COMPARISON OF LINEAR DISPLACEMENT MEASUREMENTS BETWEEN A MEMS ACCELEROMETER AND HC-SR04 LOW-COST ULTRASONIC SENSOR

Azmyin Md. Kamal
Department of Mechanical and Production Engineering, Ahsanullah University of Science and Technology
Dhaka, Bangladesh
azmyin.mpe@aust.edu

Salman Habib Hemel
Department of Mechanical and Production Engineering, Ahsanullah University of Science and Technology
Dhaka, Bangladesh
salmanhabibhemeldrmc@gmail.com

Mohsan Uddin Ahmad
Department of Mechanical and Production Engineering, Ahsanullah University of Science and Technology
Dhaka, Bangladesh
mohsan004buet@gmail.com

Abstract – **In mobile robotics measuring displacement with high level of certainty is very critical to successful localization and mapping. However, sensors such as the depth cameras and LIDARS are often very expensive and difficult to obtain in Bangladesh. To circumvent this problem researchers in Bangladesh uses HC-SR04, a low-cost ultrasonic ranger finder for displacement measurement. MPU6050 a low-cost IMU contains a 3 axis accelerometer which enables it to calculate displacement by double integrating acceleration data. This paper compares the displacement measured using HC-SR04 to the displacement measured by MPU6050 in order to determine whether the displacement calculated from the accelerometer data alone is sufficiently accurate for robot localization. Single step trapezoidal integration method was used and the accelerometer was calibrated using 7000 data sample requiring 1604ms. A convolution type Moving Average Filter took acceleration data 64 times in each time instance for removing noise signals. All data were updated in 15ms intervals. A novel multiplier called Scale Factor (S.F) was introduced to compensate for the resolution issue without amplifying noise signals. Experimental result showed that displacement calculated from accelerometer data can estimate displacement in a mobile robot with reasonable accuracy, provided some assumptions are taken which are not representative of the real world scenarios.**

Keywords — Linear displacement, MPU6050, HC-SR04, accelerometers, moving average, double integration.

I. INTRODUCTION

Localization, the process of determining a robot's position in the world coordinate frame has received much attention in the last decade. A key part of this process is finding the displacement of the robot with high accuracy. This is of paramount importance as literature on the topic indicates that accurate localization leads to accurate environment mapping and an accurate map facilitates in better localization [1 2]. Robots which have been successful in localization (and consequently mapping) are often outfitted with high-performance LIDARs, research grade Inertial Measurement Units (IMU), depth perception cameras and powerful onboard computers. A couple of examples in this regard are the University of Waterloo's 2013 NASA Sample Return Robot [1] and CMU's Atlas with the Multisense SL sensor [2]. However, hardware used in such projects are often very expensive. For example, the IMU used in [1] costs around $1500 while the price of the LIDAR used was around $2000. Funds for such high-cost equipment and technical support from sensor manufacturers are very difficult to obtain by roboticists here in Bangladesh due to various socio-economic reasons.

Thus in order to implement robot localization, engineering students and scientists in Bangladesh often rely on low-cost hobby grade sensors such as the HC-SR04 Ultrasonic ranger for object detection and distance measurements. They also use Invensense MPU6050, a low-cost IMU for measuring motion and orientation in 3D space. Use of HC-SR04 and similar low cost ultrasonic sensors have been studied extensively in works such as object detection, analysis and classification [3, 4, 5], binaural ultrasonic sensor pod design for obstacle avoidance, in a quadruped robot [6], design of one-class classifiers for human detection [7], signal analysis metrics for low cost ultrasonic sensors [8] and using an array of HC-SR04s to demonstrate the usage of Kalman Filter for accurate obstacle detection in an UAV application [26]. From these studies, it can be inferred that HC-SR04 is capable of measuring distances from inanimate objects made of materials such as wood, plastic, brick wall with the high degree of accuracy provided they are placed perpendicularly to its view cone and located at a distance between 20 cm to 150 cm.

Invensense's MPU6050, a well-known commercial IMU is composed of 3 axis gyroscope and 3 axis accelerometer with an auxiliary port for connecting external magnetometer for full 3 axis orientation measurement. Madgwick el at [9] showed the capability of this low-cost IMU for accurately measuring the 3D orientation of an object in comparison with a propriety Kalman filter and ground truth data from an external optical measurement. His work showed that MPU6050 can work very well with an 8bit microcontroller which possesses very limited computing resources for robust orientation measurement. Since then the MPU6050 have seen use in numerous scientific studies such as Fabio Varesano's open source hardware IMU framework [10], smart glove to translate hand gestures into sign language [11], direction agnostic fall monitoring device for old people [12], pose determination for balancing robots [13 14] and so on.

978-1-7281-3445-1/19/$31.00 ©2019 IEEE

A key benefit of having an onboard accelerometer on a mobile robot is that it is possible to determine displacement (or distance) from the accelerometer data directly using double integration. This has been thoroughly examined in [15, 16, 23]. Furthermore, from those studies, it was shown that a system which accelerated and decelerated relatively at a constant rate, its displacement can be estimated using the Trapezoidal numerical integration method whose result remained within an acceptable margin of error. Motivated by these findings, we present our study of comparing the displacement measured by HC-SR04 with the displacement calculated from accelerometer data of MPU6050. Our goal is to answer whether the data from an accelerometer alone is sufficient to determine the displacement of the robot with reasonable accuracy.

For this study we take the following assumptions:

i. The robot moves only in the positive X-axis direction of the IMU.

ii. Taking one axis data of the accelerometer is sufficient to remove the effects of normal forces exerted due to earth's gravity on the accelerometer.

iii. The distance measured by the HC-SR04 ultrasonic sensor is accurate and precise.

iv. The time interval between each reading is identical.

The rest of the paper is arranged as follows: Section 2 describes the experiment procedure, Section 3 describes our findings and Section 4 concludes the paper discussing some of the shortcomings and potential future studies.

II. METHODOLOGY

Fig. 1 shows our test robot for this experiment. It is a food serving robot built to study the interactions between the robot and the students in a dynamic cafeteria environment. At the time of writing this paper, it is a two-wheel drive mobile robot consisting of an Arduino Due embedded development board based on Atmel SAM3X8E, an ARM Cortex-M3 32bit microcontroller running at 86 MHz. It is outfitted with an expansion shield to interface a 16 by 2 Liquid Crystal display (shown in Fig. 2) and a GY-86 10DOF sensor board comprising of InvenSense MPU6050 a 6 axis inertial measurement unit (IMU). The robot is powered by a 3 cell 12.6V 36.6Wh Lipo battery. The Atmel SAM3X8E is programmed using the open source Arduino software framework [21]. The data from MPU6050 is read using I2CDevlib library written by Jeff Rowberg [19] and the ultrasonic sensor data was read using the NewPing library by Tim Eckel [20]. All accelerometer data were captured using CoolTerm serial monitor written by Roger Meier [24].

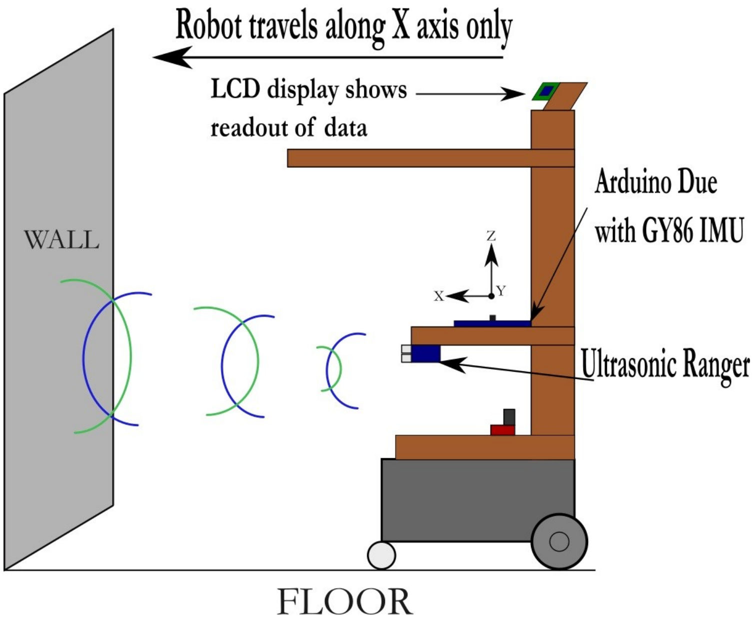

Fig. 1 Schematic diagram of the experiment process

From the datasheet of MPU6050 [22], the full-scale range of the accelerometer was set to ±2g and sensitivity scale factor was set to 16,384. This means that if the output from the accelerometer is 16,384 on a particular axis then the sensor is measuring 1g force on that axis. The accelerometer is then calibrated keeping the robot stationary and the zero reference values for all three axes were determined by averaging the values of 7000 samples. The HC-SR04 ultrasonic sensor was calibrated prior to its installation on the robot. We ascertained that our HC-SR04 unit can measure distances between 20cm to 150cm accurately with 1-1.5% of error when compared to physical measurements using a tape ruler. After calibration, the robot is placed 150mm away from the wall as shown in Fig. 1 for each run. The robot moves forward for 2s in each cycle while recording displacement measured by the ultrasonic sensor and accelerometer simultaneously. Deducting the distance measured by the ultrasonic sensor from the initial and final positions gives us the ground truth data for comparing the displacement calculated using the data obtained from the accelerometer.

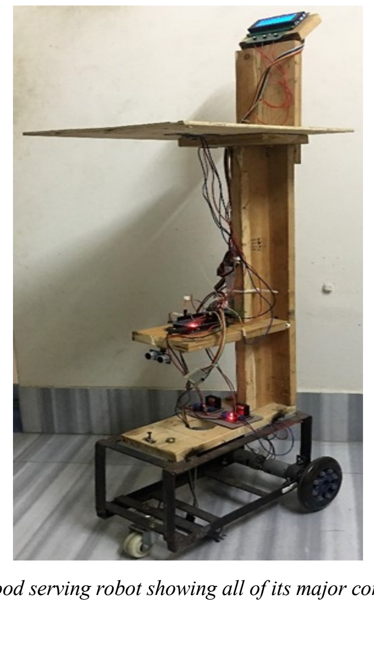

Fig. 2 Food serving robot showing all of its major components

Fig. 3 LCD showing calibration data

In each time increment, we log 64 readings from the MPU6050 and then fed the data through a Moving Average Filter, a Low Pass FIR filter programmed on basis of [17] and [18]. Our DSP filter is of convolution type, meaning each output point is produced by sampling a set of input points and then averaging the sum of those values. Mathematically this can be written as

$$y[i]=\frac{1}{M}\sum_{j=0}^{M-1}x[i+j] \tag{1}$$

The filtered value is then subtracted from the zero reference values for obtaining the body acceleration value at that instant. Even with two low pass filters implemented in software and hardware end (MPU6050 has a 250Hz low pass filter activated by default) respectively, noise signals were still present which registered erroneous displacements. To rectify this, we employed another software filter named Mechanical Noise Window (MNS) as discussed in [17]. Its low and high threshold values were determined experimentally. The acceleration values were then converted to cm/s^2 using the following equation

$$a_g=\frac{a_{LSB}*9.81*100}{16834}cm/s^2 \tag{2}$$

This data is now suitable for double integration to determine the displacement of the robot in cm. We implemented the single step Trapezoidal rule for double integrating the accelerometer data using the following formulas adapted from [23].

$$\int_{t(0)}^{t(n)}a(t)dt\cong\sum_{i=1}^{n}\left(\frac{a(i-1)+a(i)}{2}\right)\Delta t \tag{3}$$

$$v_c(i)=S.F*v_c(i-1)+\left(\frac{a(i-1)+a(i)}{2}\right)\Delta t \tag{4}$$

$$d_c(i)=d_c(i-1)+\left(\frac{v(i-1)+v(i)}{2}\right)\Delta t \tag{5}$$

Here *dc (i)* is the instantaneous displacement and S.F is a “scale factor” which will be discussed in more details in Section III.

III. RESULTS AND DISCUSSION

To commence our study we required a suitable sample size for running the calibration routine. Table 1. shows the time required in milliseconds by the Atmel SAM3X8E for executing the calibration method for different sample sizes. It

TABLE I. Effect Of Number Of Sample Points On Calibration Time

Number of samples	*Ax value (bits)*	*Time (ms)*
10	-124	2
50	-127	11
100	-99	22
150	-106	34
300	-120	68
500	-120	114
750	-114	172
1000	-117	229
1000	-119	231
2000	-123	458
3500	-123	803
5000	-122	1148
7000	-123	1604
7500	-122	1722
10,000	-122	2296

is clear from the data that once we move above 2000 sample size, the zero reference value for X-axis becomes stable. Since our experiment is designed to be implemented in short bursts, we opted to go for the 7000 sample size which requires 1604 ms to be executed.

Table 2. shows the time required in milliseconds by the microcontroller to execute the moving average method in different configurations for removing noise signals from the accelerometer data. Iteration number here refers to the number of calibrated data the MCU will poll from the MPU6050 within the 15ms window to calculate the body acceleration value in LSB/g format [22]. The data showed that doubling iteration number doubles the time required to execute the Low Pass filter. However, for 256 and 512 iterations, the time required was exactly the same and this oddity held true after multiple calibration attempts.

TABLE II. Effect of the number of iterations on the loop time for executing moving average method

Iteration number	*Loop time (ms)*
8	2
16	4
32	8
64	15
128	30
256	59
512	59
1024	118

For our study, we chose 64 iteration number which is the sample size used in Freescale application note [17]. However, even with the Moving Average filter, there existed errors in accelerometer data caused by sensor noise. These errors were interpreted as a constant velocity by the microcontroller and consequently summed as a steady movement even though the robot was stationary. A second software filter dubbed the Mechanical Noise Window (MNS) first introduced in [17] was used to cut off values within a certain band to eliminate this problem. Fig. 2 shows the raw and calibrated data from the accelerometer and Fig. 3 demonstrates the use of MNS to eliminate the erroneous displacement caused by the sensor noise shown in Fig 2. The error is linear in nature which was expected as in a standstill position, no other force was acting along the X-axis direction and on the basis of assumption 2, gravity vector which is assumed to be registered only along the Z-axis by the IMU in a stationary position was omitted from the start. On the basis of assumption 1, we only considered the absolute value of the calibrated data. As mentioned previously the motors are allowed to spin forward for only 2s. When the power is cut off the robot will decelerate but still move forward by some amount. Thus we are required to take the calibrated value because when decelerating, a negative displacement is computed which reduces the overall calculated displacement thereby giving false readings.

Table 3. shows the calculated displacement from the accelerometer data using (3 – 5). As shown in Fig. 1 the sonar, measures the robot's displacement from the wall by deducting initial and final position readings. The initial position for all runs is 150 cm. However, we noticed that the IMU data lagged behind irrespective of the number of iteration set for the moving average method. Using the CoolTerm serial monitor, we monitored the internal variables and deduced that with the default accelerometer sensitivity the IMU was not updating fast enough as its I2C bus is capped at 400Khz even though our microcontroller is capable of much faster data transfer.

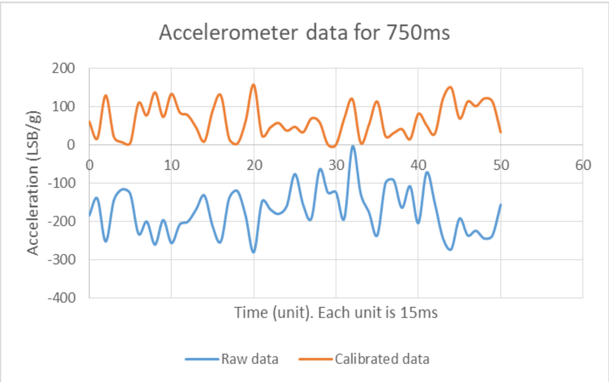

Fig. 4 Raw and Calibrated data

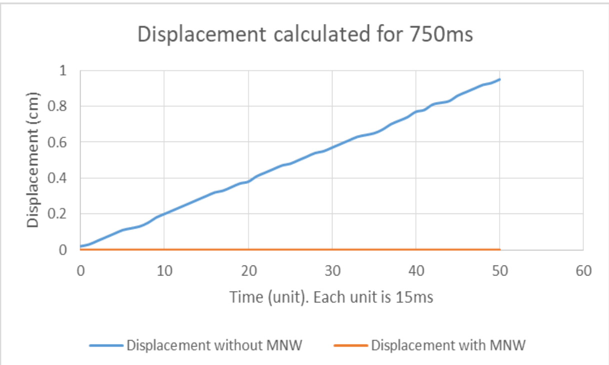

Fig. 5. Use of Mechanical Noise Window (MNW)

TABLE III. Calculated displacements with percent of error

No.	***Sonar final (cm)***	***Sonar displacement (cm)***	***Accelerometer displacement (cm)***	***Scale Factor (S.F)***	***Error (%)***
1	53	97	15	10	85
2	51	99	15	10	85
3	48	102	15	10	85
4	48	102	19	15	81
5	44	106	25	15	76
6	47	103	22	15	79
7	47	103	32	20	69
8	42	108	34	20	69
9	42	108	29	20	73
10	40	110	38	25	65
11	41	109	39	25	64
12	43	107	35	25	67
13	40	110	47	30	57
14	41	109	42	30	61
15	40	110	40	30	64
16	38	112	52	35	54
17	39	111	58	35	48
18	39	111	52	35	53
19	43	107	57	40	47
20	44	106	55	40	48
21	44	106	55	40	48
22	41	109	61	45	44
23	41	109	61	45	44
24	44	106	59	45	44
25	41	109	64	50	41
26	40	110	69	50	37
27	40	110	67	50	39
28	49	101	93	60	8
29	38	112	94	60	16
30	40	110	96	60	13
31	38	112	98	65	13
32	39	111	90	65	19
33	37	113	97	65	14
34	35	115	93	70	19
35	38	112	105	70	6
36	37	113	107	70	5
37	38	112	113	75	1
38	38	112	114	75	2
39	39	111	113	75	2
40	37	113	110	75	3

We decided not to change the sensitivity factor as most of the literature on MPU6050 used the default sensitivity scale (+/-2g). This is due to the fact that when the sensitivity of the accelerometer is increased, it proportionally increases the effect of the noise source in the measured signal. Sources of noise in MEMS accelerometer is explained in depth in [25].

For the reasons above, we introduced a new multiplication factor named Scale Factor (S.F). S.F is an integer used in (4) to increase the value of velocity to compensate for the sensitivity problem without amplifying signal noise.

From Table 3, we observed that for a S.F value of 70, the data between the Ultrasonic sensor and accelerometer starts to converge well. Table 4. shows the data for determining a S.F value for which the error will be minimum.

TABLE IV. Determination of S.F value

No.	*Sonar final (cm)*	*Sonar displacement (cm)*	*Displacement IMU (cm)*	*Scale Factor (S.F)*	*Error (%)*
1	30	120	128	70	7
2	25	125	104	70	17
3	24	126	113	70	10
4	22	128	117	71.5	9
5	22	128	115	71.5	10
6	21	129	112	71.5	13
7	23	127	125	72.5	2
8	24	126	131	72.5	4
9	25	125	127	75	2
10	**26**	**124**	**124**	**75**	**0**
11	**25**	**125**	**125**	**75**	**0**
12	**23**	**127**	**131**	**75**	**3**

From Table 4 it is evident that the S.F value for our experimental robot is 75, a value for which the error is consistently within 5% of error margin. Our experiment indicates that low-cost IMU can be used for measuring linear displacement with reasonable accuracy but requires a controlled environment. Thus we conclude that though IMUs are capable of displacement measurements, they should not be used as the only sensing device for this purpose and should be coupled with other sensors such as HCSR04, LIDARs and so on. We claim this notion from the fact that at the onset of this study we had taken four assumptions. These assumptions are not representative of most real-world applications. A byproduct of our study is that both HC-SR04 and MPU6050 have shown remarkable performance with the Arduino platform which indicates their potential for use in complex projects by roboticists here in Bangladesh.

IV. CONCLUSION

This paper presents a comparative study of the performance and reliability of using low-cost commercial MEMS accelerometer in measuring linear displacement. It was compared to the displacement measured by HC-SR04, a low-cost hobby grade ultrasonic ranger widely available in Bangladesh. Even though a good agreement between the measured values were seen, assumptions were imposed which are not representative of real-world scenarios. Most notability assumption about the accuracy of HC-SR04 is not always fixed due to the fact that HC-SR04's view-cone changes with respect to the sensor's distance from the object. Furthermore, the assumption about each loop running at exactly 15ms is not true. In our initial trials, we had observed that as the ultrasonic ranger came closer to the wall, it changed the overall loop time. Tim Eckel's library resolved this issue by utilizing an independent *timeout* argument which allowed receiving data for a fixed period of microseconds before returning 0. We suggest that linear displacement computed from accelerometer data should be used in conjunction with other sensors. In this regard, the use of the weighted average method is recommended for maximum accuracy as in perspective of a microcontroller application, this method is easy to code and requires low computing resources. Our future work will include the use of all three axes data, advanced numerical methods and will use the timer modules to independently count time increments rather than depending on Arduino's delay method which drifts with time.

V. ACKNOWLEDGMENT

The authors acknowledge the support of the Department of Mechanical and Production Engineering of Ahsanullah University of Science and Technology for providing workspace in their Simulation Lab. We are also thankful to the open source Arduino community for providing the necessary codes and helpful documentation on interfacing HC-SR04 and MPU6050 with Arduino Due.

VI. REFERENCES

[1] A. Das, M. Diu, N. Mathew, C. Scharfenberger, J. Servos, A. Wong, *et al.*, "Mapping, Planning, and Sample Detection Strategies for Autonomous Exploration," *Journal of Field Robotics,* vol. 31, pp. 75-106, 2014.

[2] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, et al., "Optimization-Based Locomotion Planning, Estimation, and Control Design for The Atlas Humanoid Robot," *Autonomous Robots*, vol. 40, pp. 429-455, 2015.

[3] C. N. Yalung and C. S. Mathew Adolfo, "Analysis of Obstacle Detection Using Ultrasonic Sensor," *International Research Journal and Engineering Technology*, pp. 1015–1019, 2017.

[4] R. A. Dimitrov and D. Minchev, "Ultrasonic Sensor Explorer," in *19th Int. Symp. on Electrical Apparatus and Tech., SIELA 2016*, Borgus, Bulgaria, Aug. 2016, pp. 1-5.

[5] D. Vakula and Y. K. Kolli, "Low Cost Smart Parking System for Smart Cities," in *Proc. Int. Conf. Intell. Sustain. Syst.,* Dec. 2017, pp. 280–284.

[6] X. T. Le, S. B. Oh, W. S. Lee, C. S. No, and S. H. Han, "Design of Intelligent Mobile Robot System Based on Ultrasonic Sensors," in *Int. Conf. Control. Autom. Syst.,* Dec. 2007, pp. 72–76.

[7] Sonia, A. M. Tripathi, R. D. Baruah and S. B. Nair, "Ultrasonic Sensor-Based Human Detector Using One-Class Classifiers," in *IEEE Int. Conf. on Evolving and Adaptive Intell. Syst.,* Dec. 2015, pp. 1-6.

[8] G. Gibbs, H. Jia, and I. Madani, "Obstacle Detection with Ultrasonic Sensors and Signal Analysis Metrics," in *Transp. Res. Procedia*, vol. 28, pp. 173–182, 2017.

[9] Madgwick, S.O.H.; Harrison, A.J.L.; Vaidyanathan, A. "Estimation of IMU And MARG Orientation Using A Gradient Descent Algorithm," in *Proc. of IEEE Int. Conf. of Rehabilitation Robotics*, Zurich, Switzerland, 29 Jun.–1 Jul., 2011, pp. 1–7.

[10] F. Varesano, "FreeIMU: An Open Hardware Framework for Orientation and Motion Sensing," 2013, Accessed on: Dec. 2018 [Online] Available: arXiv:1303.4949v1 [cs.ET].

[11] A. Yudhana, J. Rahmawan, and C. U. P. Negara, "Flex Sensors and MPU6050 Sensors Responses on Smart Glove for Sign Language Translation," in *IOP Conf. Ser. Mater. Sci. Eng.* 2018, vol. 403, no. 1, doi:10.1088/1757-899X/403/1/012032.

[12] J. Luo, B. Zhong, and D. Lv, “Fall Monitoring Device for Old People Based on Tri- Axial Accelerometer,” Int. J. Adv. Comput. Sci. Appl., vol. 6, no. 5, pp. 34–39, 2015.

[13] M V Momot, E V Politsinskaia, A V Sushko and A Semerenko, “Justification of Filter Selection for Robot Balancing in Conditions of Limited Computational Resources,” *IOP Conf. Ser. Mater. Sci. Eng,,* 2016, Accessed on: Dec. 2018 [Online] Available: https://doi.org/10.1088/1757-899X/142/1/012108

[14] A.K.M Asfaque Shakil Shaon, S. Bhowmik, B. K. Bhawmick, P. Das and N.K. Das “Design And Implementation Of A Self-Balancing Robot” *in Proc. Int. Conf. on Mechanical Eng. And Renewable Energy*, Dec. 2017, Accessed on: Dec. 2018 [Online]

[15] T.J. Cheng et al., “Characterisation of Rollator Use Using Inertial Sensors,” *Healthc. Technol. Lett.*, vol. 3, no. 4, pp. 303–309, 2016.

[16] J. Hrisko, *Introduction To Capacitive MEMS Accelerometers and A Case Study On An Elevator,* Oct. 2017, Accessed on: Dec. 2018 [Online] Available: https://engineersportal.com/blog/2017/9/25/accelerometer-on-an-elevator

[17] K. Seifert and O. Camacho, *Implementing Positioning Algorithms Using Accelerometers,* 2007, Freescale Semiconductor, Accessed on: Dec. 2018 [Online] Available:https://www.nxp.com/docs/en/application-note/AN3397.pdf [Accessed: December 2018]

[18] S.W.Smith, “Moving Average Filters,” in *The Scientist and Engineer's Guide to Digital Signal Processing*, San Diego, CA, California Technical Publishing, 1999. Accessed on: Dec. 2018 [Online] Available: http://www.dspguide.com/pdfbook.htm

[19] Jeff Rowberg, *I2CDevlib,* Sept. 2018, [Online] Available:https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050

[20] Tim Eckel, *Arduino New Ping,* Nov. 2018, [Online] Available:https://bitbucket.org/teckel12/arduino-new ping/wiki/Home#!history

[21] Arduino, *Arduino IDE 1.8.4,* Dec. 2018, Accessed on: Dec. 2018 [Online] Available: https://github.com/arduino/Arduino/

[22] InvenSense Inc. Technical Staff, “MPU-6000 and MPU-6050 Product Specification Revision 3.4”, p13, InvenSense Inc., Aug. 2013

[23] M. Arraigada and M. Partl, “Calculation of displacements of measured accelerations, analysis of two accelerometers and application in road engineering” in *Proc. 6th Swiss Transport Research Conf.,* Ascona, Switzerland, Mar. 2006*,* pp. 30-60

[24] Roger Meier, *CoolTerm*, Dec. 2018 [Online] Available: http://freeware.the-meiers.org/

[25] L. B. Pupo, “Characterization of errors and noises in MEMS inertial sensors using Allan variance method,” M.S thesis, UPC, Barcelona, Spain, 2016. [Online] Available: http://hdl.handle.net/2117/103849

[26] J.S.G.Guerrero, A.F.C.Gonzalez, J.I.H.Vega and L.A.N.Tovar, “Instrumentation of an Array of Ultrasonic Sensors and Data Processing for Unmanned Aerial Vehicle (UAV) for Teaching the Application of the Kalman Filter,” *Procedia Computer Science*, vol. 75, pp. 375-380, 2015.