Position Estimation using Inertial Measurement Unit (IMU) on a Quadcopter in an Enclosed Environment

Christian B. Abellanosa, Ruth Pearl J. Lugpatan, and Diogenes Armando D. Pascua

***Abstract*— Inertial sensors, like accelerometers and gyroscopes, are rarely used by themselves to measure displacement. Currently, the use of low cost IMUs for this particular application is very impractical due to the accumulation of errors from various sources. In this study, a position estimation using Inertial Measurement Unit (IMU) in a quadcopter is developed. The objective of the researchers is to develop an algorithm that will make the quadcopter travel a certain displacement or position. The algorithm uses mainly the information from the IMU and minimizes the error between the desired displacement and the actual displacement travelled by the quadcopter. The implemented approach is to use an error factor. The error factor is a value that is inversely proportional to the desired displacement. Experiments indicate that the system provides accurate pose estimates and are robust.**

***Keywords*-Accelerometer, Gyroscopes, Inertial Measurement Unit , Quadcopter**

I. Introduction

As Inertial Measurement Unit(IMU) sensors are unable to measure position directly (only angular velocity and acceleration), precisely determining the position of quadcopters using IMU has been proven to be a great challenge, especially in applications where the displacement is to be estimated over a long period of time due to error accumulation in the acceleration measurement. This study describes a method to implement a double integration of the signal obtained from the sensor using a microcontroller to obtain position.

The objective of the study is to develop an algorithm that will make the quadcopter travel a certain displacement or position. This algorithm uses mainly the information from the IMU and minimizes the error between the desired displacement and the actual displacement travelled by the quadcopter.

II. Methodology

A. Hardware Design of the Quadcopter

The hardware design ,as shown in Figure 1, have a major role in the algorithm for flight of the quadcopter. The quadcopter developed by the researchers is a Do-It-Yourself (DIY) quadcopter. The parts included to make this quadcopter are four motors that were connected to an Electronic Speed Controller (ESC), a li-po battery, a microcontroller or a microprocessor, and various sensors. The flight controller chosen by the researchers is an ATMega328 microcontroller with an MPU6050 sensor as shown in Figure 2. This microcontroller is where the code was uploaded to do all the processes for maneuvering the quadcopter and the position estimation

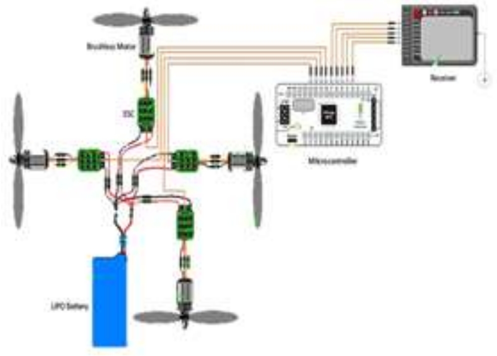


Fig. 1. The Quadcopter Hardware Components.

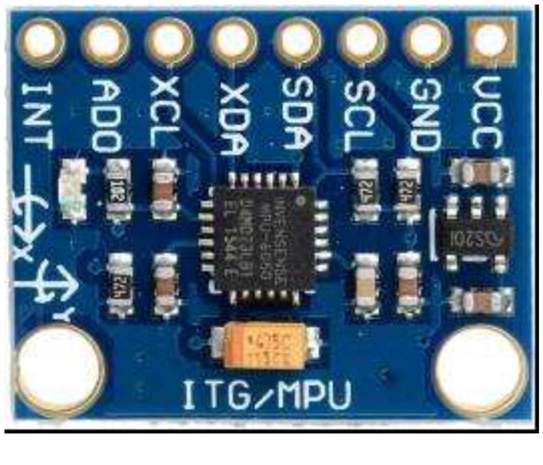


Fig 2. MPU6050 Inertial Measurement Unit Sensor

B. Software Design of the Quadcopter

The general algorithm for the development of the software to be uploaded to the to flight controller of the system is outlined in Figure 3.The use of Proportional Integral Derivative (PID) control as well as Kalman filtering are essential components of the system software.

Christian B. Abellanosa, and Ruth Pearl J. Lugpatan are graduates of BS Electronics Engineering of Mindanao University of Science And Technology.
Diogenes Armando D. Pascua is now with the Department of Electronics Engineering of Mindnanao University of Science And Technology, CM Recto Avenue, Lapasan, Cagayan de Oro, 9000 Philippines.

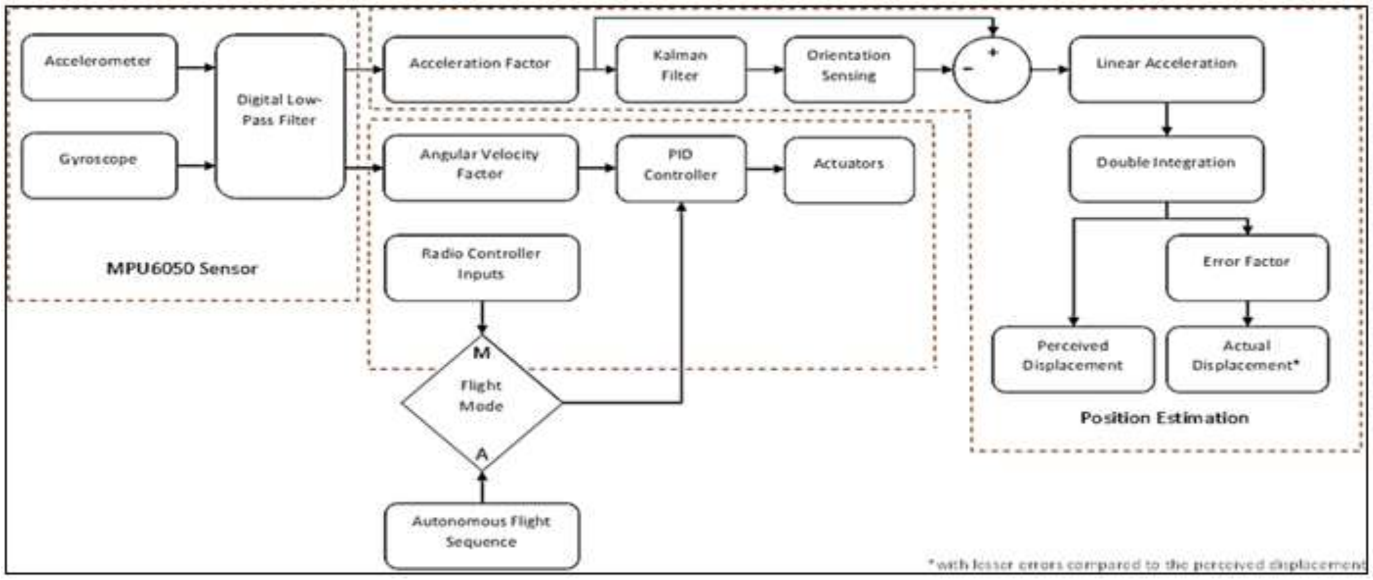


Fig. 3 General Flow Chart of the Software

Noise and drift in sensors' data are not good values to work with. The researchers used the configurable Digital Low Pass Filter of MPU6050 on board for both gyroscope and accelerometer. Filtered accelerometer and gyroscope raw readings which are in binary units were converted to its appropriate units for acceleration and angular velocity. The acceleration should be in units of m/s2 and the angular velocity should be in degrees/s. The researchers will use 1670.133/(m⁄s^2) accelerometer factor and 131.072/(°⁄s) angular velocity factor.

A. *Algorithm for Flight*

Quadcopters are aerodynamically unstable, which is why they can't be controlled just by changing the speed of the motors linearly with respect to the radio controller inputs (Aileron, Elevator, Rudder and Throttle). This is why an IMU has to be used to remove the error between quadcopter's real orientation and reference orientation from the radio controller. This is the update process of the PID controller. The controller outputs a control signal used to control the speed of the actuators. On this section, the PID output is calculated and will affect the speed of the motors. PID controller has three gains, an input, a setpoint and bounds. These three gains affect how aggressive the PID controller operates. PID input is the value coming from the sensors. PID setpoint is the value calculated from the receiver data. PID bounds is the limit of the PID output. The relationship between the three-axis sensor and throttle inputs with four outputs in the motor control system as a whole was shown in figure 4. Figure 5 shows the corresponding codes in the Arduino environment of the PID Controller Block Diagram.

B. *Position Estimation Method*

As illustrated on Figure 3, the orientation and linear acceleration of the quadcopter must be known before double integration takes place. In order to determine the orientation of the quadcopter, acceleration values are needed.

The acceleration data are noisy, resulting to a noisy orientation reading. Therefore, the researchers used a Kalman Filter to further filter out acceleration noise. After filtering the acceleration values, the researchers then calculates the orientation of the quadcopter for roll angle (ϕ) and pitch angle (θ).

For the quadcopter to travel a distance, it needs to tilt its body with respect to its roll and pitch angle. As it tilts an angle, the accelerometer is getting acceleration values with gravity. Before the researchers can calculate the displacement obtained by the quadcopter, linear acceleration must be known first, thus it is essential to remove gravity in acceleration of x and y axes. In calculating the linear acceleration, equations 1 and 2 are used by the researchers.

$$A_{x\text{-}linear} = (A_x - 9.81\sin\phi)/\cos\phi \qquad (1)$$

$$A_{y\text{-}linear} = (A_y - 9.81\sin\theta)/\cos\theta \qquad (2)$$

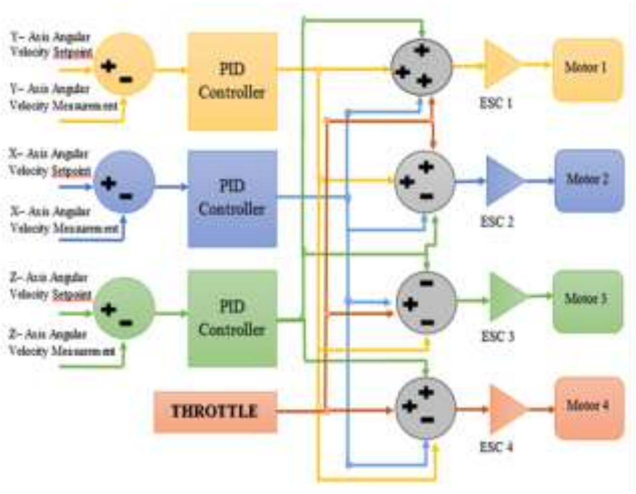
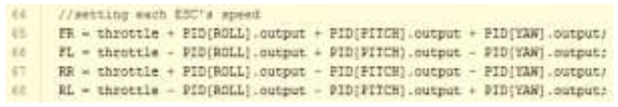

Y- Axis Angular Velocity Setpoint
Y- Axis Angular Velocity Measurement
X- Axis Angular Velocity Setpoint
X- Axis Angular Velocity Measurement
Z- Axis Angular Velocity Setpoint
Z- Axis Angular Velocity Measurement
PID Controller
PID Controller
PID Controller
THROTTLE
ESC 1
ESC 2
ESC 3
ESC 4
Motor 1
Motor 2
Motor 3
Motor 4

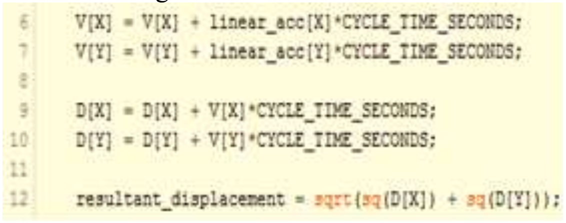
Fig. 4. PID Controller Block Diagram



```
//setting each ESC's speed
FR = throttle + PID[ROLL].output + PID[PITCH].output + PID[YAW].output;
FL = throttle - PID[ROLL].output + PID[PITCH].output - PID[YAW].output;
RR = throttle + PID[ROLL].output - PID[PITCH].output - PID[YAW].output;
RL = throttle - PID[ROLL].output - PID[PITCH].output + PID[YAW].output;
```

Fig 5. Code snippet for determining which motor is affected by a particular PID output

Figure 7 shows the position estimation method implemented by the researchers. Double integrating the linear acceleration will result to displacement in the x and y axes. The corresponding codes for double integration and pythagorean theorem for calculating the resultant displacement can be observed in Figure 6.



```
V[X] = V[X] + linear_acc[X]*CYCLE_TIME_SECONDS;
V[Y] = V[Y] + linear_acc[Y]*CYCLE_TIME_SECONDS;

D[X] = D[X] + V[X]*CYCLE_TIME_SECONDS;
D[Y] = D[Y] + V[Y]*CYCLE_TIME_SECONDS;

resultant_displacement = sqrt(sq(D[X]) + sq(D[Y]));
```

Fig. 6. Code snippet for double integration and calculating the resultant displacement

In order to minimize the error between the desired displacement and the actual displacement made by the quadcopter, the researchers obtained an error factor. The quadcopter will stop travelling when its resultant displacement reaches a value equivalent to 1/error factor. The process of how the researchers obtained an error factor equation can be observed in Figure 8. The error factor which is a function of desired displacement is not linear, thus, the researchers must conduct a test to achieve the error factor curve with a unit measure of 1 meter. The researchers conducted a test where error factors are pre-set starting from 6 to 192 to determine what percentage errors these error factors yield. Using these percentage errors, the researchers are then able to derive it to a desired displacement.

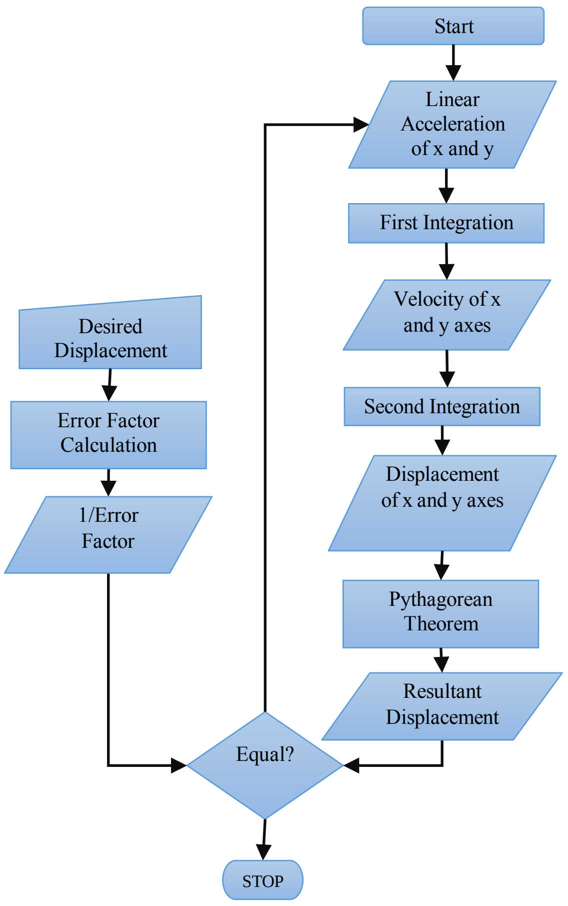


Fig. 7. Position Estimation Method

After the researchers had plotted the error factor curve, an equation of the curve was generated with respect to the percentage error. This equation was used to correct the desired displacement of the quadcopter and the actual displacement.

Error factor is the value calculated using a desired displacement to achieve a particular percentage error. This value determines when the quadcopter will stop moving. Error factor is different for different desired displacements. The quadcopter will stop moving whenever the calculated resultant displacement travelled by the quadcopter is equal or greater than 1 divided by the error factor. The calculated resultant displacement is incomparable to the actual displacement travelled by the quadcopter because of the errors present from the sensor itself to the noise effects. For instance, the desired displacement is 5 meters, the calculated resultant displacement may not be equal to 5 meters and it may be more or less. The error factor determines what value of resultant displacement is calculated before the quadcopter stops and will yield approximately 5 meters away from the initial point.

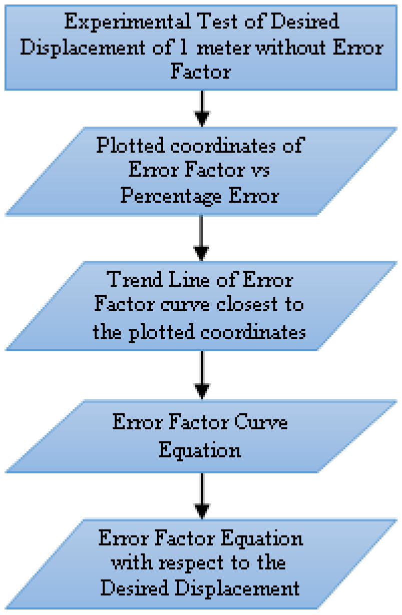

Experimental Test of Desired Displacement of 1 meter without Error Factor
Plotted coordinates of Error Factor vs Percentage Error
Trend Line of Error Factor curve closest to the plotted coordinates
Error Factor Curve Equation
Error Factor Equation with respect to the Desired Displacement

Fig. 8. How Error Factor was obtained

III. Experimental Results

For the objective of this study, the researchers tested a 1 meter desired displacement without an error factor to show how the accumulation of errors in the double integration of acceleration can greatly affect the position estimation of the quadcopter. Shown in Table 1 is the data gathered in this set-up.

Table 1. Test Results with the Desired Displacement of 1 meter without an Error Factor

Trial	Actual Displacement
1	8.14
2	14.5
3	13.46
Average	12.03

To minimize the errors, the researchers obtained an equation for the error factor with respect to the percentage error relative to a perceived displacement of 1 meter. Data in Table 2 was gathered through the series of trials made by the researchers to get a sampled data of error factor and its corresponding percentage error. Table 2 shows the distance travelled using pre-determined error factors.

Table 2. Test Results with the Desired Displacement of 1 meter without an Error Factor

Error Factor	Distance Travelled (m)	Percentage Error (%)
6	3.18	218
12	2.47	147
24	1.52	52
48	1.18	18
96	1.12	12
192	1.08	8

The data on table 4.2 was plotted using Microsoft PowerPoint 2013 on an X Y (Scatter) chart and generated an equation based on the trend line provided by the plots. The generated equation is shown in Equation 3 where *y* is the percentage error and *x* is the error factor.

$$y = 1553.7x^{(-1.064)} \quad (3)$$

Figure 9. shows the error factor curve made by the quadcopter. A trend line was generated along with the trend line equation. This equation will then be derived to an equation which is a function of desired displacement

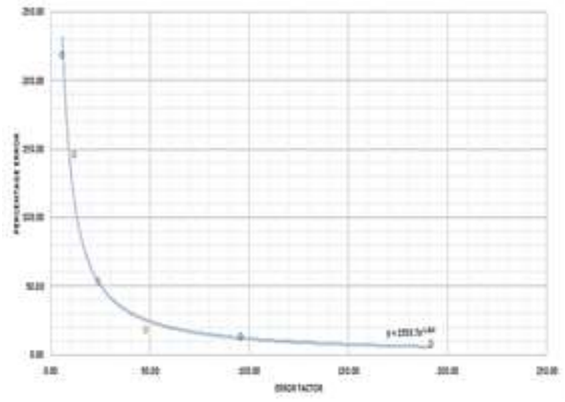
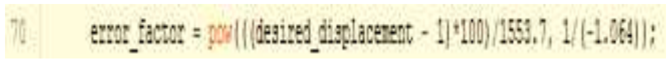


Fig. 9. Graphical Representation of Percentage Error with respect to Error Factor

The percentage error can be derived into an error factor and is a function of desired displacement from Equation 3. Figure 10 is the error factor equation in Arduino code of Equation 4.

$$e_{f\text{-}desired} = \text{-}1.064th\ root\ of\ \{[(d_{desired} - 1)(100)]/1553.7\} \quad (4)$$



```
error_factor = pow(((desired_displacement - 1)*100)/1553.7, 1/(-1.064));
```

Fig. 10. Error factor equation in Arduino Code

To test the performance of the implemented algorithm shown in Figure 7, three trials were performed in every desired displacement. In the implemented algorithm, the value of 1/error factor is compared to the displacement obtained by the quadcopter through double integration of its linear acceleration, thus making the quadcopter to stop its travel after the condition was satisfied. Table 3 – Table 8 present the results of the trials under each test condition. Using Equation 4, the researchers obtained these data with different desired displacement for three trials each displacement.

Table 3. Test Results of Experiments with the Desired Displacement of 1 meter

Trial	Actual Displacement	Percentage Error
1	1.13	13%
2	0.94	-6%
3	1.14	14%
Average	1.07	7%

Table 4. Test Results of Experiments with the Desired Displacement of 2 meters

Trial	Actual Displacement	Percentage Error
1	1.56	-22%
2	2.22	11%
3	1.87	-6.5%
Average	1.88	-6%

TABLE 5. TEST RESULTS OF EXPERIMENTS WITH THE DESIRED DISPLACEMENT OF 3 METERS

Trial	Actual Displacement	Percentage Error
1	2.43	-19%
2	2.49	-17%
3	2.84	-5.33%
Average	2.59	-13.78%

TABLE 6. TEST RESULTS OF EXPERIMENTS WITH THE DESIRED DISPLACEMENT OF 4 METERS

Trial	Actual Displacement	Percentage Error
1	3.4	-15%
2	3.32	-17%
3	3.19	-20.25%
Average	3.3	-17.5%

TABLE 7. TEST RESULTS OF EXPERIMENTS WITH THE DESIRED DISPLACEMENT OF 5 METERS

Trial	Actual Displacement	Percentage Error
1	4.68	-6.4%
2	4.17	-16.6%
3	4.41	-11.8%
Average	4.42	-11.6%

TABLE 8. TEST RESULTS OF EXPERIMENTS WITH THE DESIRED DISPLACEMENT OF 6 METERS

Trial	Actual Displacement	Percentage Error
1	4.64	-22.67%
2	7.14	23.5%
3	7.25	20.83%
Average	6.43	7.17%

It is observed that the percentage errors are increasing as the desired displacement was increased. This is because of the double integration of the acceleration data given by the sensor. Double integration is dependent to time which means that the result of the integration is a function of time. The time the quadcopter is travelling is increased when the desired displacement is increased since the quadcopter is travelling at a constant set of velocities.

IV. CONCLUSION

Inertial sensors are usually not used to obtain displacement because of the drift due to numerical integration, making these sensors most suitable for orientation sensing only. However,this study has successfully implemented a method to estimate the displacement from inertial sensors' readings with lesser errors. Experiments are also conducted to demonstrate the effectiveness of the method. This system is useful in situations where displacement precision is not extremely critical and where no external reference measurement is provided.

REFERENCES

[1] D. Vincent. (February 2013). Accurate Position Tracking Using Inertial Measurement Units [[online] Available: http://www.pnicorp.com/wp-content/uploads/Accurate-PositionTracking-Using-IMUs.pdf

[2] Novatel. (February 2014). IMU Errors and Their Effects. (Aplication Notes) [online] Available: http://www.novatel.com/assets/Documents/Bulletins/APN064.pdf

[3] H. Fourati,. N. Manamanni , L. Afilal and Y. Handrich. "Position Estimation Approach by Complementary Filter-aided IMU for Indoor Environment." in Proc.2013 European Control Conferenec.,2013, pp. 4208 – 4213.

[4] K. Gustavsson," UAV Pose Estimation using Sensor Fusion of Inertial, Sonar and Satellite Signals" Graduate Thesis. Dept. of Information technology.,Uppsala University.,Sweden,2015.

[5] N. Yadav, and C. Bleakley, "Two Stage Kalman Filtering for Position Estimation Using Dual Inertial Measurement Units." In Proc. 2011 IEEE Sensors ,2011,pp 1433 – 1436. http://dx.doi.org/10.1109/icsens.2011.6127064

[6] M. Bošnak, M., Matiko, D., & S. Blažic,.. "Quadrocopter Hovering Using Position-estimation Information from Inertial Sensors and a High-delay Video System",Journal of Intelligent Robotic System, pp. 43-60, vol. 67, New York: Springer, 2012.

[7] B. McCarron., "Low-Cost IMU Implementation via Sensor Fusion Algorithms in the Arduino Environment",Undergraduate Thesis, Aerospace Engineering Department, California Polytechnic State University, California,2013

[8] K. Kumar, A, Varghese,P. Reddy,N. Narendra, P. Swamy, M. Chandra, ;and P. Balamuralidhar, "An Improved Tracking Using IMU and Vision Fusion for Mobile Augmented Reality Applications." International Journal of Multimedia & Its Applications (IJMA) pp. 45-60, Vol.6, No.5,India:AIRCC Publishing,2014

[9] G. Tournier, M. Valenti, , and J. How. "Estimation and Control of a Quadrotor Vehicle Using Monocular Vision and Moire Patterns." In Proc. AIAA Guidance, Navigation and Control Conference, 2006,pp. 2006--6711. http://dx.doi.org/10.2514/6.2006-6711

Christian B. Abelanosa. The author is a graduate of Electronics Engineering in Mindanao University of Science and Technology In Cagayan De Oro City, Philippines.His research interest is on robotics and machine learning algorithms.

Ruth Pearl J. Lugpatan. The author is a graduate of Electronics Engineering in Mindanao University of Science and Technology In Cagayan De Oro City, Philippines.She's an active member of Junior Institute of Electronics Engineering of the Philippines – Northern Mindanao Sector. Her research interest is on sensors and artificial intelligence.

Diogenes Armando D. Pascua. The author is an active member of The Institute of Electronics Engineers of The Philippines .He was born in June 12, 1976 In Iligan City, Philippines. He earned his undergraduate degree of Electronics Engineering from Mindanao State University-Iligan Institute of Technology(MSU-IIT) in 1998.He Finished his Masters of Science in Computer Applications from the same university in year 2014.
He is a current faculty of the Electronics Engineering Dept. of the Mindanao University of Science and Technology in Cagayan de Oro City, Philippines. His research interest is on Embedded Computing and Robotics. He published a paper entitled: Development of A Mobile Robot as a Test bed for Telepresentation in The International Journal of Smart Materials and Mechatronics, Hasanuddin University,Indonesia in 2014.This paper was his graduate thesis in MSU-IIT where it was awarded as the Institute Best Thesis in 2014.