Calibration against Orientation Drifts in a Real-time Embedded Inertial Measurement Unit

Bui Hong Hue*, Nguyen Ngoc Hung†, Tran Xuan Kien‡, Do Duc Hanh§, Doan Minh Dinh§,
Le Ba Yen¶ and Le Danh Tuan¶
*College of Urban Works Construction
Ministry of Construction, Ha Noi, Viet Nam
Email:ngochuy.buihue@gmail.com
†Department of Control Engineering
Military Technical Academy, Hanoi, Vietnam
Email: ngochung212@gmail.com
‡Institute of Electronics
Military Science and Technology Institute, Hanoi, Vietnam
Email: txkien2003@yahoo.com
§General Technical Department
Ministry of Defence, Hanoi, Vietnam
Email: hanh@aitcv.ac.vn, doanminhdinh@vnn.vn
¶Institute of Automation
Military Science and Technology Institute, Hanoi, Vietnam
Email: ledanhtuan@gmail.com, yenlb.tdh@gmail.com

***Abstract*—This paper presents the development and implementation of an Inertial Measurement Unit (IMU) where Micro-Electro-Mechanical System (MEMS) sensors are used to measure angular velocity and acceleration, a magnetic sensor (magnetometer) is used to calibrate against orientation drifts, a Global Positioning System (GPS) signal receiver is integrated and an extended Kalman filter algorithm is applied for real time signal processing to estimate the attitude of an object in space. The MATLAB/Simulink Embedded tool and C30/MPLAB Compiler are used to design, compile, and download directly into the target. An update rate of 100Hz with real time floating point processing can be achieved for the extended Kalman algorithm using a Microchip 16 bit dsPIC33f256 microcontroller.**

Keywords—IMU; MEMS Sensor; Extended Kalman Filter; Attitude Estimation; Embedded System

I. INTRODUCTION

UNMANNED Aerial Vehicles (UAVs) is a rapidly growing area of research and development. Navigation is an important task of UAVs and can be defined as the process of determining their positions, orientations and velocities. One of the biggest technical barriers is the accurate estimation of object's positions in space [1]. Although GPS sensors can provide fast position information with drift free to the navigation system at all time, it requires direct lines of the signal from at least four GPS satellites. GPS signals can be often interrupted by land-based objects, e.g. high buildings, mountains, or other obstacles. When the receiver lost GPS signals from satellites, it takes a significant time to reconnect and to resolve the ambiguities of phase measurement. In this case, the navigation system can use information from inertial sensors (gyroscopes, accelerometers, etc.) and magnetometers to estimate orientation and position parameters.

The accuracy of an estimator is strongly affected by the bias drift on each axis and by the noises from its sensors and platform. Vibration of the vehicle also makes some noises. These noises need to be filtered out. Typically gyroscopes can be used to calculate the attitude of the object by integrating their signals. Nowadays, low-cost IMU modules could be developed using cheap MEMS inertial sensors combined with high speed digital signal processors (DSPs) or microcontrollers with digital signal processing support [2], [3], [4]. However, MEMS sensors have more noises and bias drifts that need to be eliminated since the accumulation of them could make a reasonable error over time.

The design of such attitude estimation requires either an expensive commercial module or significant knowledge of electronics, attitude dynamics, and Kalman filtering [5], [6], [7]. This paper presents in detail the development and implementation of a low-cost IMU module where a 10-state extended Kalman filter (4 quaternions, 3 angle velocities, 3 angle velocity biases) with MEMS sensors (accelerometers and gyroscopes) are applied to determine yaw, pitch and roll angles of the vehicle for navigation purposes. To increase the accuracy of angle estimation, a calibration process should be integrated into Kalman filtering algorithm to provide higher accurate orientation angle results from information acquired from magnetometers and accelerometers.

The rest of the paper is organized as follows. Basic concepts of calibration are presented in Section II. Section III presents extended Kalman filters and attitude dynamics. This section is a shortened version of Section II of [7] and is rewritten for convenience. The IMU implementation is introduced in Section IV. The results of performance testing are given in Section V. We conclude in Section VI.

978-1-4799-2903-0/14/$31.00

II. CALIBRATION OF MAGNETIC SENSORS

Orientation angles can be estimated for moving objects with GPS signals from GPSs, Glonass, Galileo, Beidou systems. When GPS signals absent, magnetometers (that sense Earth's geomagnetic field strength) can be utilized to determine absolute orientation angles via reference to the local magnetic North. The difference between the absolute North and local North is considered as magnetic declination and is a function of latitude, longitude and time using a global model such as World Magnetic Model (WMM). However, measured geomagnetic amplitude varies due to magnetic storms or other magnetic components e.g. vehicles, buildings, bridges, electric power lines, etc. As a consequence, the reference local north orientation is misaligned to magnetic forces in the horizontal plane.

The outputs of magnetometers depend on the tilt compensation (inaccurate horizontal plane estimates) and the calibration process including identification of error sources and noise eliminations from measurement outputs. The traditional automatic calibration is based on the fact that the locus of precise geomagnetic measurement is a circle if the sensor moves around a circle. The impact of various magnetometer errors would distort the shape of the locus. The circular constraint, therefore, can be utilized to partially estimate the local variations of the Earth's magnetic field strength.

In practice, auto-calibration implementation does not require any reference orientation angle, but orientation of the moving object depends on the positions and efficiency of magnetic field components in the proximity of the sensor. When sensors are first installed in a new environment, a calibration procedure needs to be done in advance to eliminate or compensate the biases and the effects of the new magnetic field environment.

Magnetometers' outputs $M_x^{hc}, M_y^{hc}, M_z^{hc}$ are represented as:

$$\begin{aligned} M_x^{hc} &= S_x M_x^{do} + B_x \\ M_y^{hc} &= S_y M_y^{do} + B_y \\ M_z^{hc} &= S_z M_z^{do} + B_z \end{aligned} \tag{1}$$

where S_x, S_y, S_z are scale factors, B_x, B_y, B_z are offsets along the horizontal axes of the magnetic field, and $M_x^{do}, M_y^{do}, M_z^{do}$ are the measured magnetic components of the Earth. Let $M_{xmax}^{do}, M_{xmin}^{do}, M_{ymax}^{do}, M_{ymin}^{do}, M_{zmax}^{do}, M_{zmin}^{do}$, be maximal, minimal values of measured magnetic fields along $x-$, $y-$ and $z-$ axes, respectively. Then the scale factors and offset values are calculated by using the following equations:

$$\begin{aligned} S_x &= MAX(1, \tfrac{M_{ymax}^{do}-M_{ymin}^{do}}{M_{xmax}^{do}-M_{xmin}^{do}}, \tfrac{M_{ymax}^{do}-M_{ymin}^{do}}{M_{xmax}^{do}-M_{xmin}^{do}}) \\ S_y &= MAX(1, \tfrac{M_{xmax}^{do}-M_{xmin}^{do}}{M_{ymax}^{do}-M_{ymin}^{do}}, \tfrac{M_{zmax}^{do}-M_{zmin}^{do}}{M_{ymax}^{do}-M_{ymin}^{do}}) \\ S_z &= MAX(1, \tfrac{M_{xmax}^{do}-M_{xmin}^{do}}{M_{zmax}^{do}-M_{zmin}^{do}}, \tfrac{M_{ymax}^{do}-M_{ymin}^{do}}{M_{zmax}^{do}-M_{zmin}^{do}}) \end{aligned} \tag{2}$$

$$\begin{aligned} B_x &= (\tfrac{M_{xmax}^{do}-M_{xmin}^{do}}{2} - M_{xmax}^{do}).S_x \\ B_y &= (\tfrac{M_{ymax}^{do}-M_{ymin}^{do}}{2} - M_{ymax}^{do}).S_y \\ B_z &= (\tfrac{M_{zmax}^{do}-M_{zmin}^{do}}{2} - M_{zmax}^{do}).S_z \end{aligned} \tag{3}$$

Scale factors can be determined by the ratios of the major and minor axes of eclipses formed by the locus of magnetic measurements. Bias parameters are defined as the offsets of the eclipse's centre. With these parameters, the calibration value for the each component of the magnetometers can be calculated using equation (2). To apply equations (1) and (2) in adjusting the measured magnetic values, we need to rotate the sensors several rounds (about 360^0), then specify the measured values like in the first row of the Table I and calibrate sensor outputs using equation (3). Applying scale factors and offsets in Table I for equation (1), it is possible to transform $M_{x,y,z}^{do}$ in a distorted shape into spheric $M_{x,y,z}^{hc}$.

TABLE I. PREDEFINED CALIBRARION PARAMETERS FOR HMC5833L SENSOR

M_{xmax}^{do}	M_{xmin}^{do}	M_{ymax}^{do}	M_{ymin}^{do}	M_{zmax}^{do}	M_{zmin}^{do}
-3130(nT)	3070(nT)	-2750(nT)	3240(nT)	-2960(nT)	3940 (nT)
S_x	B_x	S_y	B_y	S_z	S_z
1,119	-3,33	1,15	-27,6	1	-51

Table I provides the measured values, scale factors and bias values of magnetometer Honeywell HMC5883L, installed on the sensor system GY80. These values are given to the Kalman filter simulink block in Section IV.

III. STATE ESTIMATION AND EXTENDED KALMAN FILTER

A. Extended Kalman Filter

The expectation value of state error $\Delta\hat{x}_k$ is known as the covariance of the states expressed by the following equation:

$$P_k = \varepsilon([\vec{x}_k - \hat{x}_k][\vec{x}_k - \hat{x}_k]^T) \tag{4}$$

An optimal feedback gain K_k can be derived by minimizing P_k. For linear systems this is known as the Kalman filter and was first presented in 1960 by Kalman [6]. The estimated state and covariance are propagated at every time step, but in many systems, measurements are only sporadically available. In this case the state covariance grows for many time steps as the estimated state diverges from the true state. When a measurement becomes available, the feedback gain K_k is calculated, the state is corrected, and the state covariance is updated (reduced) accordingly. The Kalman filter has been applied to nonlinear systems in many different ways. One of the most straightforward techniques is to linearize the system dynamics and the measurement function around the expected state $\vec{x}_k(-)$, and then apply the Kalman filter as normal. This is known as extended Kalman filtering and the implementation equations (borrowed from [5]) are presented in Table II. Extended Kalman filter allows to dynamically estimate attitude and drifting gyro biases, as long as they are modeled as states in our dynamical system. Next the three-dimensional attitude dynamics and measurement equations will be derived.

B. Attitude Dynamics Equations

Suppose there is a rigid body with the body coordinate system OX'Y'Z' and a reference coordinate system OXYZ (e.g. the Earth) (Fig. 1). Orientation angles are defined by the angles between each axis couples OX, OY, OZ and OX', OY', OZ', respectively, when the origin of the reference coordinate system is translated to the origin of the body coordinate system

TABLE II. EXTENDED KALMAN FILTER IMPLEMENTATION EQUATIONS

System dynamics	$\vec{x}_k = f_{k-1}(\vec{x}_{k-1}) + \vec{\omega}_{k-1}$	$(k1)$
Measurement	$\vec{z}_k = h_k(\vec{x}_k) + \vec{v}_k$	$(k2)$
Covariance	$P_k = \varepsilon([\vec{x}_k - \hat{x}_k][\vec{x}_k - \hat{x}_k]^T)$	$(k3)$
State propagation	$\hat{x}_k(-) = \Phi_{k-1}(\hat{x}_{k-1}(+))$	$(k4)$
Dynamics linearization	$\Phi_{k-1} = \frac{\partial f_{k-1}}{\partial x}\Big\vert_{x=\hat{x}_{k-1}(+)}$	$(k5)$
Predicted measurement	$\vec{z}_k = h_k \hat{x}_k(-)$	$(k6)$
Measurement linearization	$H_k = \frac{\partial h_k}{\partial x}\Big\vert_{x=\hat{x}_k(-)}$	$(k7)$
State covariance propagation	$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi^T{}_{k-1} + Q_{k-1}$	$(k8)$
Feedback gain	$K_k = P_k(-) H_k^T H_k P_k(-) H_k^T + R_k$	$(k9)$
State update	$\hat{x}_k(+) = \hat{x}_k(-) + K_k(z_k - \hat{z}_k)$	$(k10)$
State covariance update	$P_k(+) = (1 - K_k H_k) P_k(-)$	$(k11)$

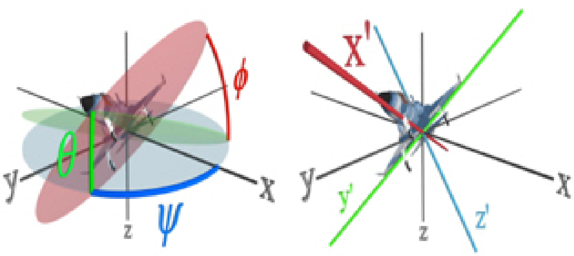

Fig. 1. Body coordinate system and reference coordinate system

preserving the directions of OX, OY and OZ unchanged. The angles representing the relationship between these two coordinate systems are called attitude Euler angles including yaw (ψ), pitch (θ) and roll (φ) angles. From orientation angles, the direction cosine matrix (DCM) between two coordinate systems is calculated as follows [3]:

$$\begin{bmatrix} u_b \\ v_b \\ w_b \end{bmatrix} = T(\varphi, \theta, \psi) \begin{bmatrix} u_e \\ v_e \\ w_e \end{bmatrix} \tag{5}$$

where

$$T = \begin{bmatrix} cos\psi cos\theta & \sin\theta & \sin\psi cos\theta \\ -cos\psi \sin\theta cos\varphi + \sin\psi \sin\varphi & cos\theta cos\varphi & \sin\psi \sin\theta cos\varphi + \cos\psi \sin\varphi \\ cos\psi \sin\theta \sin\varphi + \sin\psi cos\varphi & cos\theta \sin\varphi & \sin\psi \sin\theta \sin\varphi + \cos\psi cos\varphi \end{bmatrix}$$

$[u_e\, v_e w_e]^T$ is a vector in the reference frame, and $[u_b v_b w_b]^T$ is the vector in the rigid body frame.

It turns out that the Euler angle dynamics can be derived independently. Because the body rotation rates $\vec{\omega} = [\omega_x\, \omega_y \omega_z]$ are defined as instantaneous rotation rates about the OX, OY, and OZ axes, respectively, it can be observed that $\dot{\phi}$ is a rotation about the OX axis. $\dot{\theta}$ is a rotation about an intermediate axis, so to project $\dot{\theta}$ onto the body frame it must be rotated using the ϕ rotation. Likewise, $\dot{\psi}$ must be rotated about first θ, and then ϕ, i.e.

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + T_\phi \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + T_\phi T_\theta \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{6}$$

From (5) and (6) we obtain:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{7}$$

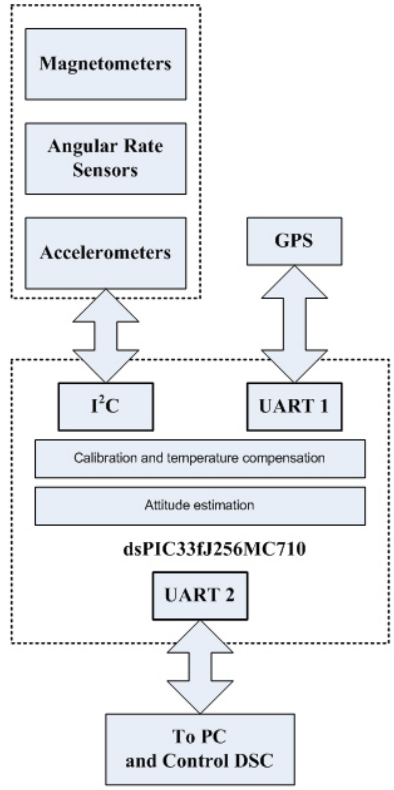

Fig. 2. IMU Block Diagram

Quaternion is hyper-complex numbers of rank 4. A quaternion q can be expressed as:

$$q = q_0 + iq_1 + jq_2 + kq_3$$

Then the DCM is a function of q:

$$\begin{bmatrix} u_b \\ v_b \\ w_b \end{bmatrix} = T(q) \begin{bmatrix} u_e \\ v_e \\ w_e \end{bmatrix} \tag{8}$$

where

$$T = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1 q_2 + 2q_0 q_3 & 2q_1 q_3 - 2q_0 q_2 \\ 2q_1 q_2 - 2q_0 q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2 q_3 + 2q_0 q_1 \\ 2q_1 q_3 + 2q_0 q_2 & 2q_2 q_3 - 2q_0 q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}$$

IV. IMU MODULE IMPLEMENTATION

A. Hardware Design

The IMU module contains a dsPIC33f-based board and auxiliary blocks including power supply, blocks for communication and interface with MEMS sensors (using I^2C protocol standard), GPS (COM1) and data transmission of data to PC (COM2) (Fig. 2). The MEMS sensor block (of GY85 type) has sensors for angular velocity, acceleration and magnetic field strength sensors (magnetometers). The GPS block is of u-blox5S type. To test the implemented module for performance, the module is mounted on a 1573 Series Automatic Positioning and Rate Table System (the Test Table System) from the Ideal Aerosmith Corporation. The Test Table System is designed to provide precise and reliable positions (with accuracy of ± 20 arc sec), rates (with accuracy of 0.005%) and acceleration motion control (with complex motion profiles) for the production test and/or development of inertial packages and their components. It can be operated from the AERO 832 Controller front panel for local control or through a computer interface for remote control and is programmed with the Ideal Aerosmith Table Language (ATL) for remote operation.

B. Implementation Equations

The IMU module is developed using an extended Kalman filter with the full quaternions $[q_0\, q_1\, q_2\, q_3]^T$. The state is $\vec{x} = [q_0\, q_1\, q_2\, q_3\, \omega_1\, \omega_2\, \omega_3\, b_1\, b_2\, b_3]^T$ where b_i are the gyro biases. The implementation follows the equations summarized in Table II and it is necessary to derive the state transfer functions f and ϕ, the measurement functions h and H, and the noise covariance matrixes Q and R.

1) State transfer functions: Rotation angular velocities and drifts of the gyroscopes are assumed to be stochastic processes, i.e.

$$\begin{aligned} \vec{\omega}_k &= \vec{\omega}_{k-1} + \vec{\omega}_{\omega,k-1} \\ \vec{b}_k &= \vec{b}_{k-1} + \vec{b}_{b,k-1} \end{aligned} \tag{9}$$

The state $\vec{x}_{k+1}$ is represented as follows: State transfer function f is derived from the above equation and equation $(k1)$. From equation $(k7)$, ϕ is obtained by taking partial derivatives of f.

2) Measurement functions: Measurement of gravity and inertial acceleration in the body frame is expressed by:

$$\vec{z}_{accel} = T \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} - g \end{bmatrix} \tag{10}$$

Most efficient computation is with accurate tracking estimation, i.e. $\hat{q} \approx q$, and then subtracting the inertial accelerations measured by GPS block we obtain:

$$z_{accel} = T_q \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} - g \end{bmatrix} - T_{\hat{q}} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \approx T_q \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{11}$$

After coordinate transformation we obtain:

$$\begin{aligned} z_{aceel} &= \begin{bmatrix} 2q_0^2+2q_1^2-1 & 2q_1q_2+2q_0q_3 & 2q_1q_3-2q_0q_2 \\ 2q_1q_2-2q_0q_3 & 2q_0^2+2q_2^2-1 & 2q_2q_3+2q_0q_1 \\ 2q_1q_3+2q_0q_2 & 2q_2q_3-2q_0q_1 & 2q_0^2+2q_3^2-1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \\ &= \begin{bmatrix} -2g(q_1q_3+q_0q_2) \\ -2g(q_2q_3-q_0q_1) \\ -g(2q_0^2+2q_3^2-1) \end{bmatrix} \end{aligned} \tag{12}$$

Local (Earth) magnetic field is measured, given as

$$z_{mag} = \begin{bmatrix} 2q_0^2+2q_1^2-1 & 2q_1q_2+2q_0q_3 & 2q_1q_3-2q_0q_2 \\ 2q_1q_2-2q_0q_3 & 2q_0^2+2q_2^2-1 & 2q_2q_3+2q_0q_1 \\ 2q_1q_3+2q_0q_2 & 2q_2q_3-2q_0q_1 & 2q_0^2+2q_3^2-1 \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} \tag{13}$$

Measured body's angular velocities including drifts are given by:

$$\vec{z}_{\omega} = \begin{bmatrix} \omega_1 + b_1 \\ \omega_2 + b_2 \\ \omega_3 + b_3 \end{bmatrix} \tag{14}$$

Combining equations (12), (13), and (14), the full measurement function h is derived (by $(k8)$). From equation $(k9)$, the linearized measurement function H is obtained by taking partial derivatives of h, given as

$$H = \begin{bmatrix} 2gq_2 & -2gq_3 & 2gq_0 & -2gq_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2gq_1 & -2gq_0 & -2gq_3 & -2gq_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4gq_0 & 0 & 0 & -4gq_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4B_xq_0-2B_zq_2+2B_yq_3 & 4B_xq_1+2B_yq_2+2B_zq_3 & -2B_zq_0+2B_yq_1 & 2B_yq_0+2B_zq_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4B_yq_0+2B_zq_1-2B_xq_3 & 2B_zq_0+2B_xq_2 & 4B_yq_0+2B_zq_1-2B_xq_3 & -2B_xq_0+2B_zq_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4B_zq_0-2B_yq_1+2B_xq_2 & 2B_zq_0+2B_xq_2 & 2B_xq_0+2B_yq_3 & 4B_yq_0+2B_zq_1-2B_xq_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2B_yq_0+2B_xq_3 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

3) System and measurement noise covariances: The system noise covariance is assumed to be:

$$Q = \begin{bmatrix} \sigma_{q_0}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{q_1}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{q_2}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{q_3}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\omega_1}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\omega_2}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\omega_3}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{b_1}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{b_2}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{b_3}^2 \end{bmatrix} \tag{16}$$

where σ_q is the expected quaternion system noise, σ_ω is the random walk on the body rates, and σ_b is the random walk on the gyro biases. The sensor or measurement noise covariance is given as:

$$R = \begin{bmatrix} \sigma_a^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_a^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_a^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{mag}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{mag}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{mag}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\omega^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\omega^2 \end{bmatrix} \tag{17}$$

where σ_a, σ_{mag} , and σ_ω are the standard deviations of the expected noises of the accelerometers, magnetometers, and rate gyros, respectively.

Though this implementation is computationally simpler than an Euler angle direct implementation, it is still too intensive for a 16-bit microcontroller with floating point DSP at a reasonable update rate.

C. Software Design

The development and implementation of the algorithm in the previous section on a high speed dsPIC microcontroller are done using the compiler tool chain from Mathworks and Microchip (Fig. 3).

These tools allow convenient and fast design of computation and processing models with graphical design, drag and drop of blocks from Simulink built-in libraries, corresponding blocks for dsPIC available in MATLAB/Simulink and MATLAB functions (**.m* files, *s*-functions).

Simulink modules for simulation and generation of embedded application programs can be compiled and then downloaded directly into the dsPIC microcontroller target. Modules

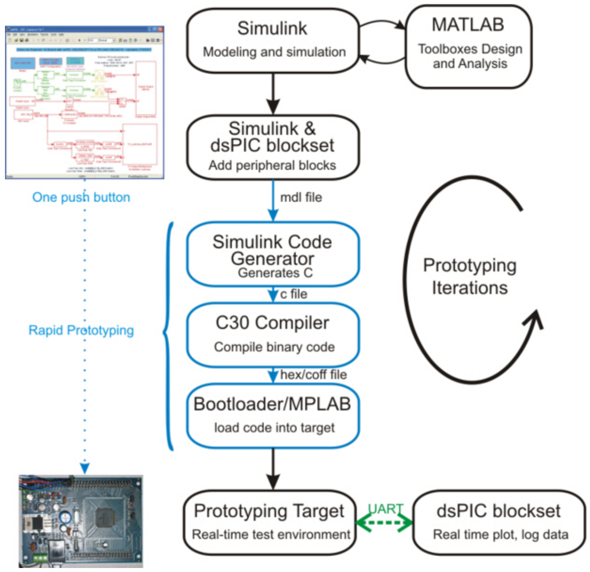

Fig. 3. Software design procedures

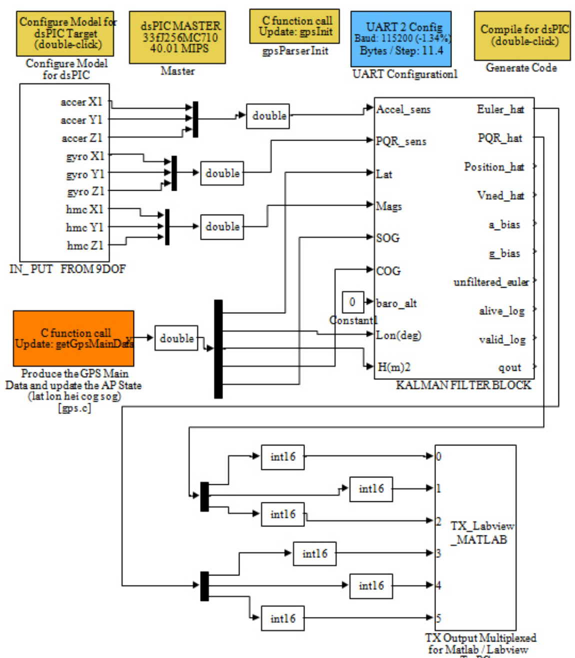

Fig. 4. Simulink implementation of our designed attitude estimation

for attitude estimation, gyro drift (biases) and quaternions are designed as shown in Fig. 4.

The main module contains a sub-module for reading data from MEMS sensors through I^2C interface, function for reading GPS data from COM1, a sub-module for Kalman filter and a sub-module for sending output data to PC through COM2.

After compiling with Real-time Workshop Embedded Coder, the code is downloaded into the target using a Microchip Development Programmer/Debugger PICKit 2.

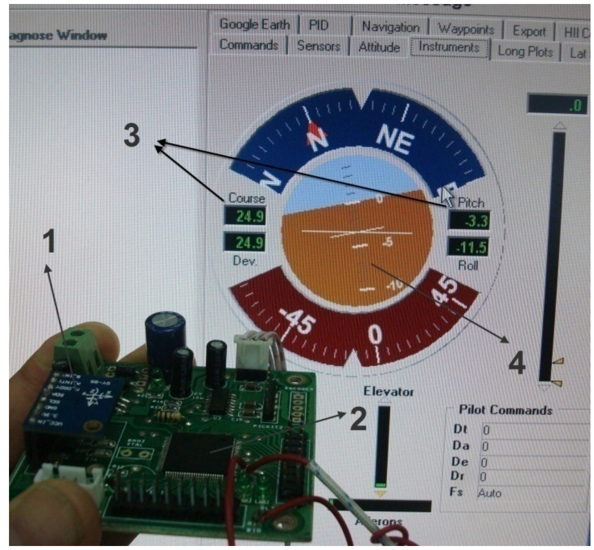

Fig. 5. Experimental results of object's rotation in space; 1 - MEMs GY80 sensor; 2 - Microcontroller of the designed IMU module; 3 - yaw, pitch and roll angles; 4 - Three Euler angles from the open-source software tool for UAV

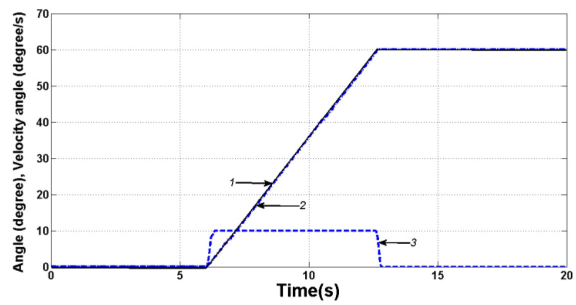

Fig. 6. Experiment results when rotating the object in space; 1 - Rotation angle of the Test Table System about Y axis; 2 - estimation angles about Y axis; 3 - estimation angular velocities about Y axis

V. PERFORMANCE TESTING

Fig. 5 illustrates the graphic user interface of a dedicated open-source UAV control tool for the designed IMU module, where the module (object) is rotated in space with roll, pitch and yaw angles at $-3.3^0, 24.9^0$ and -11.5^0, respectively.

The graph in Fig. 6 shows estimation angles (dashed line -2) and angular velocities (dashed line -3) obtained from the implemented module when the Test Table System rotates (with sinusoidal motion profile) about Z axis from 0 to 600 with rate of 100/s (solid line 1). After performing several experiments, it is figured out that the estimation angle static accuracy is statistically about 0.1 degrees. The update rate of 100Hz can be achieved.

VI. CONCLUSIONS

The experiment results show that our algorithm can calibrate outputs of magnetometers for estimating orientation angles when GPS signals lost. This paper is a continuation of [7]. The designed extended Kalman filter is implemented with the calibration procedure against orientation drifts for magnetometers in the dsPIC microcontroller, it guarantees real-time computation and floating point precision. The accuracy of orientation estimation is the same as those done in [7] despite magnetic noises from the surroundings. A high speed DSP processor can be used to increase the update rate. The designed module(IMU) is low cost and suitable for object attitude estimation in practical embedded applications. It can be used in miniature GPS/INS systems for autonomous vehicle navigation.

REFERENCES

[1] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods", *Journal of guidance control and dynamics* 30.1 (2007): 12.

[2] P. Batista, C. Silvestre, P. Oliveira, B. Cardeira., "Low-cost Attitude and Heading Reference System: Filter design and experimental evaluation", *Robotics and Automation (ICRA), 2010 IEEE International Conference*, pp. 2624-2629, May 2010.

[3] D. Gebre-Egziabher and G. Elkaim, "Mav attitude determination by vector matching", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, pp. 1012-1028, July 2008.

[4] H. Chao, C. Coopmans, L. Di, and Y. Q. Chen, "A comparative evaluation of low-cost IMUs for unmanned autonomous systems", in *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 10)*, pp. 211216, University of Utah, Salt Lake City, Utah, USA, September 2010.

[5] M. S. Grewal and A. P. Andrews, "Kalman Filtering: Theory and Practice Using MATLAB". Wiley-Interscience, January 2001.

[6] R. E. Kalman, "A new approach to linear Filtering and prediction problems", *Transactions of the ASME Journal of Basic Engineering*, No. 82 (Series D), pp. 35-45, 1960.

[7] B. H. Hue, T. X. Kien, D. M. Dinh, D. D. Hanh, "Model-based Development and Implementation of Real-time Object Spatial Attitude Estimation", *International Journal of Computer and Electrical Engineering (IJCEE)*, Vol. 5, No. 4, pp. 372-377, 2013.