Solutions for MEMS sensor fusion

Executive Overview

MEMS sensors include accelerometers to measure linear acceleration and earth gravity vectors, gyroscopes to measure angular velocity, magnetometers to measure earth's magnetic fields for heading determination and pressure sensors to measure the air pressure for altitude determinations. The number of applications becomes very large by combining these components in a sensor fusion solution. This paper explains how to make these sensors work together in a sensor fusion solution by describing some examples using complementary filters; The Kalman filter and the extended Kalman filter (EKF).

Jay Esfandyari, Roberto De Nuccio, Gang Xu, STMicroelectronics, Coppell, TX USA

For applications such as freefall detection, screen rotation, pedometer, tilt measurement and motion detection, etc., an accelerometer itself can perform the task. However, for other advanced applications including location-based services, enhanced motion gaming, pedestrian dead reckoning navigation, robot balancing, air mouse, human body tracking, unmanned aerial vehicles, etc., the fusion of MEMS sensors is necessary to achieve better performance at the system level in terms of accuracy, resolution, stability and response time. Sensor fusion is a set of adaptive algorithms for prediction and filtering. It takes advantage of different and complementary information coming from various sensors, combining it together in a smart way to optimize the performance of the system and enable new amazing applications.

Why sensor fusion?

When designing a system using multiple MEMS sensors, it is important to understand the advantages and disadvantages of accelerometers, gyroscopes, magnetometers, and pressure sensors.

Sensor fusion solves key motion sensing performance issues of 6-axis modules consisting of a 3-axis accelerometer and a 3-axis gyroscope or a 3-axis accelerometer and a 3-axis magnetic sensor. 1) A 6-axis inertial module with an accelerometer and a gyroscope loses its absolute orientation as the gyro drifts over time, requiring calibration to restore accurate heading reference. 2) A 6-axis module with accelerometer and magnetometer is prone to data corruption in the presence of ferrous materials in the environment. 3) A 9-axis module with an accelerometer, a gyroscope and a magnetometer eliminates the drift that occurs with stand-alone sensor solutions. But these can be subject to magnetic interference. Algorithms to fuse the sensor data are required to compensate for the magnetic interference.

The purpose of sensor fusion is to take each sensor measurement data as input and then apply digital
filtering algorithms to compensate each other and output accurate and responsive dynamic attitude (pitch/roll/yaw) results.

**Complementary filter**

Some applications such as robot balancing, camera image stabilization and 3D pointer have an accelerometer and a gyroscope installed in the system. The accelerometer can give accurate tilt angle measurements when the system is at rest. When the system is rotating or in motion, the accelerometer cannot follow the fast motion. The gyroscope can output dynamic angular velocity data. After single integration of this data over time, the angular displacement or tilt angle can be calculated, but this angle will become inaccurate in the long-term due to the gyroscope bias drift.

The complementary filter is a simple way to fuse the accelerometer and the gyroscope and the optional magnetometer to obtain accurate and responsive pitch/roll/yaw attitude outputs. It consists of a common low-pass filter for the accelerometer and a high-pass filter for the gyroscope, which is easier to understand and implement versus a Kalman filter. The following example explains how to implement the complementary filter for robot self-balancing.

As shown in Fig. 1, the robot has one dual-axis or tri-axis accelerometer to measure the static tilt angle, $\theta_a$, and one single-axis or dual-axis gyroscope to measure dynamic tilt angle, $\theta_g$. The complementary filter then fuses these measurements to obtain the final tilt angle $\theta$. The microprocessor uses this information and the angular velocity information from the gyroscope to control the motor in order to balance the robot.

**Figure 1.** Self-balancing robot.
The complementary filter block diagram is shown in Fig. 2 with a dual-axis accelerometer and a single-axis gyroscope configuration.

\[ \theta_a = \arctan \left( \frac{a_y}{a_z} \right) \]  

(1)

Where, \( a_Y \) and \( a_Z \) are normalized acceleration values in the range of \([-1g +1g]\) after applying zero-g offset and scale factor calibration parameters to the accelerometer raw data, \( \theta_a \) is the tilt angle in degrees.

\[ \theta_g = \theta_g + \left( \omega_X - \omega_{X0} \right) \cdot S \Delta T \]  

(2)

Where, \( \omega_X \) is the gyroscope angular velocity raw data and \( \omega_{X0} \) is the zero-rate level in the unit of \(^\circ/\text{s}/\text{LSB}\), \( S \) is the sensitivity in the unit of \(^\circ/\text{s}/\text{LSB}\), \( \Delta T \) is the sampling time interval, \( \theta_g \) is the angular displacement in degrees.

Now the final tilt angle from the complementary filter will look like the following,

\[ \theta = \theta'_g + \theta'_a = \beta \cdot \theta_g + \left( 1 - \beta \right) \cdot \theta_a \]  

(3)

Where, \( \beta \) is a constant between 0 and 1.

Let \( \beta \) be 0.95, then Equation (3) becomes,
The first part of Eq. 4 works like a high-pass filter (HPF) that allows the robot's dynamic motion to pass through the gyroscope. The second part works like a low-pass filter (LPF) that allows the robot's static or quasi-static motion to pass through the accelerometer.

If the accelerometer and gyroscope data are sampled at 100Hz, then the time interval $\Delta T$ is 0.01 second. So the time constant of the complementary filter is

$$\tau = \frac{\beta \cdot \Delta T}{1 - \beta} = \frac{0.95 \cdot 0.01}{1 - 0.95} = 0.19\text{sec}$$

The complementary filter can be considered as a simple filter by weighting the accelerometer data and gyroscope data. When the motion is faster than the 0.19 second time period, the gyroscope integration for angular displacement calculation $\theta_g$ is weighted more and the accelerometer noise is filtered out. When the motion is slower than the 0.19 second time period, the accelerometer tilt measurement $\theta_a$ has more weight than the $\theta_g$ of gyroscope to reduce the gyroscope bias drift impact from the vertical point.

Therefore, the tilt angle estimate from the complementary filter is accurate and responsive. It is not sensitive to the linear horizontal acceleration and gyroscope drift. This filter is easy to implement in a microcontroller compared to a Kalman filter. It can also be expanded to fuse multiple axes accelerometer and gyroscope data.

When the zero-rate level or bias $\omega_{x0}$ of gyroscope is constant and the robot is stationary, the tilt angle from the complementary filter output will also have a constant offset that can be compensated from the accelerometer tilt measurement [1]. If the bias is drifting over time and temperature, then the error of the tilt angle from the complementary filter will grow over time. In this case, $\omega_{x0}$ needs to be obtained when the robot is powered on and is stationary to cancel out gyroscope turn-on to turn-on bias instability. In addition, when the robot is stationary during the operation, new $\omega_{x0}$ can be obtained — again, periodically to cancel out bias in-run stability and short-term angular random walk [2].

**Kalman filter**

The Kalman filter is a mathematical method invented by Dr. Rudolf E. Kalman. Since its introduction in 1960, the Kalman filter has been implemented in many applications. The most well-known application is the GPS receiver itself and later, the integration of GPS with the inertial navigation system (INS). This recursive digital algorithm is used to integrate or fuse GPS measurement with accelerometer and gyroscope data to achieve optimal overall system performance.

The Kalman filter algorithm produces estimates of the true values of sensor measurements and their associated calculated values by predicting a value, estimating the uncertainty of the predicted value, and computing a weighted average of the predicted value and the measured value. The most weight is given to the value with the least uncertainty. The estimates produced by the algorithm tend to be closer to the true values than the original measurements because the weighted average has a better estimated uncertainty than either of the values that went into the weighted average.
Compared to the complementary filter, the Kalman filter requires a sound mathematical background including random signal processing, matrix theory and control theory. This section of the paper gives the basic discrete Kalman filter algorithm and the implementation for robot balancing as an example.

The Kalman filter tries to estimate the state $x$ of a discrete-time controlled process that is governed by the linear difference equation [3],

$$
\begin{align*}
    x_k &= Ax_{k-1} + Bu_{k-1} + w_{k-1} \\
    z_k &= Hx_k + v_k
\end{align*}
$$

(6)

Where

- $A$ is an $n$ by $n$ matrix that relates the state at the previous time step $k - 1$ to the state at the current step $k$, in the absence of either a driving function or process noise.
- $B$ is an $n$ by $l$ matrix that relates the optional control input $u$ to the state $x$.
- $H$ is an $n$ by $m$ matrix that relates the state to the measurement $z_k$.
- $w_k$ is the process noise (random variables).
- $v_k$ is the measurement noise (random variables).

$w$ and $v$ are assumed to be independent of each other, white, and with normal probability distributions so that,

$$
\begin{align*}
    p(w) &\sim N(0, Q) \\
    p(v) &\sim N(0, R)
\end{align*}
$$

where $Q$ is the process noise covariance matrix and $R$ is the measurement noise covariance matrix.

In **Fig. 3**, the Kalman filter algorithm is based on a linear dynamic system, which means that the matrices $A$, $B$ and $H$ are constant. To keep it simple, process and measurement covariance $Q$ and $R$ are also assumed as constant. Then the calculated Kalman gain $K_k$, each time step will weigh the measurement continuously to maintain the minimum error covariance matrix $P_k$ so that the updated states will be accurate and responsive.

Let's go back to the robot balancing example and try to implement the Kalman filter to fuse the accelerometer data and the gyroscope data for this application.

The first step is to construct the states and the measurement Eq. 6. Obviously, the final tilt angle $\theta$ is the state that we are going to estimate in order to compensate gyroscope bias $\omega_{X0}$ dynamically. This is the other state we are interested in. In addition, we want to use accelerometer measurements to limit the final tilt angle so that it will not drift over time due to the gyroscope random drift.

Let $b = \omega_{X0}$ and assuming that the gyroscope bias is constant coupled with white noise. Let
gyroscope measurement $\omega_X$ as the input $u_k$. Then Eq. 2 can be rewritten as,

$$\theta_{k+1} = \theta_k + \omega_X \cdot \Delta T - b \cdot \Delta T$$

where $\Delta T$ is the sampling time interval. For example, when the gyroscope data is sampled at 100Hz, then $\Delta T = 0.01$ second.

According to Eq. 1, the measurement can be written as,

$$z_k = \theta_k = \theta_a = \arctan \left( \frac{a_y}{a_z} \right)$$

Combining Equation (7) and (8), then Equation (6) can be constructed as,

$$x_{k+1} = \begin{bmatrix} \theta_{k+1} \\ b_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & -\Delta T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \theta_k \\ b_k \end{bmatrix} + \begin{bmatrix} \Delta T \\ 0 \end{bmatrix} \cdot u_k$$

$$z_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta_k \\ b_k \end{bmatrix}$$

From Eq. 9, we can see matrices $A$, $B$ and $H$ are constant so that the system is linear.

The second step is to obtain the process covariance matrix $Q$ from the offline experiments on the gyroscope and measurement covariance $R$ from the offline experiments on the accelerometer.

$$R = E(z_k z_k^T) = \sigma(\theta)$$

$$Q = E(x_k x_k^T) = \begin{bmatrix} E(\theta_k \theta_k^T) & 0 \\ 0 & E(b_k b_k^T) \end{bmatrix} \approx \begin{bmatrix} \sigma(\omega_X) \cdot \Delta T^2 & 0 \\ 0 & \sigma(b) \end{bmatrix}$$

The third step is to implement the Kalman filter algorithm shown in Fig. 3 in a microprocessor with the initial values of $x_0$ and $P_0$. 
The final step is to test the performance of the Kalman filter. Because the gyroscope bias is optimally estimated in the loop, the robot tilt angle from the Kalman filter will always be accurate and will not drift away once the Kalman filter parameters $Q$ and $R$ are fine tuned.

Extended Kalman filter

Some of the most interesting and successful applications of Kalman filtering have been situations where the process is estimated and/or the measurement relationship to the process is non-linear. A Kalman filter that linearizes the current mean and covariance is referred to as an extended Kalman filter (EKF). In something akin to a Taylor series, we can linearize the estimation around the current estimate using the partial derivatives of the process and measurement functions to compute estimates even in the face of non-linear relationships [3].

This paper will not cover the EKF algorithm. Instead, this section of the paper will illustrate a quaternion-based EKF design for fusing accelerometer, gyroscope and magnetometer data, assuming that readers have the mathematics background of rotation matrix between body frame and local horizontal reference frame, quaternion and Euler angles.

When a 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer are installed in a strap-down system as a 9-axis module, the quaternion method [4] for rotation matrix calculation based on gyroscope measurements is generally used because it doesn't have the singularity issue from the Euler angles. EKF is implemented to fuse MEMS sensor data and then to update the quaternion and gyroscope bias to output accurate attitude information for pitch, roll and yaw. This is the so-called attitude heading reference system (AHRS).
STMicroelectronics has a 9-axis MEMS sensor demonstration board available, dubbed the iNemo (iNertial MOdule V2) [5]. The quaternion-based EKF has been implemented in an STM32 microcontroller for AHRS, as shown in Fig. 4. When the board is rotating in 3D space, the fish cube will follow the motion and output real-time accurate dynamic pitch/roll/yaw values together with the quaternion values. The following presents the brief EKF design procedures in the iNemo V2 board. Detailed EKM algorithms can be found in [4].

To design the EKF, the following steps are needed:

1. Define the states and measurements using quaternion method;
2. Construct the system dynamic model and the relationship between the measurements and the states;
3. Linearize and discretize the system model and then calculate the Jacobian matrices for the matrix $A$, $B$ and $H$ at each time step $k$ [4];
4. Get the normalized quaternion values; and
5. Calculate the Euler angles pitch/roll/yaw.

**Conclusion**

When multiple MEMS sensors are included in a system, it is necessary to implement sensor fusion algorithms to take advantage of individual sensor capabilities for better overall performance. For Kalman filter and EKF, different system models with different sensor bias models can be designed while the basic recursive algorithms remain the same. Kalman filter and EKF can be considered as core to the sensor fusion scheme.

From the performance point of view, EKF is the best solution. However, it requires the most computation load, which consumes a lot of power – an example being smart phones. The trade-off between performance and power consumption is always a deciding factor with respect to which sensor fusion algorithm to use.

**References**

2. STMicroelectronics, Inc., White paper: "Introduction to MEMS gyroscopes," Nov. 2010


Biographies

Jay Esfandyari received his Master's and PhD in EE from the U. of Technology, Vienna, and is MEMS product marketing manager at STMicroelectronics, 750 Canyon Dr., Coppell, TX 75019 USA; ph.: 972-971-4969; email jalinous.esfandyari@st.com.

Roberto De Nuccio received his Master's in telecommunication engineering from the U. of Pisa, Italy, and is a business development manager at STMicroelectronics.

Gang Xu received his PhD from Shanghai Jiao Tong U., and is a senior application engineer at STMicroelectronics.

More Solid State Technology Current Issue Articles
More Solid State Technology Archives Issue Articles

To access this Article, go to: