Inertial Measurement Unit (IMU)
Algorithm Description Document (ADD)

by
Simeon Symeonidis

11/13/2018
revision 2.0

Overview

The purpose of the Inertial Measurement Unit (IMU) is to convert Microelectromechanical Systems (MEMS) accelerometer, magnetometer, and gyroscope sensor data into orientation and translational acceleration measurements. This Algorithm Description Document (ADD) captures the IMU contained within the displayIMU repository. This specific algorithm was tailored to minimize processing use via op-count optimization and increase usability via reducing algorithm parameters into the smallest, most intuitive subset possible.

The work captured hereinafter is an extension of the paper "An efficient orientation Filter for inertial and inertial/magnetic sensor arrays" written by Sebastian O.H. Madgwick. The algorithm was adapted to meet the needs of a wearable sensor operating in the presence of Electromagnetic noise. Algorithm changes include supporting asynchronous data and reworking the magnetometer processing function to minimize impact of "non-azimuth" component.

The report will be organized in two sections. The first section defines variable types and operations to be used. This is important because it will establish the conventions to be used in the paper. The second section defines the algorithm core. Not included in the document are the functions used to initialize system state, calculate figure of merits, apply weights, extract up and forward vectors, and conversions between Euler angles and rotation matrices (see Matlab files for these references).

Definitions

Sensor inputs will be represented as three dimensional vectors, denoted lowercase letter with an arrow on top. Common vector operations, normalize, dot products and cross products, are captured below.

$$\vec{v}=\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \qquad |\vec{x}|=\sqrt{x_1{}^2+x_2{}^2+x_3{}^2} \qquad \hat{x}=\begin{bmatrix} x_1/|\vec{x}| \\ x_2/|\vec{x}| \\ x_3/|\vec{x}| \end{bmatrix} \qquad \vec{x}\cdot\vec{y}=x_1\cdot y_1+x_2\cdot y_2+x_3\cdot y_3 \qquad \vec{x}\times\vec{y}=\begin{bmatrix} x_2 y_3-x_3 y_2 \\ x_3 y_1-x_1 y_3 \\ x_1 y_2-x_2 y_1 \end{bmatrix}$$

Projection and rejection operations can determine vector components parallel or perpendicular to each other. Their notation/equations are below. Bars denote a scalar value representing the magnitude of the projection.

$$proj_{\vec{x}}\vec{y}=\frac{\vec{x}\cdot\vec{y}}{\vec{y}\cdot\vec{y}}\,y \qquad |proj_{\vec{x}}\vec{y}|=\frac{\vec{x}\cdot\vec{y}}{|\vec{y}|} \qquad reject_{\vec{x}}\vec{y}=x-\frac{\vec{x}\cdot\vec{y}}{\vec{y}\cdot\vec{y}}\,y$$

Unit quaternions were selected to represent the internal state of the IMU. Unlike Euler Angles, which suffer from gimbal lock, they are robust. Also, they are mathematically more efficient and numerically stable than orientation matrices.

$$q=\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad \text{where} \quad q_1{}^2+q_2{}^2+q_3{}^2+q_4{}^2=1$$

The best way to understand orientation is through their axis-angle representation, where a unit vector, v, "indicates the direction of an axis of rotation" and an angle, theta, "describes the magnitude of the rotation around the axis." http://en.wikipedia.org/wiki/Axis-angle_representation

$$q=\begin{bmatrix}\cos(\theta/2)\\ v_1\sin(\theta/2)\\ v_2\sin(\theta/2)\\ v_3\sin(\theta/2)\end{bmatrix}$$

The conjugate and multiplication operations are defined below.

$$x^{-1}=[x_{1,}-x_{2,}-x_{3,}-x_4] \qquad x*y=\begin{bmatrix}x_1y_1-x_2y_2-x_3y_3-x_4y_4\\ x_1y_2+x_2y_1+x_3y_4-x_4y_3\\ x_1y_3-x_2y_4+x_3y_1+x_4y_2\\ x_1y_4+x_2y_3-x_3y_2+x_4y_1\end{bmatrix}$$

Algorithm

The gyroscope measures rotational rates across three all axes. To update the orientation quaternion, these rates have to be integrated. The derivative of a quaternion, where omega is the rotational rates for each axis, is defined below.

$$\omega=\begin{bmatrix}0\\ \omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \qquad q'=\frac{1}{2}q*\omega \qquad q'=\begin{bmatrix}-\frac{1}{2}q_2\omega_x-\frac{1}{2}q_3\omega_y-\frac{1}{2}q_4\omega_z\\ \frac{1}{2}q_1\omega_x+\frac{1}{2}q_3\omega_z-\frac{1}{2}q_4\omega_y\\ \frac{1}{2}q_1\omega_y-\frac{1}{2}q_2\omega_z-\frac{1}{2}q_4\omega_x\\ \frac{1}{2}q_1\omega_z+\frac{1}{2}q_2\omega_y-\frac{1}{2}q_3\omega_x\end{bmatrix}$$

The formula for rotating a vector by a unit quaternion is captured below. To rotate in opposite direction the conjugate of the quaternion is used. These can be used to extract "up" and "forward" vectors from the orientation quaternion.

$$rotate_{\vec{x}}q=q*\begin{bmatrix}0\\ v\end{bmatrix}*q^{-1} \qquad rotate_{\vec{x}}q=\begin{bmatrix}v_x(1-2q_3^2-2q_4^2)+2v_y(q_2q_3-q_1q_4)+2v_z(q_2q_4+q_1q_3)\\ 2v_x(q_2q_3+q_1q_4)+v_y(1-2q_2^2-2q_4^2)+2v_z(q_3q_4-q_1q_2)\\ 2v_x(q_2q_4-q_1q_3)+2v_y(q_3q_4+q_1q_2)+v_z(1-2q_2^2-2q_3^2)\end{bmatrix}$$

$$rotate^{-1}{}_{\vec{x}}q=q^{-1}*\begin{bmatrix}0\\ x\end{bmatrix}*q \qquad rotate^{-1}\vec{x}\,q=\begin{bmatrix}v_x(1-2q_3^2-2q_4^2)+2v_y(q_2q_3+q_1q_4)+2v_z(q_2q_4-q_1q_3)\\ 2v_x(q_2q_3-q_1q_4)+v_y(1-2q_2^2-2q_4^2)+2v_z(q_3q_4+q_1q_2)\\ 2v_x(q_2q_4+q_1q_3)+2v_y(q_3q_4-q_1q_2)+v_z(1-2q_2^2-2q_3^2)\end{bmatrix}$$

The rotation that provides the shortest arc between two vectors is captured below. The cross product does not produce valid results when the vectors are parallel. This condition has to be checked prior as part of the algorithm.

$$rotate_{\vec{x}}\vec{y} = \begin{bmatrix} \frac{s}{2} \\ \frac{x \times v}{s} \end{bmatrix} \quad \text{where} \quad s = \sqrt{2(1 + \vec{x} \cdot \vec{y})}$$

Another method of estimating the quaternion rotation between two vectors is through an optimization algorithm, which minimizes the error for the objective function below. This was used by Sebastian Madgwick and served as a starting point for the IMU algorithm.

$$\min_{q} f\left(rotate^{-1}_{\vec{x}} q - \vec{s}\right)$$

The optimization method used by Sebastian Madgwick was the gradient descent algorithm (see below). It was selected due to its efficiency and simplicity. J is the Jacobian matrix of function, f().

$$\vec{y} = \vec{x} - \mu\left(\frac{\nabla f}{|\nabla f|}\right) \quad \text{Where} \quad \nabla f = J_f f()$$

Full expanded, the function to be minimized is depicted below. This formula was originally derived in Sebastian's paper.

$$f() = \begin{bmatrix} v_x(1 - 2q_3^2 - 2q_4^2) + 2v_y(q_2q_3 + q_1q_4) + 2v_z(q_2q_4 - q_1q_3) - s_x \\ 2v_x(q_2q_3 - q_1q_4) + v_y(1 - 2q_2^2 - 2q_4^2) + 2v_z(q_3q_4 + q_1q_2) - s_y \\ 2v_x(q_2q_4 + q_1q_3) + 2v_y(q_3q_4 - q_1q_2) + v_z(1 - 2q_2^2 - 2q_3^2) - s_z \end{bmatrix}$$

The Jacobian for objective function is:

$$J_f = \begin{bmatrix} 2v_yq_4 - 2v_zq_3 & 2v_yq_3 + 2v_zq_4 & -4v_xq_3 + 2v_yq_2 - 2v_zq_1 & -4v_xq_4 + 2v_yq_1 + 2v_zq_2 \\ -2v_xq_4 + 2v_zq_2 & 2v_xq_3 - 4v_yq_2 + 2v_zq_1 & 2v_xq_2 + 2v_zq_4 & -2v_xq_1 - 4v_yq_4 + 2v_zq_3 \\ 2v_xq_3 - 2v_yq_2 & 2v_xq_4 - 2v_yq_1 - 4v_zq_2 & 2v_xq_1 + 2v_yq_4 - 4v_zq_3 & 2x_xq_2 + 2d_yq_3 \end{bmatrix}$$

For the accelerometer, the vector under rotation *v*, is the acceleration opposing gravity and should point upward. This can be represented by setting its value and making the optimizations captured below.

$$v = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad f() = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - s_x \\ 2(q_3q_4 + q_1q_2) - s_y \\ 1 - 2q_2^2 - 2q_3^2 - s_z \end{bmatrix} \quad J_f = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix}$$

For the magnetometer, the vector under rotation *v*, points towards magnetic north and has a component that points forward and another pointing downward. The angle between the "north" and "down" parts is a function of location (the angle increases as one travels between the north and south pole). The objective function and its Jacobian for the magnetometer can be seen below.

$$v=\begin{bmatrix} v_x \\ 0 \\ v_z \end{bmatrix} \qquad f()=\begin{bmatrix} v_x(1-2q_3^{\,2}-2q_4^{\,2})+2v_z(q_2q_4-q_1q_3)-s_x \\ 2v_x(q_2q_3-q_1q_4)+2v_z(q_3q_4+q_1q_2)-s_y \\ 2v_x(q_2q_4+q_1q_3)+v_z(1-2q_2^{\,2}-2q_3^{\,2})-s_z \end{bmatrix}$$

$$J_f=\begin{bmatrix} -2v_zq_3 & 2v_zq_4 & -4v_xq_3-2v_zq_1 & -4v_xq_4+2v_zq_2 \\ -2v_xq_4+2v_zq_2 & 2v_xq_3+2v_zq_1 & 2v_xq_2+2v_zq_4 & -2v_xq_1+2v_zq_3 \\ 2v_xq_3 & 2v_xq_4-4v_zq_2 & 2v_xq_1-4v_zq_3 & 2v_xq_2 \end{bmatrix}$$

When evaluating this objective function and Jacobian Matrix against theoretical and field vectors, it was determined that the incorporation of the "z" component reduced system performance, especially in scenarios w/ magnetic interference. To minimize this performance loss, the forward component was extracted by subtracting the projection of the magnetic component against the "up" vector, which can be the accelerometer vector or derived from the system quaternion. After this orthonormilization, the equations reduce to the following.

$$v=\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad f()=\begin{bmatrix} 1-2q_3^{\,2}-2q_4^{\,2}-s_x \\ 2(q_2q_3-q_1q_4)-s_y \\ 2(q_2q_4+q_1q_3)-s_z \end{bmatrix} \qquad J_f=\begin{bmatrix} 0 & 0 & -4q_3 & -4q_4 \\ -2q_4 & 2q_3 & 2q_2 & -2q_1 \\ 2q_3 & 2q_4 & 2q_1 & 2q_2 \end{bmatrix}$$