Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

# An Improved Quaternion-Based Kalman Filter for Real-Time Tracking of Rigid Body Orientation

Xiaoping Yun*, Mariano Lizarraga, Eric R. Bachmann† and Robert B. McGhee

Naval Postgraduate School
Monterey, CA 93943

Abstract — This paper presents an improved Kalman filter for real-time tracking of human body motions (see video). An earlier version of the filter was presented at IROS 2001. Since then, the filter has been substantially improved. Real-time tracking of rigid body orientation is accomplished using the MARG (Magnetic, Angular Rate, and Gravity) sensors. A MARG sensor measures the three-dimensional local magnetic field, three-dimensional angular rate, and three dimensional acceleration. A Kalman filter is designed to process measurements provided by the MARG sensors, and to produce real-time orientation represented in quaternions. There are many design decisions as related to choice of state vectors, output equations, process model, etc. The filter design presented in this paper utilizes the Gauss-Newton method for parameter optimization in conjunction with Kalman filtering. The use of the Gauss-Newton method, particularly the reduced-order implementation introduced in the paper, significantly simplifies the Kalman filter design, and reduces computational requirements.

**Keywords:** Kalman filtering, quaternions, motion tracking, orientation, Gauss-Newton method, accelerometers, magnetometers, and angular rate sensors.

## 1 Introduction

Owing to the availability of low-cost, small-size MEMS sensors, it is possible to build self-contained inertial sensors in the size of a wrist watch that can accurately track orientation in real time. One such effort is the development of the MARG sensors [1]. A magnetic, angular rate, and gravity (MARG) sensor is an integrated sensor system that consists of a triad of magnetometers, a triad of angular rate sensors, a triad of accelerometers, a microcontroller, and associated signal conditioning circuitry. The current design of the MARG sensors is less than one cubic inch, and the size becomes smaller with each iteration of re-designing. The MARG sensors can be applied where real-time measurements of orientation are required. Due to their small size, an emerging application of the MARG sensors is human body motion tracking for inserting humans in motion into virtual environments. By attaching one MARG sensor to each major body segment, motions of the entire human body can be captured in real time.

This paper is concerned with filter design aspect of the MARG sensor development. The role of a filter in this application is to fuse nine measurements generated by magnetometers, angular rate sensors, and accelerometers to produce a quaternion or other suitable representations of orientation. Kalman filtering is mostly applied in this context. However, it should be noted that a Kalman filter designed for tracking aircraft will not necessarily be appropriate for tracking human body motions. This paper describes a Kalman filter algorithm that uses angular rate and quaternion measurements. While angular rate measurements are directly available from angular rate sensors, quaternion measurements are calculated from magnetometer and accelerometer measurements by adopting an algorithm reported in [2].

There are many studies on human motion tracking using inertial sensors. A study of human motion tracking using accelerometers alone was reported by Lee and Ha [3]. During motions involving small linear accelerations, a set of tri-axial accelerometers was used to determine 2-DOF rotation angles. During motions accompanied by higher accelerations, a technique is described that involves the use of two sets of tri-axial accelerometers on a single rigid-body to differentiate gravitational acceleration from motion related linear acceleration. Rehbinder and Hu [4] described an attitude estimation algorithm based on the use of angular rate sensors and accelerometers. In this case, drift in heading estimation was unavoidable due to a lack of additional complementary sensors such as magnetometers. Gebre-Egziabher et al. [2] described an attitude determination algorithm for aircraft applications. The algorithm is based on a quaternion formulation of Wahba's problem [5, 6], where magnetometer and accelerometer measurements are used to determine attitude without the use of gyros. A Kalman filter implementation of the algorithm was also presented. Wahba's problem is to determine three-axis attitude from multiple vector observations in two coordinate frames [5]. Various solutions and algorithms to Wahba's problem have been proposed for spacecraft attitude determination [7, 8, 9].

*To whom all correspondence should be addressed: Department of Electrical and Computer Engineering, Code EC/Yx, Naval Postgraduate School, Monterey, CA 93943-5121.

†Department of Computer Science and Systems Analysis, Miami University, Oxford, OH 45056.

Use of inertial sensors for mobile robot navigation has been actively investigated in the literature [10, 11, 12].

The remaining part of the paper is organized as follows. Two different approaches to the Kalman filter design as related to choice of measurement equations are outlined in Section 2. The focus of Section 3 is on the reduced order implementation of the Gauss-Newton method for computing quaternion measurements from accelerometer and magnetometer measurements. Prior to doing so, some necessary background materials are first introduced. The Kalman filter design is presented in Section 4, and followed by concluding remarks in Section 5.

## 2 Different Approaches to the Kalman Filter Design

Two alternative approaches to the Kalman filter design for rigid body orientation estimation were presented in [13]. The state vector for both approaches is the same, and is a 7-dimensional vector consisting of three components of angular rates and four components of quaternions. Quaternions are used to represent orientation of the rigid body to which a MARG sensor is attached. The difference between the two approaches is in the measurement or output equation for the Kalman filter. In the first approach, the output is a 9-dimensional vector, consisting of 3-dimensional angular rate, 3-dimensional acceleration, and 3-dimensional local magnetic field. The first three components of the output equation (angular rate portion) is linearly related to the state vector. However, the other six components of the output equation are nonlinearly related to the state vector. The nonlinear relationship is quite complicated. As a result, the extended Kalman filter designed with this output equation is computationally inefficient.

The second approach uses the Newton method or Gauss-Newton method to find a corresponding quaternion for each pair of accelerometer and magnetometer measurements. The computed quaternion is then combined with the angular rate measurements, and presented to the Kalman filter as its measurements. By doing so, the output equations for the Kalman filter is linear, and the overall Kalman filter design is greatly simplified. In the original design as presented in [13], both the Newton method and Gauss-Newton method were considered. Subsequently, it was determined that the Gauss-Newton method is more appropriate for this application. It is the approach that will be used in this paper. Furthermore, a reduced-order implementation of the Gauss-Newton method is possible and described in detail in this paper.

## 3 Computing Quaternions from Accelerometer and Magnetometer Measurements

In this section, an algorithm for computing orientation represented by quaternions from accelerometer and magnetometer measurements is presented. This algorithm is depicted as the Gauss-Newton method block in Figure 1. The Gauss-Newton method itself and some quaternion notations are first presented.

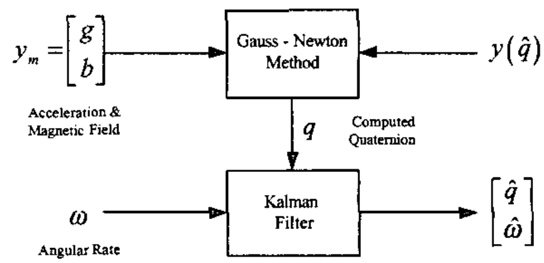


Figure 1: Overall filter diagram.

### 3.1 Gauss-Newton Method

The nonlinear least squares minimization problem is typically stated as [14]:

$$\min_x f(x) = \frac{1}{2}\varepsilon(x)^T\varepsilon(x) \qquad (1)$$

where $x$ is an $n$-dimensional vector, $\varepsilon(x) = [\varepsilon_1(x) \ \cdots \ \varepsilon_m(x)]^T$ is an $m$-dimensional vector-valued nonlinear function in $x$, and $m > n$. $f(x)$ is generally referred as the "cost function" and $\varepsilon(x)$ is referred as the "error function". The solution to such a problem using the Gauss-Newton Method is given by an iterative update equation [14]:

$$x_+ = x_c - \left(J(x_c)^T J(x_c)\right)^{-1} J(x_c)^T\varepsilon(x_c) \qquad (2)$$

where $x_c$ is the current estimate, $x_+$ is the new estimate, and $J(x_c)$ refers to the Jacobian of the error function $\varepsilon(x)$ evaluated at $x_c$:

$$J(x) = \begin{bmatrix} \frac{\partial\varepsilon_1(x)}{\partial x_1} & \cdots & \frac{\partial\varepsilon_1(x)}{\partial x_n} \\ \frac{\partial\varepsilon_2(x)}{\partial x_1} & & \vdots \\ \vdots & \ddots & \vdots \\ \frac{\partial\varepsilon_m(x)}{\partial x_1} & \cdots & \frac{\partial\varepsilon_m(x)}{\partial x_n} \end{bmatrix}. \qquad (3)$$

### 3.2 Rotation Using Quaternions

A quaternion may be used to rotate a 3-dimemsional vector $u$ using quaternion multiplications [15]:

$$u_{rotated} = q^{-1} \otimes u \otimes q \qquad (4)$$

where the unit quaternion q is defined as:

$$q = \cos\left(\frac{\theta}{2}\right) + w\sin\left(\frac{\theta}{2}\right) \qquad (5)$$

with $\theta$ being the angle that the vector $u$ is to be rotated and $w$ a unit vector that defines the axis of rotation. It is convenient and custom to use a unit quaternion in Equation (4), but it is not necessary. As a matter

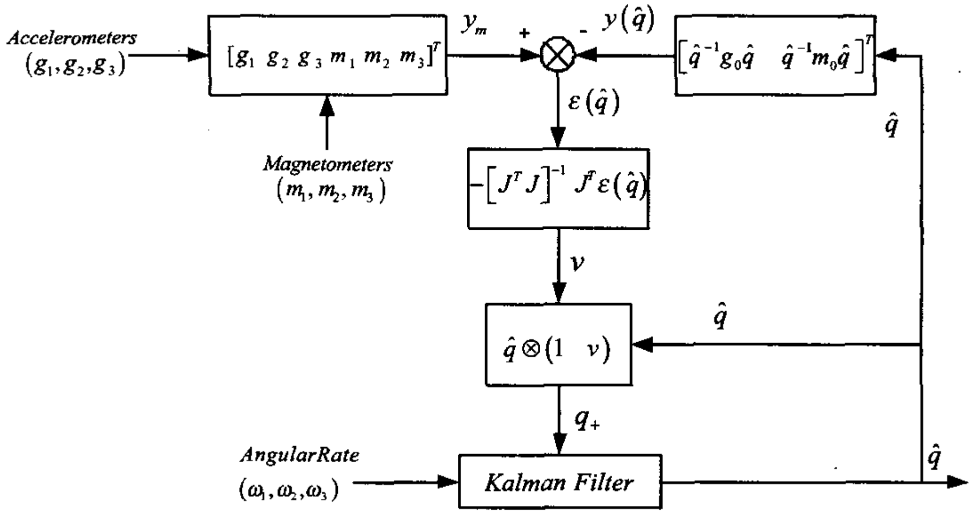


Figure 2: Reduced-order Gauss-Newton Algorithm and Kalman filter.

of fact, it can be shown that the multiplication of $q$ in Equation (4) by any non zero scalar achieves the same rotation [15]. If the rotation angle $\theta$ is small, the so-called error quaternion may be used for the purpose of rotation [2], and it is obtained by dividing the unit quaternion $q$ in Equation (5) by its scalar part:

$$\begin{aligned} q_v &= \frac{\cos\left(\frac{\theta}{2}\right) + w \sin\left(\frac{\theta}{2}\right)}{\cos\left(\frac{\theta}{2}\right)} = 1 + w \tan\left(\frac{\theta}{2}\right) \\ &= \left[\ 1 \quad v_1 \quad v_2 \quad v_3\ \right] = \left[\ 1 \quad v\ \right]. \end{aligned} \tag{6}$$

### 3.3 Rotation Matrix

The rotation of a vector can be achieved by multiplying it by a quaternion as in Equation (4). The same rotation may also be achieved by multiplying it by a rotation matrix [16]

$$u_{rotated} = M_{rotation} u. \tag{7}$$

The quaternion $q$ and rotation matrix $M_{rotation}$ are related as follows:

$$M_{rotation}(q) = \frac{1}{\|q\|} \cdot$$

$$\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\left(q_1 q_2 + q_3 q_0\right) & 2\left(q_1 q_3 - q_2 q_0\right) \\ 2\left(q_1 q_2 - q_3 q_0\right) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2\left(q_2 q_3 + q_0 q_1\right) \\ 2\left(q_1 q_3 + q_2 q_0\right) & 2\left(q_2 q_3 - q_1 q_0\right) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

where $q_0$ is the scalar part, $[q_1\ \ q_2\ \ q_3]$ is the vector part of the quaternion $q$, and $\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$. The rotation matrix shown above does not require that $q$ be a unit quaternion.

### 3.4 Direct Application of the Gauss-Newton Method

Given measurements of accelerometers and magnetometers in a body cooridnate system, the problem here is to find the corresponding quaternion representing the orientation of the body in a static manner. This problem can be cast into a nonlinear minimization problem that can be solved using the Gauss-Newton Method. As shown in Figure 2, the error function is given by:

$$\varepsilon\left(\hat{q}\right) = y_m - y\left(\hat{q}\right) \tag{8}$$

where $y_m$ is a 6-dimensional vector constructed by concatenation of the accelerometers and magnetometers measurements. This vector is considered constant for the purpose of minimization. $y(\hat{q})$ is the body-fixed acceleration and magnetic field computed by rotating $g_0$ and $m_0$ using the current estimate of the quaternion $\hat{q}$. $g_0$ is the gravity vector in the earth coordinate, and $m_0$ is the local magnetic vector in the earth coordinate. For human body motion tracking applicaitons, $g_0$ and $m_0$ can be considered to be constants. The problem is to find a new quaternion estimate such that the cost function of Equation (1) with the error function of Equation (8) is minimized.

Applying the Gauss-Newton Method to the current problem, the new estimate will be equal to the current estimate minus some increment which is a function of the error function $\varepsilon$:

$$q_+ = q_c - \left(J(q_c)^T J(q_c)\right)^{-1} J(q_c)^T \varepsilon(q_c) \tag{9}$$

where $q_c$ is the current estimate. In the first iteration of executing the Gauss-Newton algorithm, $q_c$ is initialized to the output of the Kalman filter, $\hat{q}$.

$$J\left(q_c\right) = \frac{\partial \varepsilon(q)}{\partial q}\Big|_{q=q_c} = -\frac{\partial y\left(q\right)}{\partial q}\Big|_{q=q_c}.$$

The above updating equation can be stated in a more compact fashion:

$$q_+ = q_c + \Delta q \tag{10}$$

where

$$\Delta q = -\left(J(q_c)^T J(q_c)\right)^{-1} J(q_c)^T \varepsilon(q_c)$$

which can be phrased as: "the new estimate of the quaternion that minimizes the error function is equal to the current estimate available plus an increment". It is noted that the dimension of Jacobian is $6 \times 4$, and the matrix to be inverted is thus $4 \times 4$.

While the above algorithm of finding a new estimate of the quaternion is straightforward, it is nevertheless not computationally efficient. It requires the computation of the inverse of a $4 \times 4$ matrix, which is not an issue if the algorithm is to be implemented on a desktop computer. Since the target processor for implementing the algorithm is a microcontroller such as the 16-bit RISC microcontroller MSP430F149 from Texas Instruments, an alternative and more efficient algorithm based on the work of [2] is described in the following subsection.

### 3.5 Reduced-Order Implementation of the Gauss-Newton Method

Vector $u$ in Equation (4) is rotated by quaternion $q$. The rotated vector can be rotated again by another quaternion. That is, quaternion rotations can be compounded as follows [15]:

$$u_{rotated} = q_{r2}^{-1} \otimes (q_{r1}^{-1} \otimes u \otimes q_{r1}) \otimes q_{r2}. \qquad (11)$$

In this case, vector $u$ is first rotated by $q_{r1}$, followed by another rotation by $q_{r2}$. The above equation can be rewritten as:

$$u_{rotated} = (q_{r1} \otimes q_{r2})^{-1} \otimes u \otimes (q_{r1} \otimes q_{r2}). \qquad (12)$$

This equation simply states that the overall compound rotation quaternion is the product of two individual quaternions.

Applying this concept to the problem of finding new quaternion estimate $q_+$, it is possible to consider $q_+$ as a compound rotation, starting from the current quaternion estimate followed by an error quaternion of the form shown in Equation (6):

$$q_+ = q_c \otimes q_v = q_c \otimes [1 \quad v]. \qquad (13)$$

Comparing Equations (10) and (13), $\Delta q$ is four dimensional whereas $v$ is three dimensional. The goal now is to find a three-dimensional $v$ such that the cost function is minimized. To do so, the right-hand side of Equation (13) is used to replace $\hat{q}$ in Equation (8) to obtain the error function in terms of $v$:

$$\varepsilon(v) = y_m - y(q_c \otimes [1 \quad v]). \qquad (14)$$

The cost function is then:

$$\min_v f(v) = \frac{1}{2}\varepsilon(v)^T \varepsilon(v). \qquad (15)$$

The Gauss-Newton method may be utilized again to minimize the cost function with respect to the variable $v$. The update equation is given by:

$$v_+ = v_c - \left(J(v_c)^T J(v_c)\right)^{-1} J(v_c)^T \varepsilon(v_c). \qquad (16)$$

It is noted that if the vector part $v$ of $q_v$ is zero in Equation (13), no rotation is performed and $q_+ = q_c$. Thus, the starting value of $v$ in each iteration is zero. As a result, the update equation is further simplified to:

$$v_+ = -\left(J(v_c)^T J(v_c)\right)^{-1} J(v_c)^T \varepsilon(v_c). \qquad (17)$$

The Jacobian $J(v_c)$ in this case is surprisingly simple [17, 2]. The result is given below, with detailed derivation provided in Appendix.

$$J(v_c) = -\frac{\partial y(v)}{\partial v}\Big|_{v=0} = -\begin{bmatrix} 0 & -2y_3 & 2y_2 \\ 2y_3 & 0 & -2y_1 \\ -2y_2 & 2y_1 & 0 \\ 0 & -2y_6 & 2y_5 \\ 2y_6 & 0 & -2y_4 \\ -2y_5 & 2y_4 & 0 \end{bmatrix} \qquad (18)$$

where $y_i$, $i = 1, \ldots, 6$, are components of $y(\hat{q})$ in Equation (8). The Jacobian now is $6 \times 3$ dimensional, the matrix to be inverted is $3 \times 3$. Equation (13) can now be rewritten as:

$$q_+ = q_c \otimes \left[1 \quad -\left(J(v_c)^T J(v_c)\right)^{-1} J(v_c)^T \varepsilon(v_c)\right]. \qquad (19)$$

The updated quaternion $q_+$ from Equation (19) will then be used as measurements to the Kalman filter as discussed in the next section. The algorithm presented in this subsection is depicted in Figure 2.

## 4 Kalman Filter Formulation

Figure 3 depicts the process model for the filter design. $\omega$ is the three dimensional angular rate vector, and is modeled as the output of a first order linear system driven by a three-dimensional white noise vector $w_n$. The angular rate $\omega$ and the quaternion derivative $\dot{q}$ is related by the well-known identity [15]:

$$\dot{q} = \frac{1}{2}\omega \otimes q. \qquad (20)$$

The quaternion derivative is integrated, and the resultant quaternion is then normalized to a unit quaternion.

The state vector $x$ is 7-dimensional, with the first three components being the angular rate $\omega$, and the last four components being the quaternion $q$. Based on Figure 3, the state equations are then given by:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \frac{1}{\tau}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_{n1} \\ w_{n2} \\ w_{n3} \end{bmatrix}\right) \qquad (21)$$

$$\begin{bmatrix} \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \otimes \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \qquad (22)$$

where $\otimes$ denotes quaternion multiplication.

As a result of Section 3.5, quaternions may be treated as measurements available to the Kalman filter. Combining with the angular rate measurements, the Kalman filter has a total of seven measurements. Let $z$ denote the measurement vector. The measurement equation for the Kalman filter is given by:

$$z = Hx + v_n \qquad (23)$$

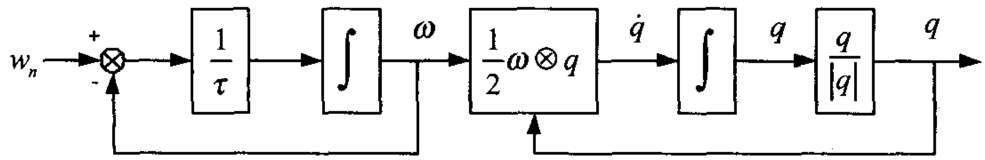


Figure 3: Kalman filter process model.

where the measurement matrix $H$ is simply the $7 \times 7$ identity matrix and $v_n$ is the vector of measurement noise.

To apply the Kalman filtering theory and for computer implementation, the state equations (21) and (22) and the measurement equation (23) are first discretized:

$$x_{k+1} = \Phi_k \, x_k + w_k \tag{24}$$

$$z_k = H_k \, x_k + v_k \tag{25}$$

where $x_k = x(t_k)$, $z_k = z(t_k)$, $\Phi_k$ is the state transition matrix, $H_k = H$, and $w_k$ and $v_k$ are discrete white noises. The standard Kalman filter equations can then be utilized to implement the filter.

## 5 Conclusion

A quaternion-based Kalman filter for rigid body orientation tracking is presented. The novelty of the design lies in the combined use of the Gauss-Newton method and Kalman filter. Because of this combination, the Kalman filter is significantly simplified and lends itself to embedded implementation on low-cost microcontrollers. It should be noted that the filter described in the paper is specifically designed for human body motion tracking, and can not be used for tracking systems (e.g., aircraft or spacecraft) that have significantly different dynamic characteristics.

## Acknowledgment

This research was supported in part by Army Research Office (ARO), Navy Modeling and Simulation Management Office (N6M), and Naval Surface Warfare Center (NSWC).

## References


[1] Eric R. Bachmann, Xiaoping Yun, Doug McKinney, Robert B. McGhee, and Michael J. Zyda. Design and implementation of MARG sensors for 3-DOF orientation measurement of rigid bodies. In *Proceedings of 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, May 2003.

[2] Demoz Gebre-Egziabher, Gabriel H. Klkaim, J.D. Powell, and Bradford W. Parkinson. A gyro-free quaternion-based attitude determination system suitable for implementation using low cost sensors. In *Proceedings of IEEE 2000 Position Location and Navigation Symposium*, pages 185–192, San Diego, CA, March 2000.

[3] J. Lee and I. Ha. Sensor fusion and calibration for motion captures using accelerometers. In *ICRA1999*, pages 1954–1959, Detroit, MI, May 1999.

[4] Herik Rehbinder and Xiaoming Hu. Drift-free attitude estimation for accelerated rigid bodies. In *Proceedings of 2001 IEEE International Conference on Robotics and Automation*, pages 4244–4249, Seoul, Korea, May 2001.

[5] Grace Wahba. Problem 65-1: A least squares estimate of satellite attitude. *SIAM Review*, 7(3):409, July 1965.

[6] J.L. Farrell, J.C. Stuelpnagel, R.H. Wessner, and J.R. Velman. Solution to problem 65-1: A least squares estimate of satellite attitude. *SIAM Review*, 8(3):384–386, July 1966.

[7] M.D. Shuster and S.D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance and Control*, 4(1):70–77, January-February 1981.

[8] Daniele Mortari. ESOQ: a closed-form solution to the wahba problem. *The Journal of the Astranautical Sciences*, 45(2):195–204, April-June 1997.

[9] F. Landis Markley and Daniele Mortari. Quaternion attitude estimation using vector observations. *The Journal of the Astranautical Sciences*, 48(2 and 3):359–380, April-September 2000.

[10] Billur Barshan and Hugh F. Durrant-Whyte. Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342, June 1995.

[11] Salah Sukkarieh, Eduardo M. Nebot, and Hugh F. Durrant-Whyte. Achieving integrity in an INS/GPS navigation loop for autonomous land vehicle applications. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 3437–3442, Leuven, Belgium, May 1998.

[12] Stergios I. Roumeliotis, Gaurav S. Sukhatme, and George A. Bekey. Smoother based 3D attitude estimation for mobile robot localization. In *Proceedings of 1999 IEEE International Conference on Robotics and Automation*, pages 1979–1986, Detroit, MI, May 1999.

[13] Joao Louis Marins, Xiaoping Yun, Eric R. Bachmann, Robert B. McGhee, and Michael J. Zyda. An extended Kalman filter for quaternion-based orietation estimation using MARG sensors. In *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2003–2011, Maui, Hawaii, October 2001.

[14] J.E. Dennis, Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall, Englewood, NJ, 1983.

[15] Jack B. Kuipers. *Quaternions and Rotation Sequences.* Princeton University Press, Princeton, NJ, 1999.

[16] John J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison Wesley Publishing Company, 2nd edition, 1989.

[17] Eric R. Bachmann, Robert B. McGhee, Xiaoping Yun, and Michael J. Zyda. Inertial and magnetic posture tracking for inserting humans into networked virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST 2001)*, pages 9–16, Banff, Alberta, Canada, November 2001.

## Appendix: Derivation of Jacobian Matrix

In this appendix, the Jacobian matrix of Equation (18) is derived in detail. It is recalled that $y(q_+)$ is the body-fixed acceleration and magnetic field computed from the quaternion $q_+$, and is given by

$$y(q_+) = \begin{bmatrix} q_+^{-1} \otimes g_0 \otimes q_+ \\ q_+^{-1} \otimes m_0 \otimes q_+ \end{bmatrix} \qquad (26)$$

where $g_0$ is the constant gravity vector in the earth coordinate, and $m_0$ is the constant local magnetic vector in the earth coordinate. In the above, quaternion rotations are used to rotate $g_0$ and $m_0$ into the body coordinate. Because of Equation (13), $y(q_+)$ can be expanded as:

$$y(q_+) = \begin{bmatrix} (q_c \otimes [1 \;\; v])^{-1} \otimes g_0 \otimes q_c \otimes [1 \;\; v] \\ (q_c \otimes [1 \;\; v])^{-1} \otimes m_0 \otimes q_c \otimes [1 \;\; v] \end{bmatrix}$$

which can be further stated as:

$$y(q_+) = \begin{bmatrix} [1 \;\; v]^{-1} \otimes (q_c^{-1} \otimes g_0 \otimes q_c) \otimes [1 \;\; v] \\ [1 \;\; v]^{-1} \otimes (q_c^{-1} \otimes m_0 \otimes q_c) \otimes [1 \;\; v] \end{bmatrix}$$

Recognizing the fact that $\left(q_c^{-1} \otimes g_0 \otimes q_c\right)$ and $\left(q_c^{-1} \otimes m_0 \otimes q_c\right)$ are the values of $y(q_c)$, the above equation can be finally stated as:

$$y(q_+) = \begin{bmatrix} [\,1 \;\; v\,]^{-1} \otimes [\,y_1 \;\; y_2 \;\; y_3\,] \otimes [\,1 \;\; v\,] \\ [\,1 \;\; v\,]^{-1} \otimes [\,y_4 \;\; y_5 \;\; y_6\,] \otimes [\,1 \;\; v\,] \end{bmatrix}$$

Now using the rotation matrix introduced in section 3.3, $y(q_+)$ can be written in terms of rotation matrices as:

$$y(q_+) = y(v) = \begin{bmatrix} M_{rotation}([1 \;\; v]) \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \\ M_{rotation}([1 \;\; v]) \begin{bmatrix} y_4 \\ y_5 \\ y_6 \end{bmatrix} \end{bmatrix}$$

The Jacobian is defined as:

$$J(v) = -\frac{\partial y(v)}{\partial v} = -\begin{bmatrix} \frac{\partial y(v)}{\partial v_1} & \frac{\partial y(v)}{\partial v_2} & \frac{\partial y(v)}{\partial v_3} \end{bmatrix}$$

Carrying differentiation and evaluating the Jacobian at $v = v_c = 0$, it follows

$$J(v)\,|_{v=0} = -\begin{bmatrix} 0 & -2y_3 & 2y_2 \\ 2y_3 & 0 & -2y_1 \\ -2y_2 & 2y_1 & 0 \\ 0 & -2y_6 & 2y_5 \\ 2y_6 & 0 & -2y_4 \\ -2y_5 & 2y_4 & 0 \end{bmatrix}$$

which is exactly Equation (18).