Sensor fusion for IMUs : Real-time Quaternion-Based Orientation Estimation using Extended Kalman Filter

Ajin Tom
Music Technology Department
McGill University - Montreal, CA
ajin.tom@mail.mcgill.ca

ABSTRACT

This project addresses the orientation estimation problem for IMUs using sensor fusion with extended Kaman filter. The MPU6050 IMU, used for this project, consists of a three-axis magnetometer, a three-axis angular rate sensor (gyroscope), and a three-axis accelerometer. The filter represents rotations using quaternions rather than Euler angles, which eliminates the problem of singularities associated to orientation estimation across the 3 axes. The main challenge in using Kalman filter for this scenario was that it requires a linear model, which is why the Gauss-newton iteration algorithm was used to find the best quaternion that relates to the measurements from the accelerometer and magnetometer. These values served as measurement to the Kalman filter model while the estimates were derived from the angular velocity measurements from the gyroscope. The results were tested on Jitter objects in MaxMSP, MATLAB plots comparing estimated and observed quaternions as well as through a gesture based musical task using the IMU running the filter. Future work involves comparing extensively linearization techniques like gradient descent and Levenberg-Marquardt method for better quaternion measurements along with fine-tuning the algorithm bias estimates using motion capture techniques.

Keywords
sensor fusion, IMU, extended kalman filter, gauss-newton iteration algorithm, quaternions, real-time orientation

1. Introduction

Accurate real-time tracking of orientation or attitude serves several applications in robotics, automotive/aerospace industry, virtual reality as well as for gesture based musical tasks. Several works in the past [2][3] use various kinds of techniques involving computer vision, ultrasonics, active magnetic tracking, mechanical sensing, GPS + IMU based attitude tracking and several other sensor fusion algorithms. For my scenario I chose to explore a real-time sensor fusion algorithm that works with a standard widely used 6-DOF Inertial Measurement Unit, MPU-6050 compatible with micro-controllers like Teensy and Arduino through communication protocols like I2C, for low latency and highly accurate orientation estimation.

Some of the major challenges in orientation estimation are associated to singularities in measurements from IMUs after 360 degrees of rotation about an axis. The best approach for this problem involves using quaternions instead. Orientation can be defined as a set of parameters that relates the angular position of a frame to another reference frame. Quaternions are a 4-dimensional extension to complex numbers which has a real component and 3 imaginary components, each directed along x, y and z axis.

This project involves comparing various algorithms like Complimentary filter, AHRS algorithm, Kalman filter, etc. This report will include the concepts and implementations of the Kaman filter and other sensor fusion techniques for addressing the orientation estimation problem. One of the main challenges in using Kalman filter involves having a linear model for the algorithm to function. Therefore to linearize the orientation measurements, the gauss-newton iteration algorithm is used and then fed to the Kalman filter as measurement updates. The accelerometer and magnetometer measurements, being noisy, were fed through butter-worth low-pass filters. This linear model now uses measurements from the gyroscope for angular velocity estimation and prediction. The Kalman filter then uses both these measurements to provide accurate quaternion estimates from the observations and the model estimates.

In the next sections, basic overview of Kalman filter, implementation of quaternions and the updates steps over each iteration will be presented followed by evaluations, conclusion and future work.

2. Kalman Filter Overview

Kalman Filter is an efficient recursive and optimal filter that estimates an unobservable noisy state. Several works that involve Kaman filter implementations can be found in [4][5]. The orientation problem is addressed using quaternions [7].

Kalman filter uses a system's dynamics model (i.e., physical laws of motion), known control inputs to that system, and measurements (such as from sensors) to form an estimate of the system's varying quantities (its state) that is better than the estimate obtained by using any one measurement alone.
The model of the generic dynamic system is as follows:

$$\begin{cases} \vec{x}_t = F \cdot \vec{x}_{t-1} + B \cdot \vec{u}_{t-1} + \vec{w}_{t-1} \\ \vec{z}_t = H \cdot \vec{x}_t + \vec{v}_t \end{cases}$$

where the noise terms *w* and *v* are the state/process noise and state observation noise respectively; both modelled as Gaussians.

The Kalman filter is based on 2 steps:

- **Prediction step:** it predicts the state on the basis of the previous realization using the F matrix which represent the state evolution matrix.
- **Update step:** the state is updated using the observation z_t.

3. Quaternion Kalman Filter - Implementation

The filter is implemented with quaternion components such that q1 is the real part and q2, q3 and q4 are the imaginary parts. This is used to represent the state of the system, x:

$$\vec{x} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

3.1. Prediction step

Just like in [1], the equation that links the angular velocity to quaternion variation is as follows:

$$\dot{q} = \frac{1}{2} \cdot \vec{x} \otimes \vec{\omega}$$

In the sensor values acquisition loop, a time ticking mechanism is implemented to keep track of sample period over which the integration is carried out. The following equation is used to predict value of the state:

$$\vec{x}_t = \vec{x}_{t-1} + \dot{q}_t \cdot \delta t = F \cdot \vec{x}_{t-1}$$

The matrix F for the Kaman filter, the state transition model is as follows:

$$F = \begin{bmatrix} 1 & -\frac{1}{2} \cdot \omega_x \cdot \delta t & -\frac{1}{2} \cdot \omega_y \cdot \delta t & -\frac{1}{2} \cdot \omega_z \cdot \delta t \\ \frac{1}{2} \cdot \omega_x \cdot \delta t & 1 & \frac{1}{2} \cdot \omega_z \cdot \delta t & -\frac{1}{2} \cdot \omega_y \cdot \delta t \\ \frac{1}{2} \cdot \omega_y \cdot \delta t & -\frac{1}{2} \cdot \omega_z \cdot \delta t & 1 & \frac{1}{2} \cdot \omega_x \cdot \delta t \\ \frac{1}{2} \cdot \omega_z \cdot \delta t & \frac{1}{2} \cdot \omega_y \cdot \delta t & -\frac{1}{2} \cdot \omega_x \cdot \delta t & 1 \end{bmatrix}$$

To model the process noise, *w* the variance of the gyroscope on each axis is estimated for the Q matrix computation:

$$Q = E\left[\begin{bmatrix} -\omega_x - \omega_y - \omega_z \\ \omega_x - \omega_y + \omega_z \\ \omega_x + \omega_y - \omega_z \\ -\omega_x + \omega_y + \omega_z \end{bmatrix} \cdot \begin{bmatrix} -\omega_x - \omega_y - \omega_z & \omega_x - \omega_y + \omega_z & \omega_x + \omega_y - \omega_z & -\omega_x + \omega_y + \omega_z \end{bmatrix}\right]$$

Assuming zero mean gaussian model for *w* and gyroscope bias compensation (carried out at Teensy itself), the covariance Q matrix is computed as follows:

$$Q = \begin{bmatrix} \sigma_x^2 + \sigma_y^2 + \sigma_z^2 & -\sigma_x^2 + \sigma_y^2 - \sigma_z^2 & -\sigma_x^2 - \sigma_y^2 + \sigma_z^2 & \sigma_x^2 - \sigma_y^2 - \sigma_z^2 \\ -\sigma_x^2 + \sigma_y^2 - \sigma_z^2 & \sigma_x^2 + \sigma_y^2 + \sigma_z^2 & \sigma_x^2 - \sigma_y^2 - \sigma_z^2 & -\sigma_x^2 - \sigma_y^2 + \sigma_z^2 \\ -\sigma_x^2 - \sigma_y^2 + \sigma_z^2 & \sigma_x^2 - \sigma_y^2 - \sigma_z^2 & \sigma_x^2 + \sigma_y^2 + \sigma_z^2 & -\sigma_x^2 + \sigma_y^2 - \sigma_z^2 \\ \sigma_x^2 - \sigma_y^2 - \sigma_z^2 & -\sigma_x^2 - \sigma_y^2 + \sigma_z^2 & -\sigma_x^2 + \sigma_y^2 - \sigma_z^2 & \sigma_x^2 + \sigma_y^2 + \sigma_z^2 \end{bmatrix}$$

The prediction equation can be defined as follows using the prediction vector and output of the filter at previous step:

$$\vec{x}_{t|t-1} = F \cdot \vec{x}_{t-1|t-1}$$

3.2. Observations

The observations consist of four components of a quaternion, as the system's state, which are achieved from accelerometer and magnetometer signals. To compute the orientation quaternion using these two signals, an optimization problem has to be solved in order to have the quaternion which minimized the error, as shown in [6]. Such minimization tasks are carried out using non-linear estimation methods using Gradient descent or Gauss-Newton methods. This linearizes the measurements and thus makes it possible to use for Kalman filter model. Before the sensor acquisition steps begin, a short loop is carried out for calibration involving magnetic compensation to reduce errors due to electromagnetic interferences and filter parameters are also set to low pass both signals using butter-worth function on Matlab.

H matrix is the observation model that maps the true state space into the observation space. Since in this case, states spaces corresponds to observations space, H is identity.

3.3. Update step

In this step, the filter updates the predicted status using the computed observations. This update is performed by mean of a weighted sum where K is the Kalman filter gain is calculated as follows, $P_{t|t-1}$ is the prediction error :

$$\vec{x}_{t|t} = \vec{x}_{t|t-1} + K \cdot \left(\vec{x}_t - \vec{x}_{t|t-1}\right)$$

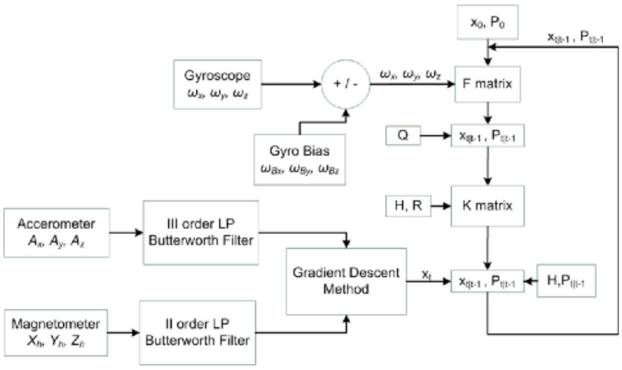
$$K = P_{t|t-1} \cdot H \cdot \left(H \cdot P_{t|t-1} \cdot H^T + R\right)^{-1}$$

$$P_{t|t-1} = \left(F \cdot Q \cdot F^T\right) + Q$$

3.4. Initialization step

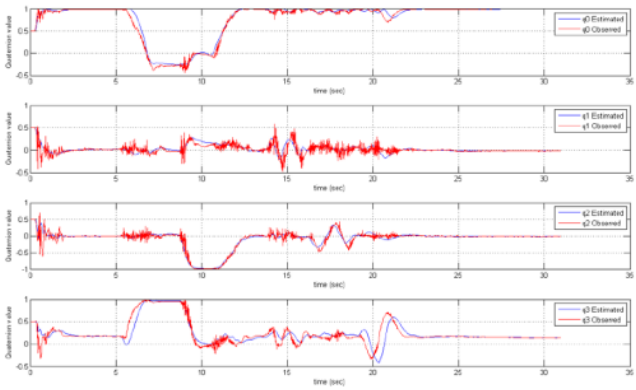
The initialization step is crucial here for the filter to perform accurately. Kalman filter requires an initial value of the state update, variance update and covariance matrix. The initial quaternion matrix is initialized as [0.5 0.5 0.5 0.5]. *To be explored:* some initial values can be after continuous tuning or using motion capture.

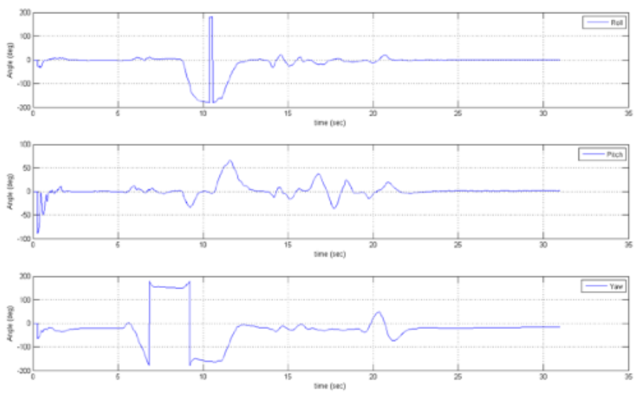
3.5. Block diagram



4. Results

Following are the plots obtained from an orientation task:





5. Conclusion and Future Work

This project addressed the real-time orientation problem using a recursive and optimal sensor fusion algorithm using the Kalman filter and pre-processing steps involving linearizing the model by using the Gauss-Newton iteration algorithm. The implementation was carried out using a 6-DOF MPU6050 IMU sending raw Accelerometer, Gyroscope and Magnetometer values via I2C and converted to radians in the Teensy and send serially to MATLAB. In MATLAB except for the butter-worth low pass filter, all other implementation involve self-written functions. The processed quaternion output is broadcasted via OSC to a Max patch where the observations of the filter can be made.

Along with this, a gesture based musical task was carried out by using the IMU, mounted on a keyboardist's finger for specialized mapped task such as pitch bend and other expressive modulations. The mapping which involved the Kalman filtered Pitch, Yaw and Roll proved to be much more comfortable to use for these tasks as compared to raw values mappings. [1] was the main reference for this implementation.

Future works involves porting the MATLAB script to a C++ implementation so that the whole algorithm works in a self-contained device like a Teensy. On the algorithm side, evaluation and comparison of other non-linear estimation methods like Gradient Descent and Levenberg-Marquardt method is to be carried out. Also, a motion capture setup could help tune the parameters for better calibration and faster convergence of the iterative algorithms. This implementation along with other implementations which uses complimentary filter and AHRS could be compiled and published as a C++ library.

6. ACKOWLEDGEMENT

I would like to thank Dr. Marcelo Wanderley for providing me with the opportunity to work on this exciting project. I would like to express my gratitude to Darryl Cameron for helping with the procurement of components required for the completion of this project.

7. REFERENCES

1. Marins, Joao Luis et al. “An extended Kalman filter for quaternion-based orientation estimation using MARG sensors.” *IROS* (2001).

2. Colton, Shane. "The balance filter: a simple solution for integrating accelerometer and gyroscope measurements for a balancing platform." *white paper, Massachusetts Institute of Technology, Cambridge, MA* (2007).

3. D. Gebre-Egziabher, G. H. Elkaim, J. D. Powell and B. W. Parkinson, "A gyro-free quaternion-based attitude determination system suitable for implementation using low cost sensors," *IEEE 2000. Position Location and Navigation Symposium*, San Diego, CA, 2000.

4. S. O. H. Madgwick, A. J. L. Harrison and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," *2011 IEEE International Conference on Rehabilitation Robotics*, Zurich, 2011.

5. D. Gebre-Egziabher, R. C. Hayward and J. D. Powell, "Design of multi-sensor attitude determination systems," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, April 2004.

6. Comotti, Daniele, "Orientation estimation based on gauss-newton method and implementation of a quaternion complementary filter.", 2011.

7. Kremer, Verena Elisabeth. "Quaternions and SLERP.", 2008.

8. Link to GitHub repo: https://github.com/ajintom/imu_kalman.git