A sensor fusion algorithm for an integrated angular position estimation with inertial measurement units

Simone Sabatelli[*], Francesco Sechi[*], Luca Fanucci[#], Alessandro Rocchi[*]

[#] *Dept. of Information Engineering, University of Pisa, Pisa, Italy*

[*] *SensorDynamics AG, Navacchio (Pisa), Italy*

***Abstract*—This work presents an orientation tracking system for 6D inertial measurements units. The system was modeled with MathWorks Simulink and experimentally tested with the Cube Demo board by SensorDynamics, used to simulate a 3D gyro and a 3D accelerometer. Quaternions were used to represent the angular position and an Extended Kalman filter was used for the sensor fusion algorithm. The goal was to obtain an integrated system that could be easily integrated within the logic of the new 6D sensor family produced by SensorDynamics. We propose a Kalman filter simplification for a fixed point arithmetic implementation to reduce the system complexity with negligible performance degradation.**

Keywords: orientation tracking; angular position; Kalman filter; quaternions; inertial measurement unit; sensor fusion.

I. Introduction

MEMS sensors are widely used in many consumer and automotive applications, due to their low cost, small size and low power consumption. However, there are also some disadvantages: MEMS sensors suffer from higher errors than other expensive sensors. In particular, gyro drift is a problem in many applications that require a high precision. In an orientation tracking system the problem is that, with the simple integration of the gyro data to obtain the angular position, the drift error is always increasing, so that after some time the information of the angular position is meaningless.

To avoid this problem, some other kind of sensors must be used in addition to gyros, such as an accelerometer or a magnetic compass, so as to obtain an external reference for the angular positioning system. There are many papers that propose sensor fusion algorithms for this purpose.

Almost all the papers refer to quaternions to represent the angular position, because the use of quaternions instead of the Euler angles eliminates the problems related with the singularities and the gimbal lock. Different types of Kalman filters and state equations are used to describe the system.

In [1], an Extended Kalman filter is used with a three rate gyro and three accelerometers. The state equation is composed by the quaternion and also by the angular velocity and the gyro drift. The system has a good estimation of roll and pitch angles, and there is also a low correction on yaw angle.

For a better estimation of the yaw angle, a magnetic compass, in addition to the gyro and the accelerometers, is used in [2]. The process is described with the quaternion and the angular velocities, and a linear Kalman filter is developed. In particular, to avoid the use of an Extended Kalman filter, the read equation of the filter is not composed by the data of the MARG sensor (magnetic, angular rate and gravity), but it is composed by a quaternion, which is estimated applying the Gauss-Newton method to the sensor data. This simplifies the filter design, but implies a quite complicated algorithm to estimate the quaternion.

In [3] and [4] an adaptive filter is developed for a MARG sensor unit for automotive use. When the sensor is in high acceleration mode, the angular position is calculated mainly with the data of the gyro, updating the quaternion. The accelerometers are used to estimate the roll and pitch angles, and the magnetic compass to estimate the yaw angle. The angle estimation is more influential in the state correction when the system is in non-acceleration mode.

In [5] an Unscented Kalman filter is used instead of the traditional Extended Kalman filter, because the Unscented filter is deemed to be more accurate and less costly to implement. In [6] there is a comparison between the Extended Kalman filter and the Unscented Kalman filter: the resulting precision is found comparable, but the Unscented filter requires much more computation time.

It is noteworthy that all these works refer to discrete sensor systems, where the filter algorithm is processed on a microcontroller or a PC. This work proposes the design of a simplified Kalman filter and relevant fixed point architecture to be integrated on silicon together with the 6D Inertial Measurement Unit (IMU) sensor.

In the following, section II describes how the system is modeled and the filter tuning, section III describes the simplifications proposed for the Kalman filter algorithm, in section IV are presented some experimental test results and in section V is estimated the number of bits needed for the implementation of a finite precision arithmetic.

II. System modelling

MathWorks Simulink was used to build a model of our system and to verify the correct operation of the filter. The state equation is composed by the quaternion only, because the addition of other variables does not increase the precision of the angular estimation significantly, but requires a lot more hardware requirements. The state equation is the following:

978-3-9810801-7-9/DATE11/©2011 EDAA

$$\dot{q}_n^b = \frac{1}{2}\Omega_{\mathrm{nb}}^{\mathrm{n}} q_n^b. \quad (1)$$

Where q_n^b is the quaternion representing the rotation of the body-frame, united to the IMU sensor, with respect to the inertial n-frame, and $\Omega_{\mathrm{nb}}^{\mathrm{n}}$ is the rotational matrix, derived from the quaternions properties.

$$\Omega_{\mathrm{nb}}^{\mathrm{n}} = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \quad (2)$$

This matrix is formed by the angular velocities ω_x, ω_y and ω_z measured by the gyro. Equation (1) is a time-continuous equation that can be easily transformed in the time-discrete equation to be used in our system, obtaining A_k matrix that relates the state evolution of the system using only the gyro data. To ensure data integrity, the quaternion must be normalized to unit norm. A correction equation that uses the data from the accelerometers has to be introduced. From the estimated angular position it is possible to estimate the gravity vector $\hat{g}$ using the direction cosine matrix R_n^b, assuming constant the g-force acceleration |g|:

$$\hat{g} = R_n^b \begin{bmatrix} 0 \\ 0 \\ |g| \end{bmatrix} = |g| \begin{bmatrix} 2q_1q_3 - 2q_0q_2 \\ 2q_0q_1 + 2q_2q_3 \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \quad (3)$$

This vector is compared with the measured gravity a, which obviously suffers of high errors, because the accelerometers do not measure the gravity only, but also the external accelerations of the system. For this reason, the correction factor must be weighted with the Kalman gain K_k, a coefficient that is calculated from the statistics of the noise covariance matrices of the system.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \quad (4)$$

P_k^- and P_k are the a priori and a posteriori error covariance matrices, R_k and Q_k are the error covariance matrices of the read and the state equation of the filter, assumed constant at each filter iteration, $H_k$and V_k are the Jacobian matrices of the partial derivatives with respect to the quaternion and to the noise of the nonlinear equation (3), which relates the quaternion to the estimated gravity.

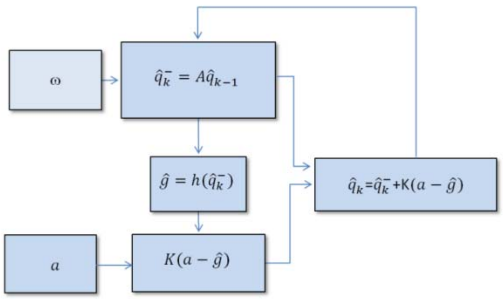
ω
$\hat{q}_k^- = A\hat{q}_{k-1}$
$\hat{g} = h(\hat{q}_k^-)$
$\hat{q}_k$=$\hat{q}_k^-$+K$(a - \hat{g})$
a
$K(a - \hat{g})$

Figure 1. Kalman filter algorithm.

Before calculating the gain K_k, the a priori error covariance P_k^-, a matrix that represents the error in the state estimation when only the state equation is used, needs to be calculated:

$$P_k^- = A_k P_{k-1} A_k^T + Q_{k-1}. \quad (5)$$

Finally, the a posteriori error covariance matrix P_k that is needed for the following step of the filter is obtained.

$$P_k = (I - K_k H_k) P_k^- \quad (6)$$

In this work, considering that the system may be subject to high external acceleration, the covariance matrix of the measurement noise R has a quite high estimation:

$$\mathrm{R} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}. \quad (7)$$

After some tests, the following value was found as the best fit for the process noise covariance matrix Q:

$$\mathrm{Q} = 10^{-6} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

and the following value was chosen to initialize the starting covariance matrix:

$$P0 = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}. \quad (9)$$

In order to have a better angular estimation, [8] uses a complementary filter on the gyro and accelerometers data. In this work, a dead zone on the gyro data is applied with a limit of 0.01 radians per second, while a low-pass filter on the accelerometer data will be evaluated in a future work.

III. Algorithm Simplifications

There is no problem to run this kind of filter on a PC, but the goal is to obtain an integrated system, with minimal area complexity and power consumption. For each filter iteration about 600 multiplications, 400 additions, a square root and 13 divisions are required, so some algorithm simplifications, with negligible performance degradation, were devised.

First, the a priori error covariance P_k^- is approximated with the a posteriori error covariance P_k. This is based on the assumption that the angular position estimation before the correction is approximately the same as after the angular position correction. This is always verified when the frequency of the filter iteration is greater w.r.t. the gyro drift.

Second, a pre-computed value of P_k is used. This is a tuning parameter of the filter that has been calculated offline, from an average of the values assumed by P_k in some runs of the filter. Then, the following value was calculated:

$$\mathrm{P} = \begin{bmatrix} 0.001 & 0.0002 & 0.0002 & 0.0002 \\ 0.0002 & 0.001 & 0.0002 & 0.0002 \\ 0.0002 & 0.0002 & 0.001 & 0.0002 \\ 0.0002 & 0.0002 & 0.0002 & 0.001 \end{bmatrix}. \qquad (10)$$

To allow a faster evaluation of the initial position, a higher value of P_k is used in the first 50 iterations of the filter.

$$\mathrm{P} = \begin{bmatrix} 0.5 & 0.0003 & 0.0003 & 0.0003 \\ 0.0003 & 0.5 & 0.0003 & 0.0003 \\ 0.0003 & 0.0003 & 0.5 & 0.0003 \\ 0.0003 & 0.0003 & 0.0003 & 0.5 \end{bmatrix} \qquad (11)$$

A higher value of P_k implies a higher value of K_k, so that the accelerometers are mainly used to calculate the initial position of the system.

Third, because the matrix P_k is pre-computed, the determinant used to invert a matrix in the K_k calculation was considered a constant, eliminating 9 division operations.

Fourth, it was verified that the normalization operation on the quaternion, that involves the calculus of a square root and 4 divisions, can be eliminated when the Kalman filter is working, because the norm of the quaternion is controlled by the filter itself.

Applying these simplifications, the final amount of operations is estimated as 400 multiplications and 300 additions.

IV. SIMULATIONS

The Simulink model was experimentally tested with SensorDynamics Cube Demo board, described in [9]. It is composed by three orthogonal SD755 sensors, each one integrating a gyro and an accelerometer. Thus, the Cube Demo board simulates a 6D integrated sensor, not yet available. The maximum declared gyro bias is ±0.5°/s at 25°C and ±1°/s within the entire temperature range.

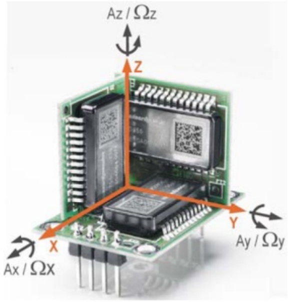

Figure 2. SensorDynamics Cube Demo board.

The test were carried out with a test equipment specially designed for gyro calibration, which allows to rotate the sensor on two axes simultaneously, with an accuracy of 0.01°; the data were imported in the Matlab workspace and processed with the Simulink model. For a sake of clarity, the following results are presented for the *x* axis angle, or roll angle, but it was verified that the obtained results are valid for both roll and pitch angle, whereas for the yaw angle there is no filter correction, because the gravity vector is parallel to this rotation axis so that it cannot be used to correct it.

A first test was carried out with the still sensor to verify the influence of the drift on the angular position estimation. Disabling the filter, the drift is quite high, about 30 degrees per minute, as shown in Figure 3.

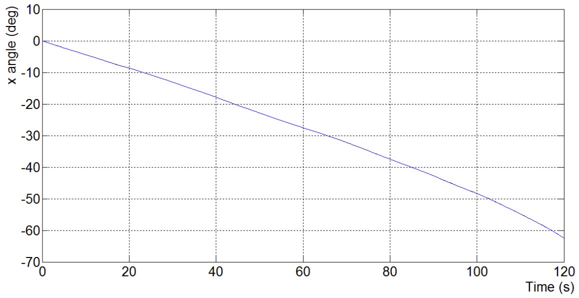

Figure 3. Drift on the *x* axis

Using the Kalman filter, the drift is eliminated. Figure 4 shows that, after the initial position correction caused by the high starting value of P_k, the angular position is stabilized with a maximum error of 0.7° during a simulation of 120 seconds.

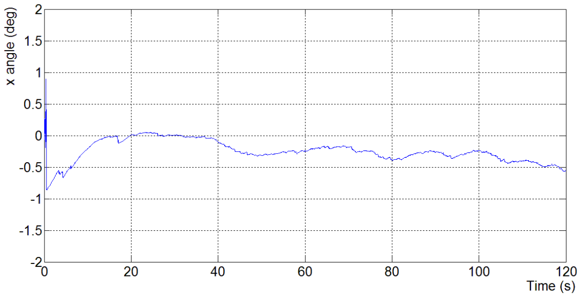

Figure 4. Drift correction on the *x* axis with the Kalman filter

We performed a test also to verify the ability of the system to correctly estimate the initial position of the sensor, even if it is upside down at the startup. The result is very encouraging, because the estimation of the angular position is corrected after 0.5 seconds only, as shown in Figure 5.

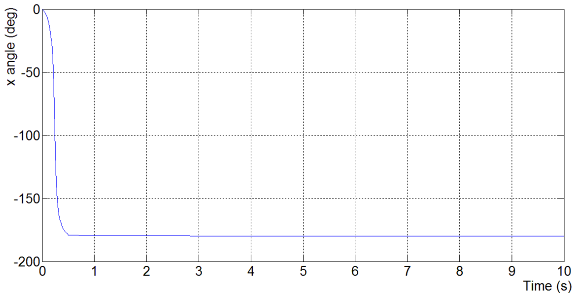

Figure 5. Initial angular position estimation on the *x* axis

Subsequently, the path in the second column of Table I was set in the test equipment and the mean square error of the angular position estimated by our model was calculated. This table shows that, when the filter is not active, the drift is quite high, but with the simplified Kalman filter the mean square error is very low.

TABLE I. ROTATION SEQUENCE

Time (s)	Angular positions for the x axis (deg)		
	Position expected (deg)	*Kalman filter off*	*Kalman filter on*
11.25	20.00	18.70	19.73
14.25	0.00	-1.70	-0.25
17.25	45.00	42.90	44.94
20.50	0.00	-2.55	-0.17
23.50	45.00	42.00	45.01
26.50	135.00	131.55	134.88
29.50	45.00	40.25	44.46
32.50	135.00	129.40	134.29
35.50	180.00	173.30	179.80
38.50	135.00	128.10	135.75
41.50	180.00	171.75	180.61
50.00	0.00	-9.40	-0.26
RMS		28.27	0.17

V. BIT TRUE MODEL

A quantized Simulink model was built and tested to simulate fixed point architecture. The rotation sequence of the previous test was utilized to calculate the mean square error, varying the number of bits used to represent each single parameter. Table II shows the results obtained by varying the number of bits for the quaternion q, the A matrix and the P and K matrices.

For the quaternion, the optimum number of bits to obtain the best precision without increasing the logic complexity is 20 bits. For the A matrix representation, even if there is a small increase of precision with the use of 24 bits, 20 bits are recommended to avoid an explosion of the system complexity. Finally, the use of 16 bits is adequate to represent P and K matrices. Using the above mentioned arithmetic precision, a final RMS value of 0.19 was obtained, with only a slight degradation w.r.t. the simplified floating point model, whose RMS is 0.17.

TABLE II. ROOT MEAN SQUARE ERROR RELATED TO THE NUMBER OF BITS USED FOR DATA REPRESENTATION

	Root Mean Square error				
Number of bits used for data	***10 bit***	***12 bit***	***16 bit***	***20 bit***	***24 bit***
quaternion	-	8.86	0.24	0.19	0.19
A matrix	2.36	0.75	0.29	0.19	0.18
P and K matrices	-	0.23	0.19	0.19	-

VI. CONCLUSIONS

In this work a Simulink model of a Kalman filter for an orientation tracking system with 6D IMU sensors was experimentally tested using the SensorDynamics Cube Demo board. We proposed some simplifications of the Kalman filter algorithm and we experimentally verified that the filter is able to correct the gyro sensor drift and also the initial position at system startup, even if the sensor is upside down. Finally, Simulink model was quantized and we verified the precision loss in fixed point architecture by varying the number of bits used to represent each data. These results are fundamental to finalize the digital design of an integrated system to provide orientation tracking capability in a future 6D IMU sensor.

REFERENCES

[1] Anthony Kim, M. F. Golnaraghi, “A Quaternion-Based Orientation Estimation Algorithm Uning an Inertial Measurement Unit”. Proceedings of International Conference on Intelligent Robots and Systems, IEEE press, May 2001.

[2] Xiaoping Yun, Mariano Lizarraga, E. R. Bachmann, Robert B. McGhee, “An Improved Quaternion-Based Kalman Filter for Real-Time Tracking of Rigid Body Orientation”. Proceedings of International Conference on Intelligent Robots and Systems, IEEE press, December 2003.

[3] J.L. Marins, Xiaoping Yun, E. R. Bachmann, R. B. McGhee, M.J. Zyda, “An extended Kalman Filter for Quaternion-Based Orientation Estimation Using MARG Sensors”. Proceedings of International Conference on Intelligent Robots and Systems, IEEE press, 2001.

[4] Mei Wang, Yunchun Yang, Ronald R. Hatch, Yanhua Zhang, “Adaptive Filter for a Miniature MEMS Based Attitude and Heading Reference System”. Proceedings of Position Location and Navigation Symposium, IEEE press, July 2004.

[5] Edgar Kraft, “A Quaternion-Based Unscented Kalman Filter for Orientation Tracking”. Proceedings of the Sixth International Conference of Information Fusion, IEEE press, 2003.

[6] Joseph J. LaViola Jr, “A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion”. Proceedings of American Control Conference, IEEE press, June 2003.

[7] Wassim Khoder, Bienvenu Fassinut-Mombot, Mohammed Benjelloun, “Inertial Navigation Attitude Velocity and Position Algorithms using Quaternion Scaled Unscented Kalman Filtering”. Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE.

[8] Hyung-Jik Lee1, Seul Jung, “Gyro Sensor Drift Compensation by Kalman Filter to Control a Mobile Inverted Pendulum Robot System”. IEEE International Conference on Industrial Technology, 2009. ICIT 2009.

[9] F. Sechi, L. Fanucci, E. Maci, A. Giambastiani, A. Rocchi, M. De Marinis “Demonstrator of 3 axis gyroscope/accelerometer using embedded 3D demo architecture”, 11th Euromicro Conference on Digital System Design, Parma, 2008

[10] Leandra Vicci, “Quaternions and Rotations in 3-Space: The Algebra and its Geometric Interpretation”. Microelectronic Systems Laboratory, Departement of Computer Science, University of North Carolina at Chapel Hill, 2001.

[11] Jack B. Kuipers, “Quaternions and Rotations Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality”. Princeton University Press, 1999.

[12] Rudolf Emil Kalman, “A New Approach to Linear Filtering and Prediction Problems”. Transactions of the ASME–Journal of Basic Engineering, 1960.

[13] Greg Welch and Gary Bishop, “An Introduction to the Kalman Filter”. Department of Computer Science, University of North Carolina at Chapel Hill, 2001.

[14] M. S. Grewal, A. P Andrews, “Kalman Filtering: Theory and Practice Using Matlab”, Second Edition. New York: John Wiley & Sons, 2001.