Embedded Kalman Filter for Inertial Measurement Unit (IMU) on the ATMega8535

Conference Paper · July 2012

DOI: 10.1109/INISTA.2012.6246978

3 authors:

Hany Ferdinando
University of Oulu

Handry Khoswanto
Petra Christian University

Djoko Purwanto
Institut Teknologi Sepuluh Nopember

Embedded Kalman Filter For Inertial Measurement Unit (IMU) on the Atmega8535

Hany Ferdinando, Handry Khoswanto
Department of Electrical Engineering
Universitas Kristen Petra
Surabaya, Indonesia
hanyf@petra.ac.id

Djoko Purwanto
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia

***Abstract*—The Kalman Filter is very useful in prediction and estimation. In this paper, the Kalman Filter is implemented for Inertial Measurement Unit (IMU) on the ATMega8535. The sensors used in this system are accelerometer MMA7260QT and gyroscope GS-12. The system chooses the arbitrary sampling time and then it is evaluated for possible using smaller value. As the Kalman Filter operation needs matrix calculation, the formula is converted into several ordinary equations. The parameter being investigated in this paper is measurement covariance matrix. This parameter influences the way the Kalman Filter responses to noise. Bigger value makes the Kalman Filter less sensitive to noise and the estimation is too smooth, thus it does not give real angle estimation. Using smaller value makes the Kalman Filter more sensitive to noise. This makes the estimated angle still suffers from noise and it is likely that the Kalman Filter is useless. This paper recommends 0.0001 to 0.001 for the measurement covariance noise parameter. This paper also recommended a pipeline configuration if the control algorithm needs more space in a sampling time.**

Keywords-Kalman Filter; IMU; accelerometer; gyroscope

I. INTRODUCTION

In a two-wheeled balancing robot, it is important to detect the tilt of the platform in order to correct its position. An accelerometer as used in [1] cannot be used to detect tilt correctly. The signal is noisy although the platform is still.

The noisy signal from accelerometer is common to all accelerometers. So, more sophisticated method to measure tilt position is needed.

For this purpose, the system needs additional sensor, i.e. gyroscope. This sensor is used in airplane to detect the tilt position. Gyroscope usually has 3-axis, i.e. x, y and z axes.

Unfortunately, there is drift in the gyroscope operation [2]. This makes the gyroscope alone will not give exact tilt position for a long period.

The simplest implementation of integrating these two sensors are found in [3]. Here, the author used simple low pass filter (LPF) to handle the noise from accelerometer and take advantage of stable signal from gyroscope. Although this simple implementation might be helpful to measure tilt of a platform, it cannot handle the drift from the gyroscope. Besides that, when the coefficient is too small, it does not have any impact on general microcontroller. The general microcontroller only takes integer values.

Actually, the implementation in [3] can be simplified with average filter. But, the result of average filter is more or less similar to that of [3]

In order to give exact tilt position, the system needs a collaboration of accelerometer and gyroscope. To integrate these two sensors, a filter is used, i.e. Kalman Filter, developed by Rudolf Kalman.

The goal of this paper is to show that Inertial Measurement Unit (IMU) can be developed by combining accelerometer and gyroscope with Kalman Filter. Another goal is to evaluate some drawbacks in the implementation on microcontroller.

First, this paper discuss other all related things about accelerometer and gyroscope as Inertial Measurement Unit (IMU) and some ideas to handle noise in accelerometer signal. Since the fusion of two sensors use the Kalman Filter, brief implementations of this filter are elaborated. Next, the implementation of the Kalman Filter algorithm on ATMega8535 is shown. The results of experiments with parameter of the Kalman Filter follow the implementation. Last, some conclusions and further possible works are given to close this paper.

II. REVIEW OF PREVIOUS RESULTS

Another interesting approach is using Kalman Filter. Kalman Filter is useful to handle noisy signal [4]. The Kalman Filter also has such a prediction feature [4] that is useful in adjusting the gyroscope drift [5].

Actually, there is a single sensor for IMU, e.g. ADIS16334 Analog Devices [6]. Other manufacturers place accelerometer and gyroscope in one board, e.g. ADXL335 and IDG500 are placed in single board [7]. With a lot of sensor for IMU, user can select which sensor is more applicable for the research.

Most of the implementations of Kalman Filter are in computer. Only a few of the implementation are in microcontroller. Some examples of implementation in microcontroller are found in [8] with PIC32 and [9] with PIC16F877.

The main problem in the Kalman Filter implementation in microcontroller is the programming language. Since most of microcontroller compiler support C programming, the implementation becomes easier. So, instead of programming the Kalman Filter with low level language, the high level approach is used.

Reference [10] shows implementation of Kalman Filter in Arduino board. Although Arduino is based on AVR microcontroller, the programming style is different from the ordinary AVR microcontroller. This implementation clarifies that it is possible to have Kalman Filter run on a microcontroller. This is important since Kalman Filter has complex computational relative to a microcontroller.

Reference [11] gives an idea how the Kalman Filter works. This information will be used later in this research. Unfortunately, some parameters are still unknown. These are left for the programmer during implementation.

III. IMPLEMENTATION

The system uses ATMega8535 as the microcontroller. This microcontroller has 8 channel of 10-bit internal ADC and also serves as data processor for the signals from both accelerometer and gyroscope.

The system uses MMA7260QT as the accelerometer. The MMA7260QT is 3-axis accelerometer with several user's settings for various application. One important thing to operate this sensor is that the voltage supply is 3.3 volt. Using voltage supply higher than 3.3 volt will damage it [12]. It is configured with 1.5g sensitivity. It gives 800 mV/g for conversion factor.

The GS-12 is a gyroscope sensor used in Bioloid, a humanoid robot. It is 2-axis gyroscope which can detect angular speed up to $\pm 300^{o}$/s. If there is no angular movement, the GS-12 gives 1.23 volt. The output is between 0.23 volt and 2.23 volt. So 1.23 volt is at the center of its response.

Timing is very important in this application. The ATMega8535 reads signals from two sensors every 5 ms. This value is determined arbitrarily and will be evaluated later. 5 ms is fast enough for two-wheeled balancing robot application as in [1].

Since both sensors have enough voltage level for the ADC, they do not require signal conditioners. This makes the implementation simple enough. Besides, the reference signal for ADC is 2.56 volt. This makes the implementation even much simpler.

Reading values via ADC need conversion from the 10-bit binary to real acceleration and angular speed for accelerometer and gyroscope respectively. These values become inputs for the Kalman Filter.

The implementation of Kalman Filter on the ATMega8535 is based on [11]. Unfortunately, it gives no all parameters for the implementation but there is an explanation how to choose the parameters.

Given a linear system, a state space for the system might be written as

$$x_{k+1} = Ax_k + Bu_k \tag{1}$$

In Kalman Filter, (1) will be changed into

$$\begin{pmatrix} \alpha \\ b \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & -dt \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ b \end{pmatrix}_k + \begin{pmatrix} dt \\ 0 \end{pmatrix} u_k \tag{2}$$

u is reading from the gyroscope. α and b are chosen arbitrary for the first time. This will bring no problem because it will be corrected later. Parameter α is estimated angle and β is gyro bias. Estimated angle has to consider the gyro and its bias in order to get the correct angle.

The most important parameter within this project is the error estimation for measurement covariance noise. This value gives such a constraint how much the jitter we expect on our accelerometer data [11]. The experiments will show this phenomenon.

Since the ATMega8535 is general purpose microcontroller, the matrices operation must be converted into several ordinary mathematics equations. This not only simplifies the operation bit also reduces the complexity of the calculation.

Equation (2) will be converted into

$$\alpha_{k+1} = \alpha_k - b \cdot dt + u_k \cdot dt \tag{3a}$$

$$b_{k+1} = b_k \tag{3b}$$

Equations (3a) might be written as

$$\alpha_{k+1} = \alpha_k - (u_k - b_k) \cdot dt \tag{3c}$$

IV. EXPERIMENTS AND RESULTS

Before implementing the code in the ATMega8535, it is simulated within Matlab first. Figure 1 shows the Matlab simulation with 0.001 as measurement covariance noise.

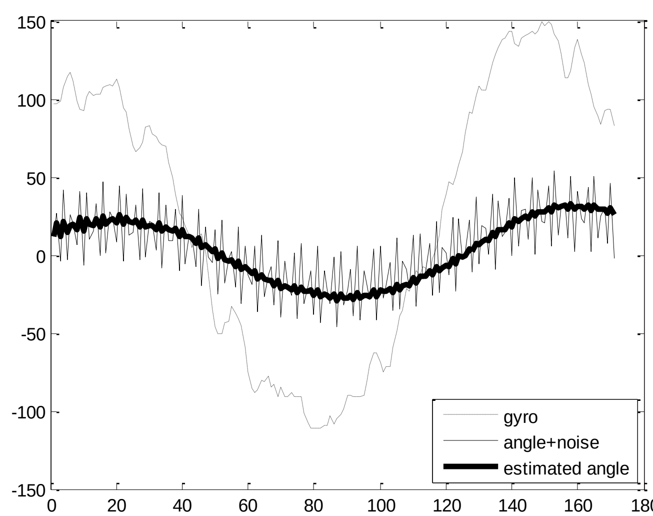

Figure 1. Matlab simulation with 0.001 as measurement covariance noise

Roughly, the code is implemented well as the measurement from MMA7260QT (solid thin line) suffers from noise; the Kalman Filter successfully suppresses it (solid thick line). Figure 1 shows that the Kalman Filter work as an average filter. So, for some application, we can use average filter instead.

One important point of using the Kalman Filter is to compensate the gyroscope drift. Since the bias cannot be measured directly, the Kalman Filter estimates it. This value then can be used to adjust the gyroscope signal. For this reason, it is interesting to see the estimated bias.

Figure 2 shows the estimated bias from the same parameter as in figure 1. It is shown that the estimated bias is under controlled, for it converges instead of diverges.

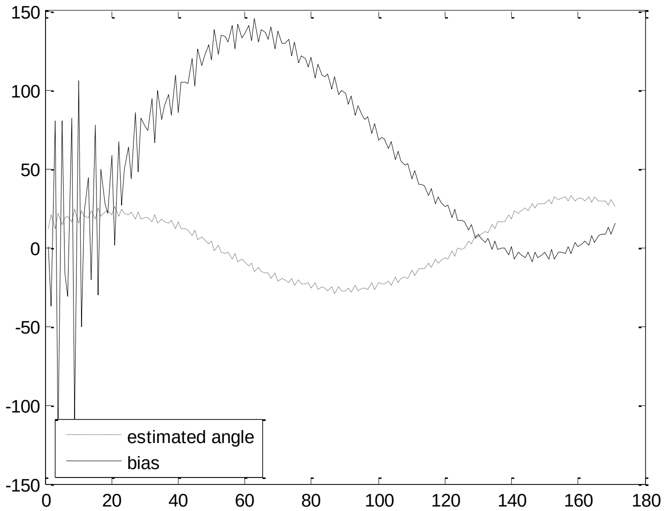

Figure 2. Estimated bias based on Matlab simulation

Figure 3 to 6 is the Kalman Filter response from the ATMega8535. The ATMega8535 supplies data for gyro reading, angle calculation from the accelerometer and estimated angle from the Kalman Filter operation. The data is imported to and plotted in Matlab.

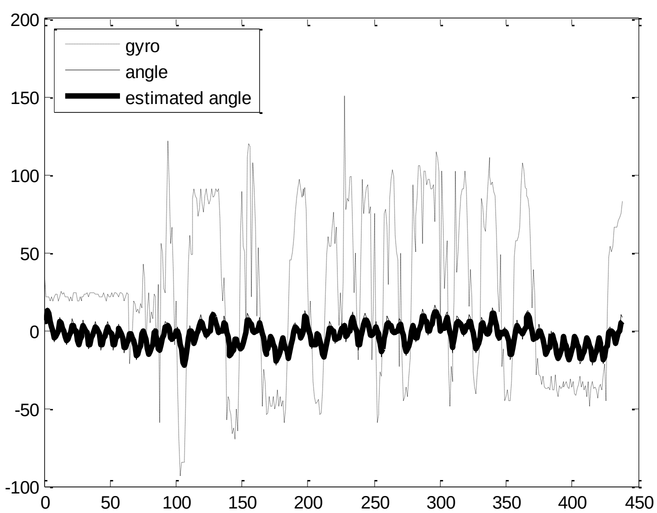

Figure 3. Estimated angle from the Kalman Filter with 0.00001 as measurement covariance noise

Figure 3 shows how the Kalman Filter estimates the angle based on the information from accelerometer and gyroscope. This experiment uses 0.00001 as the measurement covariance noise.

The Kalman Filter successfully estimated angle (solid thick line) from accelerometer (solid thin line) and gyroscope (dashed line). The problem is accelerometer signal cannot be seen because the estimated angle overrides it. The estimated angle still suffers from noise. So, higher value for measurement covariance noise is needed.

Figure 4 shows the experiment with different measurement covariance noise, i.e. 0.0001. Here, the accelerometer signal is seen but not much. Compare to figure 3, the estimated angle in figure 4 has less noise. So the measurement covariance noise takes its role here.

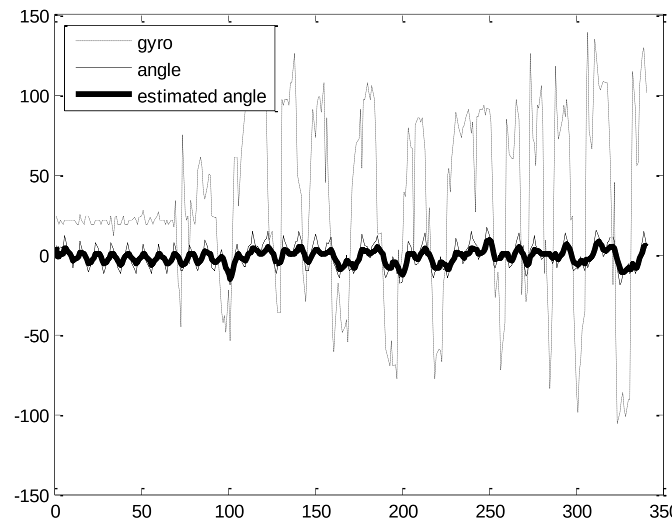

Figure 4. Estimated angle from the Kalman Filter with 0.0001 as measuremen covariance noise

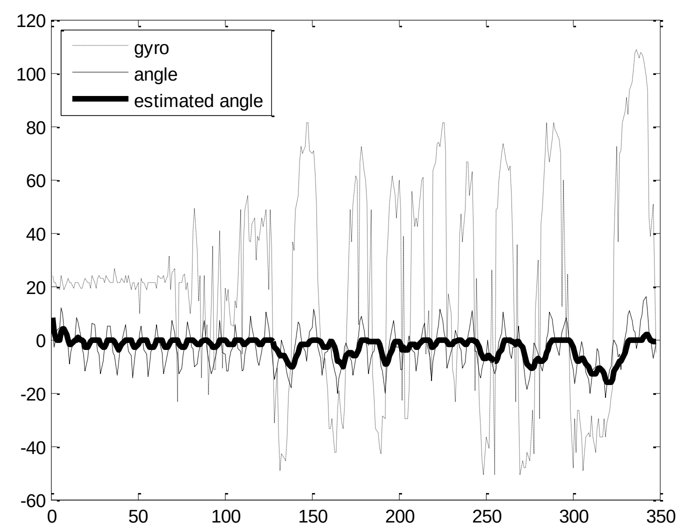

Figure 5. Estimated angle from the Kalman Filter with 0.001 as measuremen covariance noise

The estimated angle from figure 4 is better than figure 3. Here, the solid line (angle from accelerometer) and thick line (estimated angle from the Kalman Filter) can be seen although it is little bit difficult.

It is interesting to see the result for greater measurement covariance noise. Figure 5 shows the experiment with 0.001.

From figure 5, we can see the accelerometer signal more clearly. The estimated angle has less noise than that of figure 4.

Figure 6 shows that the value of measurement covariance noise has certain limit. It looks like that the 0.01 is not good enough to estimate the angle although the noise is suppressed successfully. The estimated angle in figure 6 is not acceptable. It is less sensitive to the noise and the estimation has big error.

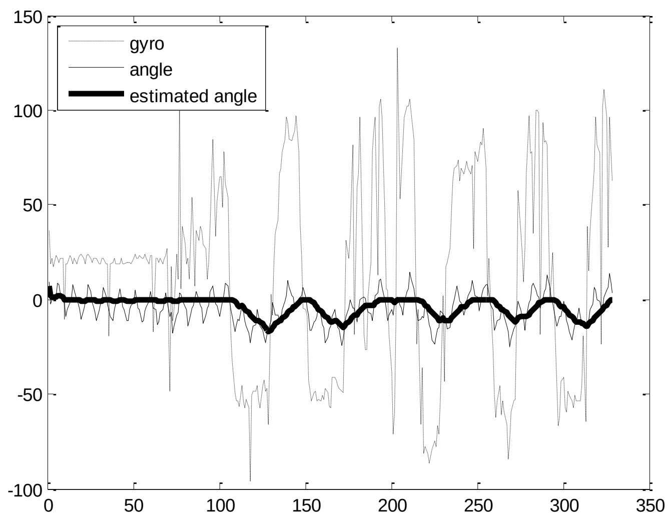

Figure 6. Estimated angle from the Kalman Filter with 0.01 as measuremen covariance noise

Implementing the Kalman Filter calculation on the ATMega8535 needs careful calculation related to the sampling time. The arbitrary chosen sampling time, i.e. 5 ms, is enough for the Kalman Filter operation.

To see whether the system can use smaller sampling time, the sampling time is reduced to 2.5 ms. There is no problem in reading the values from ADC but the remaining time is not enough for the Kalman Filter operation. So, it is likely 5 ms is the best sampling time for this application.

V. DISCUSSIONS

From figure 3 to 6, we can see that the measurement covariance noise parameter has important role in the Kalman Filter operation for IMU application. Adjusting the value of this parameter enables us to gain different result.

As this parameter is getting higher and higher, the noise on the estimated angle is reduced. This parameter enables us to set how much the jitter we expect on our accelerometer data. Bigger value makes the system less sensitive to the noise. This results a free-noise estimation angle. On the contrary, small value pushes the system to more sensitive to noise. The system cannot neglect the noise in the signal and the estimated signal still suffers from noise.

From the experiments, this project recommended to use measurement covariance noise parameter between 0.0001 and 0.001. Using value in this range makes the system not too sensitive to noise and the estimated angle is still acceptable.

Another important result is sampling time for the system. Since the system is two-wheeled balancing robot, it needs fast response. For this reason, small sampling time is necessary. The arbitrary chosen value, i.e. 5 ms, is good enough. When this sampling time is change to 2.5 ms, the result is not good. The Kalman Filter operation does not have enough time to finish the calculation.

As the time sampling is also used for the controller calculation, then 5 ms is enough if the controller is PID Controller. When the controller is Fuzzy Logic, then it is necessary to make the sampling time little bit higher.

Another interesting solution is to make such a pipe line for this process. The first ATMega8535 reads signal and then estimate the angle with the Kalman Filter. The estimation result is used by the next ATMega8535 in order to determine the control action. So the second ATMega8535 does the controller calculation with Fuzzy Logic or other control algorithms with long process.

VI. CONCLUSIONS

Although the Kalman Filter operation needs matrix calculation, the ATMega8535 can handle it by converting the matrix operation into several ordinary equations. This interesting result opens for more complex matrix operation to be implemented on the ATMega8535 as long as the matrix operation can be converted into several ordinary equations.

This research recommends using measurement covariance noise between 0.0001 and 0.001. Value in this range makes the Kalman Filter not too sensitive to noise but the estimation angle is still acceptable.

From the experiments, it is recommended using 5 ms as time sampling of the system. Smaller time sampling is not recommended because the Kalman Filter will not have enough space to play. Another reason is 5 ms still gives more space for controller calculation for PID controller. Using more sophisticated control algorithm with more complex operation, e.g. Fuzzy Logic, seems not fit within 5 ms. In order to implement Fuzzy Logic controller and the Kalman Filter on the ATMega8535, the pipeline method can be a solution.

ACKNOWLEDGMENT

The authors would like to thank to Directorate General of Higher Education, Indonesia for funding this research through Research Grant program number 526/E5.2/PL/2011 and supporting this presentation through Travel Grant program of 2012.

REFERENCES

[1] Hany Ferdinando, Handry Khoswanto, Djoko Purwanto, and Stefanus Tjokro Laksmono, "Design and Evaluation of Two-wheeled Balancing Robot Chassis: Case Study for Lego Bricks," in *Proc. International Symp. on Innovations in Intelligent Systems and Applications*, Istanbul, 2011, pp. 514-518.

[2] Johann Borenstein, Lauro Ojeda, and Surat Kwanmuang, "Heuristic Reduction of Gyro Drift for Personnel Tracking Systems," *Journal of Navigation*, vol. 62, no. 1, pp. 41-48, January 2009.

[3] Shane Colton, "The Balance Filter: a Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform", Massachusetts Institute of Technology, Cambridge, MA, Rev.1: Submitted as a Chief Delphi white paper, June 2007.

[4] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter," University of North Carolina at Chapel Hill, Chapel Hill, Lecture Notes TR 95-041, 2006.

[5] Hyung-Jik Lee and Seul Jung, "Gyro Sensor Drift Compensation by Kalman Filter to Control a Mobile Inverted Pendulum Robot System," in *IEEE Int. Conf. on Industrial Technology (ICIT) 2009*, Gippsland, VIC, 2009, pp. 1-6.

[6] Analog Devices. (2011, Jan 4). ADIS16334: Low Profile Six Degree of Freedom Inertial Sensor. Document. [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/ADIS16334.pdf

[7] Sparkfun Electronics. (2008, April 27) IMU Analog Combo Board - 5 Degrees of Freedom IDG500/ADXL335. [Online]. Available: http://www.sparkfun.com/products/9268

[8] José Luis Corona Miranda, "Application of Kalman Filtering and PID Control for Direct Inverted Pendulum," M.S. Thesis, Department of Electrical and Computer Engineering, California State University, Chico,USA, 2009.

[9] Chirayu Shah, "Sensorless Control of Stepper Motor Using Kalman Filter," Bachelor Thesis, Department of Electrical Engineering, Cleveland State University, Cleveland, Ohio, USA, 2004.

[10] Starlino. (2010, January 17) Arduino code for simplified Kalman filter Using a 5DOF IMU (accelerometer and gyroscope combo). [Online]. Available: http://letsmakerobots.com/node/15688

[11] Tom Pycke. (2006, May 10) Kalman filtering of IMU data. [Online]. Available: http://tom.pycke.be/mav/71/kalman-filtering-of-imu-data

[12] Freescale Semiconductor. (2010, March 28). MMA7260QT. [Online]. Available: http://www.freescale.com/files/sensors/doc/data_sheet/MMA7260QT.pdf