A double stage Kalman filter for sensor fusion and orientation tracking in 9D IMU

Simone Sabatelli[#], Marco Galgani[#], 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 based on a double stage Kalman filter for sensor fusion in 9D IMU. The IMU is composed by a 3D gyro, a 3D accelerometer and a magnetic compass. The filter was divided into two stages to reduce algorithm complexity. Gyro data are used to first estimate the angular position, then the first stage corrects roll and pitch angles using accelerometer data. The second stage processes magnetic compass data to correct the yaw angle. One of the advantages of this kind of filter is that a magnetic anomaly does not influence roll and pitch estimation accuracy. The flexibility is also desirable, because if the magnetic compass is not available, it is simply possible to switch off the second stage of the filter. In this work an ASIP was designed to process the filter algorithm and a proof of concept on FPGA was successfully realized. In the future the ASIP will be integrated within the logic of a new 6D sensor that could be optionally interfaced with an external magnetic compass.**

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

I. INTRODUCTION

MEMS sensors are used in many consumer applications, like smartphones and tablets. Nowadays high-end smartphones have a lot of different sensors, such as gyro, accelerometer and magnetic compass. Now integrated sensors, like a gyro plus accelerometer 6D sensor, are available on the market, but all the sensor fusion algorithms necessary to obtain additional information, such as an estimated angular position, are executed via software by the smartphone processor.

There are a lot of works about sensor fusion algorithms, but in most of them a quite complex algorithm is developed that must be executed by a separate microcontroller or is executed offline on a PC. Generally the angular position is represented with a quaternion, because it is a compact and singularity free way to represent angular position and can be quickly reconverted in a rotational matrix form or in Euler angles. Kalman filters with different formulations are used as sensor fusion algorithms.

For example, in [1] an Extended Kalman Filter (EKF) is used with a three rate gyro and three accelerometers. In the state equation the position quaternion and also the angular velocities and the gyro drift are included.

A complete set of sensors to form a 9D IMU is used for better angle estimation in [2]: a complete EKF is developed with accelerometer and magnetic compass data in the correction equation. Also gyro bias is estimated. In [3], instead of using an EKF, a QUaternion ESTimate (QUEST) algorithm is used to estimate the position quaternion from accelerometer and magnetic compass data, so that is possible to use a linear Kalman Filter (KF). The algorithm complexity is moved from the EKF to the QUEST algorithm. In a more recent work [4] an alternative to QUEST algorithm was proposed: it is called Factorized Quaternion Algorithm (FQA) and has some computational advantages.

In [5] an Unscented Kalman filter is used instead of the traditional Extended Kalman filter, because the Unscented filter is considered 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.

Discrete sensors are widely used in previous works. The purpose of this work is to design a very simplified and flexible algorithm to be integrated within the logic of a 6D sensor. The algorithm must be flexible to process data from the 6D unit only or also from a 9D unit.

In this work a new method to process data from 9D IMU is proposed: instead of using a complex Kalman filter with a read equation composed by 6 elements, the filter is composed by two separate stages (see Figure 1). The first stage processes accelerometer data to correct roll and pitch angles, while the second stage processes magnetic compass data to correct yaw angle.

With this method it is possible to separate the effects of the correction between the three angles, so that a magnetic anomaly does not influence precision in roll and pitch estimation, in a similar way to what happens in a FQA algorithm. The advantage of this kind of approach w.r.t. FQA is that in this case it is very simple to adapt the same algorithm to work with accelerometer correction only, if the magnetic compass isn't available, simply switching off the second stage of the filter. This is very desirable, because this algorithm is designed to work with a 6D IMU that could be optionally interfaced with an external magnetic compass: if the magnetic compass is available, the IMU will process the entire algorithm having an optimal angle estimation, otherwise it will process the first part of the algorithm only, losing the correct estimation of the heading angle.

The other big advantage of this two stage filter w.r.t. a unique EKF filter is that it is less computationally intensive to work with two systems having smaller matrices than having a unique system with bigger matrices.

978-1-4577-1725-3/12/$26.00 ©2012 IEEE

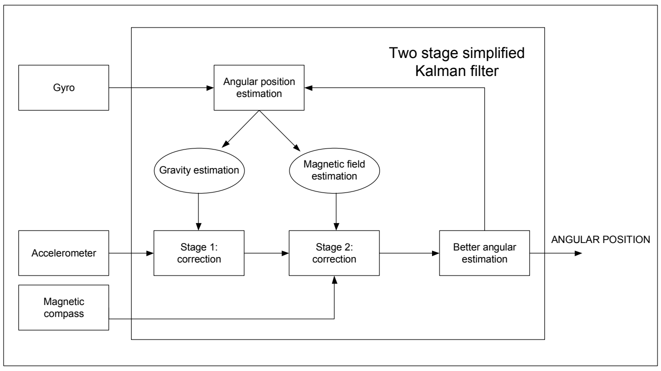

Figure 1. Two stage Kalman filter algorithm

This paper is so organized: in Section II the system model and algorithm is exposed and in Section III some algorithm simplifications are presented. The algorithm was first tested with synthetic data on MathWorks Simulink, then an Applications Specific Instruction Set Processor (ASIP) was specifically designed to process the algorithm. This is illustrated in section IV, while in Section V the experimental results are presented. Finally, conclusions are in Section VI.

II. SYSTEM MODELLING

The Kalman filter designed in this work uses the direct formulation approach. The state equation is composed by the quaternion only, because the addition of other variables does not significantly increase the precision of the angular estimation, but requires a lot more hardware requirements. The a priori estimation of the angular position is calculated using gyro data in the state update equation. Following quaternion algebra, the state equation is the following:

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

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.

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

This matrix is formed by the angular velocities ω_x, ω_y and ω_z measured by the gyro. Equation (1) can be easily transformed from the time-continuous form into the time-discrete one, obtaining the A_k matrix that relates the state evolution of the system.

To ensure that the output quaternion represents an angular position, this output must be normalized. It was discovered that, when the filter is active, it is self-normalizing.

As seen in Kalman filter theory, the system state, i.e. the angular position, is not directly observable, but there are some outputs of the system that are observable. Now there are two outputs, also called read equations: the accelerometer data and magnetic compass data. The first stage of the filter calculates an estimated gravity vector $\hat{g}$ using the direction cosine matrix R_n^b.

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

This gravity vector is compared with the accelerometer data and a correction quaternion $q_{\epsilon 1}$ is calculated multiplying this difference, called residual, with a gain factor called Kalman gain K_{k1} . The correction quaternion represents an angular rotation, it has non-null terms for all three angles, but from the theory it is known that with the gravity vector it is possible to correct two angles only. So the third vectorial part q_4 of the quaternion is set to zero to ensure that the yaw angle is not influenced from the first correction stage.

$$q_{\epsilon 1} = q_1 + q_2 + q_3 + 0 \cdot q_4 \tag{4}$$

The Kalman gain is calculated from the statistics of the noise covariance matrices of the system, and usually is a small quantity, because accelerometers do not measure the gravity only, but also the external accelerations of the system. These accelerations are considered as noise in the filter formulation.

The formula to calculate the Kalman gain in the original filter formulation is the following:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \tag{5}$$

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 Jacobean 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.

In a similar way, the second stage of the filter is developed. The expected magnetic vector is calculated from the estimated angular position, normalizing its value to 1.

$$\widehat{m} = R_n^b \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2q_1q_2 + 2q_0q_3 \\ q_0^2 - q_1^2 - q_2^2 - q_3^2 \\ 2q_2q_3 - 2q_0q_1 \end{bmatrix}. \tag{6}$$

For this calculation the earth magnetic vector was considered directed along y axis only and the vertical component was neglected. This because this component is dependent from the geographical position and is not necessary for north estimation. Also the magnetic field intensity is variable: it is between 0.25 to 0.65 gauss, weaker near the equator and stronger at the poles. For this reason the magnetic compass readings are normalized to 1.

A residual is calculated subtracting the estimated value to the read one and with a second computed Kalman gain K_{k2} a second correction factor is calculated. To uncouple the effects of this correction from roll and pitch correction, the first and second vector parts q_2 and q_3 of the quaternion are set to 0:

$$q_{\epsilon 2} = q_1 + 0 \cdot q_2 + 0 \cdot q_3 + q_4 \quad (7)$$

The two stages of the filter work sequentially and can be switched off separately.

III. ALGORITHM SIMPLIFICATIONS

Kalman filter algorithms may be very computationally intensive, especially if the system state has a lot of elements because it includes also the biases. For this reason in this work the system state equation was maintained as simple as possible to save computational power. Also the use of a two stage filter is in this direction, because it allows the use of two separated read equations, each with 3 elements only, instead of a unique read equation with 6 elements (the 3 accelerations plus the 3 magnetic elements). As it is known, it is simpler to execute matrix operations, such as matrix inversion, on two 3x3 matrices than on a bigger 6x6 matrix.

To allow system silicon integration and to run this algorithm on an ASIP with minimal hardware requirements, the current Kalman filter algorithm was further simplified. In particular, it was chosen to calculate the noise covariance matrices offline, so that the online calculation of the Kalman gain is simpler. The a priori error covariance P_k^- is approximated with the a posteriori error covariance P_k and a fixed precomputed value is used. It is relevant that the overall algorithm performance is not affected by this simplification and it has negligible performance degradation.

A variable gain was introduced to compensate startup performance, because the system must be able to quickly correct the initial angular position even if the system starts upside down. During the first filter iterations, accelerometer and magnetic compass data are weighted more in the correction equation, so that the real position is quickly hooked. It was seen that it is possible to correct the initial angular position in less than half a second. In the future, a variable gain might be introduced also to increase the performance and reducing errors when the system is subjected to high external accelerations or to magnetic anomalies, reducing the weights of the corrections.

An effort was done to eliminate all the division operations, to avoid the use of a divider into the ASIP. So, it was verified that the normalization operation on the quaternion can be eliminated when the Kalman filter is working, because the norm of the quaternion is controlled by the filter itself. Furthermore, with some matrix algebra approximation it was possible to reduce the complexity of the Kalman gain calculation and to avoid divisions.

IV. ASIP DESIGN

The system constraints to integrate the ASIP within sensor logic were the use of a TSMC 0.18 µm technology and area and power consumption requirements of 0.5 mm^2 and 100 µA respectively. It was chosen to implement an ASIP because this allowed a good balance between performance, flexibility for future algorithm upgrades and a reduced area occupation.

For the ASIP design a custom Instruction Set Architecture was designed to maximize the performance in Kalman filter processing and special attentions were paid to power and area occupation.

The ASIP has three main blocks, the ALU, the control unit and the regbank. The ALU has a RISC architecture because this allows low area occupation. It is possible to implement a RISC architecture because the operations to be executed are much lower than the available clock cycles: the filter has an update rate of 1 kHz, meanwhile the ASIP has a clock frequency of more than 1 MHz. The ALU is capable of operations on 20 bit data, because this is the minimum precision required for the Kalman filter algorithm.

The operations to be executed by the ALU are stored on a program ROM, so it is possible to upgrade the algorithm simply changing the ROM that stores the operations to be executed.

The control unit has a mixed approach between the hardwired and the programmed solution because it was found to be more efficient and also flexible. A regbank was instantiated to store the data and it was divided into two main registers blocks to optimize ALU access time to data.

The synthesis of the DSP on TSMC 0.18 µm technology showed an area occupation of 0.36 mm^2, that is equivalent to 28.8 kgate. The estimated current consumption is 75 µA in active mode and 1.5 µA for the leakage current.

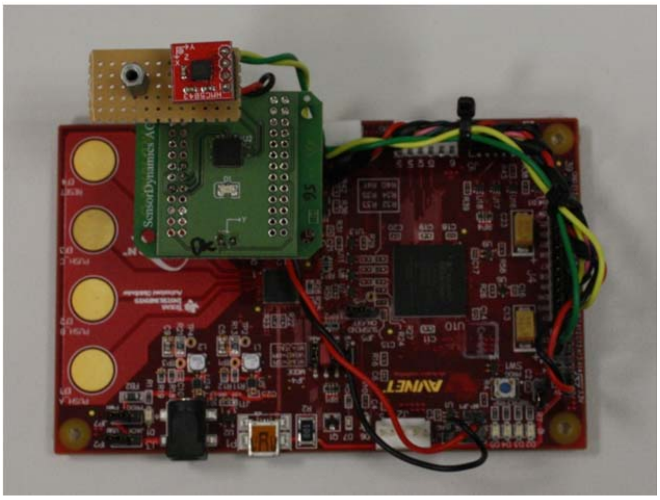

Figure 2. FPGA board with SensorDynamics SD746 gyro plus accelerometer and Honeywell HMC5843 Magnetic compass.

V. SIMULATIONS

A proof-of-concept demonstrator was implemented on FPGA (Figure 2) to verify the correct operation and to test its performance. The FPGA was interfaced via SPI with an existing prototype of 6D IMU sensor, SD746 by SensorDynamics. A magnetic compass HMC5843 from Honeywell was interfaced with I2C bus and the board was connected via UART to a PC for data recording and to perform a live demo. The output data update rate was set at 1 kHz, which is high enough to track also high bandwidth human body movements.

In the first test (Figure 3-5) it was verified the ability of the filter to correct the angular position if the system starts up-down. After a first transient, the angular position is correctly hooked for all the three axes in about half a second.

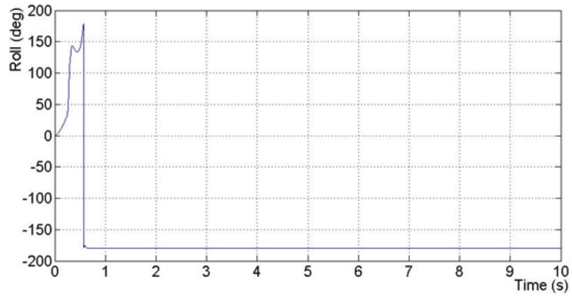

Figure 3. Up-down startup for roll angle

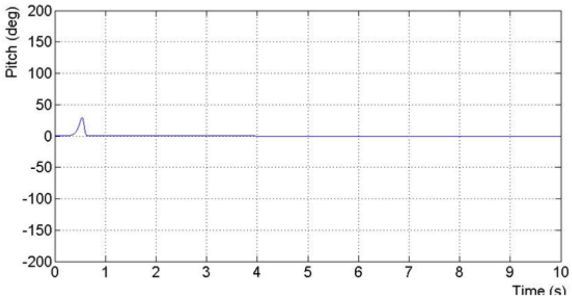

Figure 4. Up-down startup for pitch angle

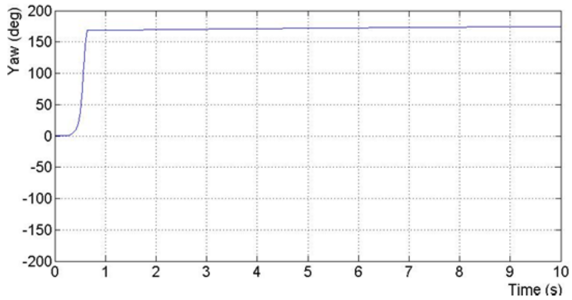

Figure 5. Up-down startup for yaw angle

After that, it was verified that the sensor is able to correct the angular position drift due to gyro biases (Figure (6-8). A long acquisition was done with the still sensor, and it was evidenced that for all the three axes the drift is nonexistent and the angular position error is less than 1 degree for roll and pitch angles, while it is about 3 degrees for yaw angle estimation.

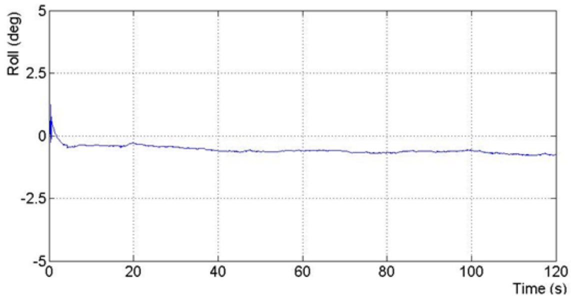

Figure 6. Roll angle position estimation with still sensor

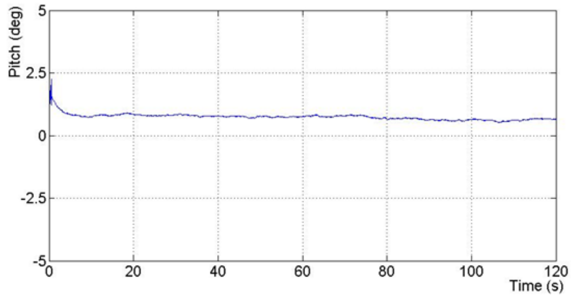

Figure 7. Pitch angle position estimation with still sensor

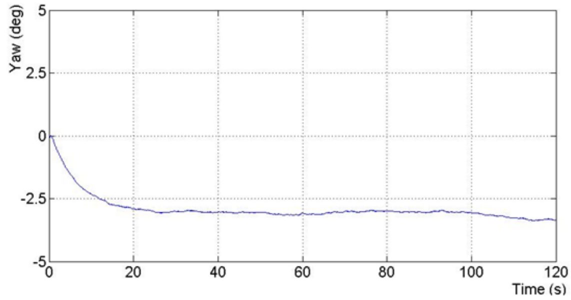

Figure 8. Yaw angle position estimation with still sensor

Finally, the sensor was placed on a test machine to perform a reproducible sequence of rotations. Due to the limits of the test machine, it was possible to rotate the sensor only around the roll angle. On Figure 9 the roll angle estimation during this test is shown: the error between the true and the estimated position is less than 1 degree.

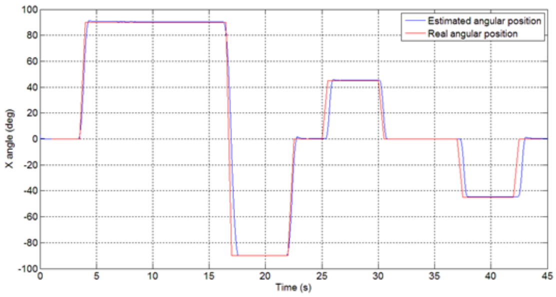

Figure 9. Roll angle real and estimated position

VI. CONCLUSIONS

In this work a novel angular position estimation system to work with 9D IMU was successfully designed and tested with an FPGA prototype. The algorithm is based on a simplified two stage Kalman filter to process separately the data from the accelerometer and from the magnetic compass. The advantages of this algorithm are: the fact that a magnetic anomaly will not reduce estimation precision on roll and pitch angles; reduced computational requirements; flexibility, because the algorithm can be used with a 6D IMU only or also with a magnetic compass.

An Application Specific Instruction Set Processor was specifically designed to process the algorithm and a synthesis with TSMC 0.18 µm technology showed an area occupation of 0.36 mm^2 that meets our requirements. The ASIP was synthesized on FPGA and interfaced with real sensors to completely verify his functionality.

In the future it will be possible to integrate the ASIP within the sensor logic of a 6D IMU sensor. Without a magnetic compass, roll and yaw angle estimation only will be possible. Optionally, it will be possible to use an I2C interface to connect an external magnetic compass and process the full algorithm, having also an optimal estimation of yaw angle.

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] R. Munguia, A. Grau, "Attitudeand Heading System based on EKF Total State Configuration", IEEE International Symposium on Industrial Electronics (ISIE), June 2011.

[3] Xiaoping Yun, E. R. Bachmann, "Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking", IEEE Transactions on Robotics, 2006.

[4] Xiaoping Yun, E. R. Bachmann, "A Simplified Quaternion-Based Algorithm for Orientation Estimation From Earth Gravity and Magnetic Field Measurements" IEEE Transactions on Instrumentation and Measurement, March 2008.

[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] S. Sabatelli, F. Sechi, A. Rocchi, L. Fanucci, "A sensor fusion algorithm for an integrated angular position estimation with inertial measurement units", Design, Automation & Test in Europe Conference (DATE), 2011

[8] S. Sabatelli, M. Galgani, A. Rocchi, L. Fanucci, "An application specific instruction set processor for angular positionestimation with inertial measurement units", IEEE Sensors Conference, October 2011.

[9] 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.

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

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

[12] Micheal Keating, Pierre Bricaud, *Reuse Methodoogy Manual for System On a Chip Design*, Third Edition, 2002.

[13] Richard R. Eckert, *Micro-programmed versus hardwired control units: how a computer really work*, Department of Computer Sciences, Binghmanton (NY) 1988

[14] N.S Matlo, *Introduction of a Microcode Implementation of a CPU Architecure*, 1997