35400-000, Brazil

Abstract: This paper demonstrates the assembly of a servo-controlled platform with two degrees of freedom, empirical methods and a developed closed-loop control found in the system mathematical model. This control aims to stabilize and hold small objects on the platform. We parsed the step response in X and Y axes, hence we found the first and second-order models for each one. We did some further analyses to decide which one would better represent the behavior of the system. The MATLAB software provided step response for the model empirically obtained and latter compared it to experimental data acquired in the trials. Accelerometers and gyro sensors from the MPU-6050 sensor measured the angular position of platform on X and Y axes. In order to improve measurements accuracy and eliminate noise effects, we implemented the complementary filter to the firmware system. We used Arduino to control servomotors through PWM pulses and perform data acquisition.

Key words: Balanced platform, MPU6050 sensor, complementary filter, Arduino.

1. Introduction¹

Many systems, nowadays, present stability and position controls. Some modern vehicles use smart suspension to maintain passengers stable while crossing rough terrains. Some drones used for video recording purposes, have cameras installed on their base using smart systems, which named gimbals, they control the camera stability, in order to make the video footage more attractive for viewers. There are also active systems attached to buildings, which mitigate vibrations [1]. Recently, Google acquired a startup company called Levante Lab, which has created the Liftware. It is about a spoon that uses a series of algorithms to stabilize the hand of a person with motor shaking problems. This tool is tremendously useful to people with Parkinson disease. According to Google, this device reduced food drop during tests in 76% [2]. For the effective development of a control system, identifying mathematical model of the mechanism is essential. The mechanical system consists of different

parts, each one with many distinct properties to perform measurements. Thus, calculation of these mathematical models may become very complex.

Calculating complex dynamic system models mathematically are hard tasks. However, there are practical methods to do that without the need of studying each part separately, once these methods study the system entirely. These methods are practical and applicable techniques, which calculate efficiently the real dynamics of the system through theoretical models [3].

This work aimed to develop a mathematical model of a two-degree-of-freedom platform by means of practical experiments and evaluation of their results. Section 2 presents the theory and calculations behind devices used; Section 3 shows materials and methods of analysis used in the experiments; Section 4 presents results obtained; finally, Section 5 contains the conclusion for this work.

2. Theory

2.1 Accelerometer and Gyro Sensors

There are two types of position sensors: one to

Corresponding author: Thomás Vargas Barsante e Pinto, bachelor student, research field: use of servomotor to control movements.

measure linear movement of an object and another to measure the angular shift [4]. For this work, we used a sensor of the second type. The integrated circuit MPU-6050 is a sensor manufactured by IvenSense Inc. Enterprise. It contains three gyroscope sensors (one for each of the X, Y and Z axes), three accelerometers (one for each axis as well) and a DMP (Digital Motion Processor) in a single chip [5]. It uses an I2C communication for data transmission and internally has three 16 bits analogical/digital converters, used for accelerometers and gyro sensors data discretization. The sensor has good accuracy for either fast or slow movements. Scales for the gyro sensor are of ± 250 , ± 500 , $\pm 1,000$ or $\pm 2,000$ degrees per second (also known as dps). In addition, accelerometers have a measurement range of ± 2 g, ± 4 g, ± 8 g or ± 16 g, where 1 g is the gravity acceleration, which corresponds to approximately 9.81 m/s^2 .

Accelerometers are equipment that measures the acceleration of a body [6]. Positioning systems, slope sensors, as well as vibration sensors are some of their main uses. A practical and largely known use of accelerometers are the mobile phone screens, that rotate its displayed image to adjust itself according to the current angle concerning to gravity acceleration. Built into the circuitry, it is a tiny device, which detects changes in orientation, and sends a signal to the screen to rotate.

An accelerometer will measure the force of 1 g in one of its axis whenever it is in horizontal position, like on a table, for instance. However, when the sensor is inclined, the g force distributes itself for two perpendicular axes. Therefore, it is possible to measure angles using trigonometry.

Fig. 1 shows some possible arrangements for sensor and, Eqs. (1)-(3) calculate the angles. Accelerometers deliver A_x , A_y and A_z values for the X, Y and Z axes, respectively.

The platform uses only the X and Y axes, sometimes called roll (ρ) and pitch (ϕ).

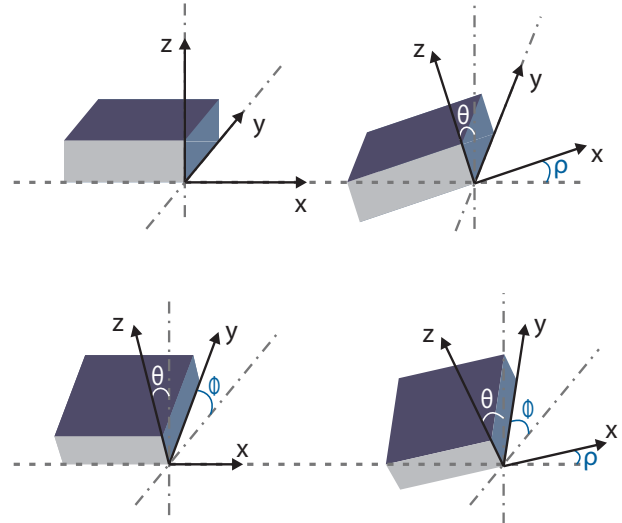


Fig. 1 Angles through accelerometer for three axis.

$$\rho = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \quad (1)$$

$$\phi = \arctan\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right) \quad (2)$$

$$\theta = \arctan\left(\frac{\sqrt{A_x^2 + A_y^2}}{A_z}\right) \quad (3)$$

MPU-6050 gives dimensionless values, called raw values, which suffer conversion to m/s^2 or to multiples of “g” (9.81 m/s^2). The datasheet informs that for a ± 2 g reading measurement range, the system divides the value measured by 16.384. Therefore, if the sensor returns value of 16.384 in X axis, it means that there is 1 g in that axis. The angles measurement can occur before or after this division.

The value obtained is not real for an accelerated movement, horizontally for example, even without modifying the angular orientation, because the movement creates new forces besides the gravitational one on the accelerometers. Gyro sensors assist minimizing this effect.

Gyro sensors maintain or measure orientation. Its

theory is quite simple: the sensor determines how much the orientation over a segment has changed. Once it calculates this parameter, it measures the angular velocity (degrees per second), from which calculates angular shift [7, 8].

To process orientation, we need to initialize the sensor with a known position (using the accelerometer), and then, measure the angular velocity on the X, Y and Z axes using the gyro sensors, within a known time interval. Eq. (4) gives the value of angular shift G_y , measured by the gyro sensor.

$$G_y = \int \omega dt \quad (4)$$

MPU-6050 offers dimensionless values for the gyro sensors. To achieve measures in degrees per second, datasheet informs that we must divide values by 131, considering a measurement range of ± 250 per second.

Gyro sensor undergoes a real value deviation effect, and over time, we made measurements to keep the track of it. When it is on stable positions, its value increases or decreases due to the indefinite integral inherent errors. This deviation rate, known as bias, is relevant for calculations.

Accelerometer has a very precise long-term value, however after sensor stabilization it presents many noises during the instant readings and movements. The gyro sensor on the other hand, has a good short-time value, as well as during orientation changes, but due to final angle measurement integration, it ends up generating an error that tends to increase with time.

The solution to reduce deviations, obtains values that are more precise and reduce calculation of error spreading, is to use an accelerometer and a gyro sensor combined. To achieve that, we decided to use a complementary filter.

2.2 Complementary Filter

The complementary filter is a simple mathematical linear combination, which defines a weight for each input variable, in this case one for the accelerometer and another for the gyro sensor. The sum of these

weights must be equal to 1. Eqs. (5)-(7) express mathematical relations used by this filter.

$$\theta = \alpha G_y + (1 - \alpha)A_{cc} \quad (5)$$

$$\alpha = \frac{\beta}{\beta + dt} \quad (6)$$

$$G_y = \theta + \omega dt \quad (7)$$

where, dt is the sampling time; β the time constant in seconds; ω is the angular velocity measured by the gyro sensor, in degrees per second; and A_{cc} is the angle measured by the accelerometer in degrees.

In order to obtain satisfactory results, the time constant β must be greater than the typical time scale of accelerometer noise. In this work, the sampling time used was $dt = 0.04$ s and the time constant was $\tau = 1$ s, obtaining $\alpha \approx 0.96$. By using this value as α , the system gets a more precise and less noisy measurement.

This filter requires low computational power, therefore it is not mathematically accurate, because of that, small and low-power processing applications have commonly used it [9]. Hence, the complementary filter is more appropriate to this project.

2.3 Mathematical Modelling

Step response analyses generate an empirical model of the system. Eq. (8) describes the system's behavior in case it is of first order [10].

$$T(s) = \frac{1}{\tau s + 1} \quad (8)$$

where, s is the Laplace variable, and τ the time constant.

The time constant τ of a first order system corresponds to 63.2% of the rise time. Nevertheless, if the system behavior is of second order, Eq. (9) describes it [10].

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (9)$$

where: ω_n is the natural frequency, and ζ the damping factor.

To calculate the second order model parameters, it is necessary to determine the settling time T_s , the overshoot O_s , and the peak time T_p , according to Eqs.

(10)-(12), respectively. Eq. (13) calculates the damping factor [10].

$$Ts = \frac{4}{\zeta \omega_n} \quad (10)$$

$$\%Os = 100 e^{\left(\frac{-\zeta \pi}{\sqrt{1-\zeta^2}} \right)} \quad (11)$$

$$Tp = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \quad (12)$$

$$\zeta = \frac{-\ln\left(\frac{\%Os}{100}\right)}{\sqrt{\pi^2 + \ln\left(\frac{\%Os}{100}\right)^2}} \quad (13)$$

3. Materials and Methods

The platform mechanical assembly was possible with low cost materials easily found in the national market, quickly and easily mounted. In order to make it, we have used: one 0.2 m × 0.1 m alveolar carbonate plate; one 0.05 m diameter and 1 m length PVC pipe; one 0.7 m length aluminum angle bracket; one 0.7 m length and 0.01 m width aluminum bar; two servomotors of 0.03 kg·m torque; one MPU-6050 sensor; and one Arduino Nano 3.0.

The MPU-6050 sensor measures the platform angles in real-time. It sends the accelerometers and gyro sensors the values acquired by the Arduino via an I2C communication protocol. Calculating the angle is possible and quite simple after this procedure.

Two servomotors, one for each degree of freedom, control the platform angular position. They are devices capable of moving objects with a determined torque in accordance with an angular velocity previously specified [11]. In addition, as a closed-loop operational system, it received a proportional signal from the controller, leading to a zero error [12]. The Arduino Nano generates PWM signals, which activate the servomotors.

Using the MATLAB software, we mathematically modelled the platform. The Arduino, via USB communication with a microcomputer, made the data

acquisition from the sensors. In order to study and compare the results, we adjusted the models: one of first order (disregarding the overshoot), and another of second order.

For modelling the platform with a step input, it starts from a position, usually zero, and changed its orientation to a known value as fast as possible. Once finished, analyses of the responses on both X and Y axes contributed to evaluate whether the process is of first or second order.

Keeping the platform horizontally stabilized, it rotated 45°, first over the X axis, then the same movement on the Y axis. Finally, Fig. 2 shows the result for a +45° step input over the X axis.

The platform converged for 45° quite fast (about 280 ms) and the overshoot was small. The overshoot occurred, probably, due to the platform inertia and the own servo motor intern control. Now, for a -45° movement on X axis, Fig. 3 shows the result.

On this case, the overshoot was higher, probably due to the bigger platform inertia on this movement direction, but still small and acceptable. The step input

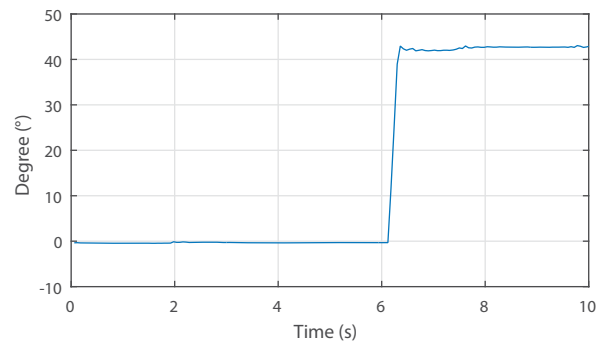


Fig. 2 +45° step input on X axis.

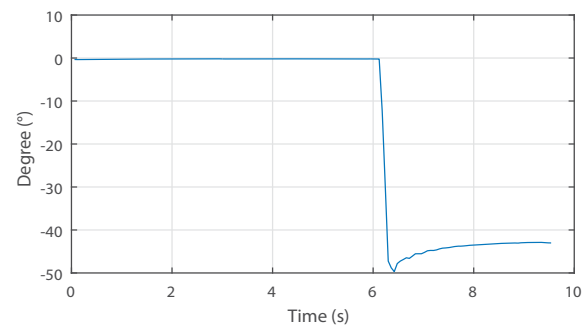


Fig. 3 -45° step input on X axis.

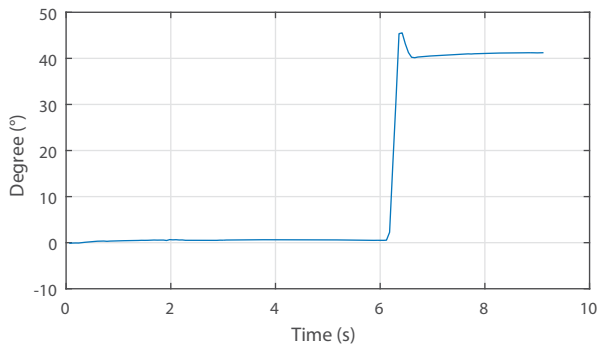


Fig. 4 +45° step input on Y axis.

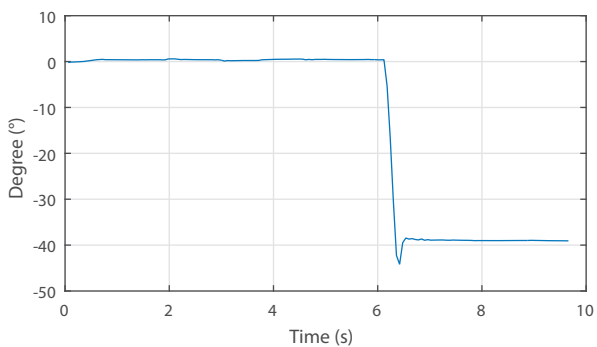


Fig. 5 -45° step input on Y axis.

applied to the Y axis generated the following result, as shown in Fig. 4.

Clearly, the overshoot to this movement was bigger. The fact that the platform has its width bigger than the length, causes a bigger inertia on the Y axis, and then makes this behavior comprehensible. However, the servomotor quickly corrected the effect and stabilized the platform. After the -45° movement, the same result occurred, as shown in Fig. 5.

It is important to inform that the servomotor that controls Y axis has a big clearance among its gears, which are made of plastic and denigrate naturally (approximately 3° of clearance). This is the reason why the overshoot on this axis was bigger. Clearance acts directly on overshooting, which was of 4° (about 8% of 45°) for Y axis.

The platform has a slight different behavior for the X and Y axes, which gets clear after the results. Nevertheless, in order not to build a different controller for each axis, we assumed that they act equally. The step responses on X and Y axes were similar. Therefore, we took only the curve obtained for the Y axis as

reference for this paper.

4. Results and Discussion

Fig. 6 shows the platform after assembled; the sensor is in the bottom, the servomotors and a small counterweight on the left side.

An important information regarding to the platform is that the MPU-6050 sensor finds itself upside down on the bottom of the carbonate plate. The X and Y axes orientation chosen for platform is according to MPU-6050 sensor placement. Looking at the platform from behind, X is positive forwards, and Y is positive to the right side. Therefore, the followings are the movements orientation considered during this work:

- X axis platform rotation: it is the rotation done by the servomotor installed on the right side of the platform, it is positive when moving forwards and negative when moving backwards;
- Y axis platform rotation: For the accelerometer, it is the rotation done by the servomotor installed on the back of the platform, being positive when rotating to the left and negative when rotating to the right. Now, regarding the gyro sensor, the Y gyro sensor has its orientation reversed compared to the accelerometer on Y axis, in other words, positive to the right and negative to the left.

Fig. 7 shows that the complementary filter improved the measured results, since the data from the accelerometer and gyro sensor are noisy and present angle values of the Y axis with large variation.

We have noticed that the complementary filter rejects very well the noises and variations that occurred.



Fig. 6 Platform after assembled.

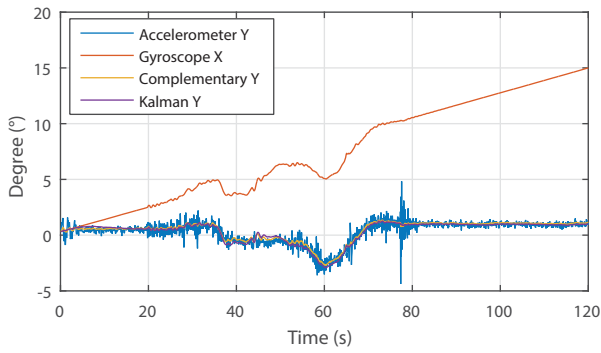


Fig. 7 Complementary filter implementation.

Besides, we also implemented and analyzed the Kalman's filter, which is a mathematical method created by Rudolf Kalman, and appeared on literature on 1960 [13]. It consists in a set of mathematical equations shaped as an interactive computational algorithm, aiming to foresee and estimate model variations for time series, written as state space forms [14]. Kalman's filter also presented itself very efficient. Despite many applications aiming to determine and analyze position and speed coordinates use Kalman's filter [15], we did not use it during the measurements and trials done in this work. Although its behavior has been very close to the complementary one, its implementation is more complex and has increased computational cost.

After analyzing the response of the system seen in Fig. 4, it is possible to observe the platform behavior is similar to a first order model, not considering the overshoot. However, a second order model should better represent the system.

To first order approximation, the results provide the time constant as $\tau = 0.167$ s. Hence, we have gotten the mathematical model for the platform, as shown in Eq. (14).

$$T(s) = \frac{1}{0.167s + 1} \quad (14)$$

Therefore, using MATLAB, we have obtained the model response of Eq. (14) to a step input of $+45^\circ$ and 6.1 s delay. We then compared it to the response obtained on the experiments. It is also important to highlight that the step input used in the experiment had

also a 6.1 s delay. Fig. 8 shows the theoretical results achieved.

By disregarding the overshoot observed in practice, and by analyzing Fig. 5, we noticed that theoretical response came very close to the real one. Once we considered the platform step response as a second order one, we acquired following parameters:

- Damping time $T_s = 1.61$ s;
- Overshoot $O_s = 4.43^\circ$, or 10.7%;
- Peak time $T_p = 0.35$ s.

Then, it is possible to calculate the damping factor ζ and natural frequency ω_n of the system, as follows:

- $\zeta = 0.59$;
- $\omega_n = 4.20$.

Thus, Eq. (15) gives the second order model.

$$T(s) = \frac{17.66}{s^2 + 4.97s + 17.66} \quad (15)$$

Fig. 9 shows the response to a step input of $+45^\circ$, with 6.1 s delay, obtained via MATLAB.

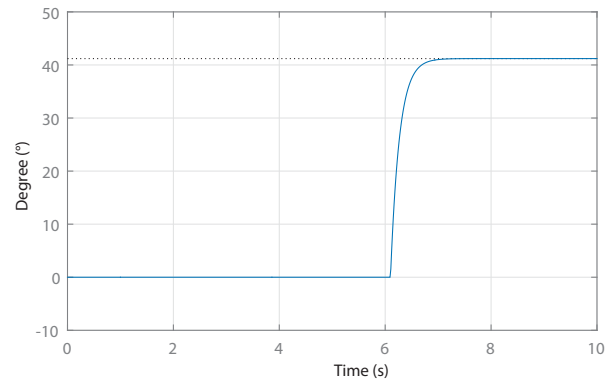


Fig. 8 Response to a step input in a first order model.

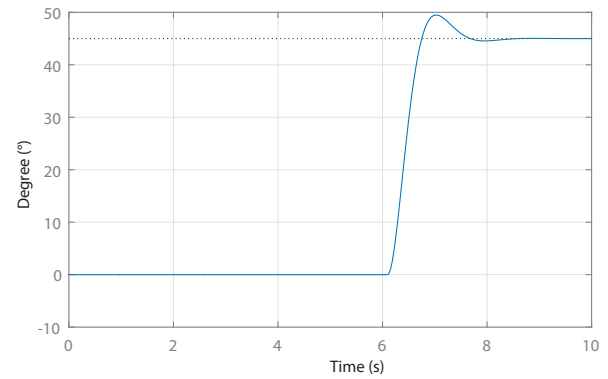


Fig. 9 Response to a step input in a second order model.

The result was tremendously satisfactory, once the behavior of the second order theoretical model got even better when compared to the first order one, as it takes into account the overshoot, verified on the real dynamic of the platform.

5. Conclusions

Mathematical modeling through practical analyses and experiments has proved to be a very effective and applicable method to real processes. We usually calculate mathematical models of dynamic systems from nominal parameters, which are variables often hard to measure accurately. Moreover, it is even more difficult to get those measures when distinct elements and various mechanisms are part of the system.

The mathematical model found in this work has shown to be very close to the real dynamic of system of the platform, even without supplying all the information about the parameters needed to calculate the mathematical model. Once both, first and second order models found in this work achieved approximate results to the real dynamic system, both of them are ideal to describe the behavior of the platform, mostly the second order one.

As a proposal for future works, we suggest switching the servomotors used in this paper for a more accurate one. Servomotors with metal gears would be a good choice in this case. Brushless motors are also good replacements to the servomotors, as they generate faster responses, are more accurate; the only limitation is their lower torque. We also suggest creating a supervisory system that allows changing the parameters of the controller quickly; as well as to generate instant graphics of the data platform, such as current position, control signals, etc. Another idea is to implement another MPU-6050 sensor on the pipe of the platform, in order to identify exactly its position during the movement, improving the control over the system and making a more precise comparison between the

mathematical model and the real process.

References

- [1] Ubertini, F., Venanzi, I., and Comanducci, G. 2015. "Considerations on the Implementation and Modeling of an Active Mass Driver with Electric Torsional Servomotor." *Mech. Syst. Signal Process.* 58: 53-69.
- [2] Miocinovic, S., Shoeb, A., Wang, S., Swann, N., Pathak, A., and Ostrem, J. 2016. "Tremor Severity Estimation Using Liftware Instrumented Eating Utensil." *Neurology* 86 (16).
- [3] Nise, N. S. 2011. *Control Systems Engineering*.
- [4] Ibrahim, D. 2006. *Microcontroller Based Applied Digital Control*. John Wiley and Sons.
- [5] MPU-6000 and MPU-6050 Product Specification Revision 3.4, 2013.
- [6] Accelerometers and How They Work. Texans Instruments. Dec. 2015.
- [7] Tong, K. Y., Mak, A. F. T., and Ip, W. Y. 2003. "Command Control for Functional Electrical Stimulation Hand Grasp Systems Using Miniature Accelerometers and Gyroscopes." *Med. Biol. Eng. Comput.* 41 (6): 710-7.
- [8] Fei, J., Dai, W., Hua, M., and Xue, Y. 2011. "System Dynamics and Adaptive Control of MEMS Gyroscope Sensor." *IFAC Proc. Vol.* 44 (1): 3551-6.
- [9] Coopmans, C., Jensen, A. M., Chen, Y., Jensen, A. M., and Chen, Y. 2014. "Fractional-Order Complementary Filters for Small Unmanned Aerial System Navigation." *J Intell Robot Syst* 73: 429-53.
- [10] Ogata, K. 2010. *Engenharia de Controle Moderno*. Pearson Prentice Hall.
- [11] Venanzi, I., Ierimonti, L., and Ubertini, F. 2016. "Effects of Control-Structure Interaction in Active Mass Driver Systems with Electric Torsional Servomotor for Seismic Applications." *Bull. Earthq. Eng.* 1-15.
- [12] Dorf, R. H. B., and Richard C. 2011. *Modern Control Systems*.
- [13] Grewal, M. S., and Andrews, A. P. 2008. *Kalman Filtering: Theory and Practice Using MATLAB®*: third edition. John Wiley & Sons, Inc..
- [14] Harvey, A. C. 2001. *Forecasting, Structural Time Series Models, and the Kalman Filter*. Cambridge University Press.
- [15] Singhal, T., Harit, A., and Vishwakarma, D. 2012. "Kalman Filter Implementation on an Accelerometer Sensor Data for Three State Estimation of a Dynamic System." *International Journal of Research in Engineering and Technology* 1 (6): 330-4.