Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Implementation and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking

Xiaoping Yun, Conrado Aparicio, Eric R. Bachmann[+], and Robert B. McGhee

Naval Postgraduate School, Monterey, CA 93943, USA
[+]Miami University, Oxford, OH 45056, USA

***Abstract* - A human body motion tracking system based on use of the MARG (Magnetic, Angular Rate, and Gravity) sensors has been under development at the Naval Postgraduate School and Miami University. The design of a quaternion-based Kalman filter for processing the MARG sensor data was described in [1]. This paper presents the real-time implementation and testing results of the quaternion-based Kalman filter. Experimental results validate the Kalman filter design, and show the feasibility of the MARG sensors for real-time human body motion tracking.**

Index Terms – Quaternion-based Kalman filter, human body motion tracking, MARG sensors, inertial/magnetic sensors.

I. Introduction

Inertial/magnetic sensor modules can be used to estimate orientation of a rigid body relative to an Earth fixed reference frame without the need of an artificially generated reference. The estimates produced are based entirely on inertial quantities related to the motion and attitude of the module and the orientation of the ambient magnetic field relative to the module. If a single sensor module is placed on each of the segments of an articulated rigid body, the "posture" of the structure can be determined. Such "sourceless" orientation tracking has significant advantages over other methods owing to its low susceptibility to various sources of noise and lack of range limitations [8,9]. If the human body is modeled as articulated rigid bodies consisting of approximately fifteen segments, posture and gait could be accurately tracked and measured over an unlimited area. Thus, this methodology of body tracking could have important applications in virtual environments, robotic teleoperation, personal navigation, and human monitoring applications [10].

The Naval Postgraduate School and Miami University have teamed up to develop an inertial/magnetic sensor module called the MARG sensor for tracking human body motions in real time [2]. MARG (Magnetic, Angular Rate, and Gravity) sensor modules contain three magnetometers, three angular rate sensors, and three accelerometers. Each sensor type is orthogonally mounted in a triad. This paper presents the implementation and experimental testing results for a quaternion-based Kalman filter designed for the MARG sensors.

An earlier version of the Kalman filter implemented here was described in [1]. The overall filter design remains unchanged. However, some portions of the filter design have been modified. In particular, the original design used a reduced-order Gauss-Newton method to compute an orientation quaternion from accelerometer and magnetometer measurements. This part of the filter was first modified to use the QUEST Algorithm [3] and later the Factored Quaternion Algorithm [4,5]. The QUEST algorithm [3,6] was created to determine the attitude of a rigid body in reference to a fixed coordinate system, using a set of measurement vectors. The algorithm computes a rotation (attitude) quaternion that rotates the measurement vectors to match the reference vectors. More recently, the Factored Quaternion Algorithm [4] was derived. It has the same goal as the QUEST algorithm but orientation estimate are derived through the measurement of sequential rotations about three orthogonal axes. It has been shown that the Factored Quaternion Algorithm has equal or better performance than the QUEST algorithm in estimating orientation quaternions with MARG sensor measurements [5]. Nevertheless, the Factored Quaternion Algorithm is computationally more efficient by about 25%, and is thus used as part of the filter design in the latest implementation [5].

This paper is organized as follows. Section II presents the process model of the Kalman filter for human body motion tracking. Section III describes implementation issues of the Kalman filter with a focus on how the nonlinear process model was first linearized and then discretized. Experimental modeling of the process noise covariance matrix and the measurement noise covariance matrix is also detailed. Section IV reports the MATLAB simulation and offline testing results of the Kalman filter. Section V describes the implementation and testing results, followed by conclusions in section VI.

II. Kalman filter process model

The process model of the quaternion-based Kalman filter presented in [1] will be briefly reviewed in this section. A diagram of the process model is shown in Figure 1. In this model, the angular rates ω in body coordinates are assumed to be generated by a first-order linear system with a white noise forcing function w. The time constant of the first-order linear system is τ. The orientation estimate produced by the filter is $\hat{q}$. The angular rates ω and the quaternion derivative $\dot{q}$ are related by [7]:

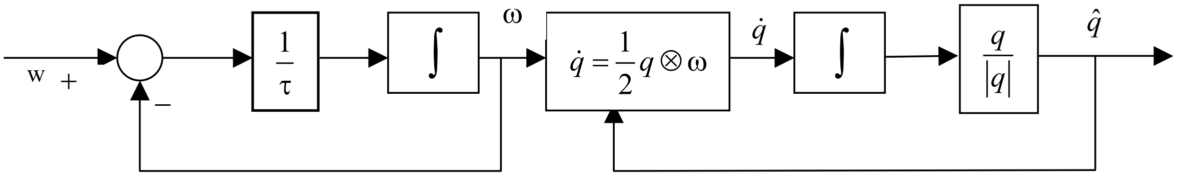

Figure 1. Kalman Filter Process Model.

$$\dot{q}=\frac{1}{2}q\otimes\omega \tag{1}$$

where q is the orientation quaternion in Earth coordinates, and $\otimes$ represents quaternion multiplication. In order to take advantage of computational simplifications and efficiencies possible of unit quaternions, the quaternion is normalized to unit length in the last step of the process model. It is noted that quaternions are used to represent orientation in the filter design because quaternions do not have the singularity problem associated with Euler angles and eliminate the computational expenses related to approximation of transcendental functions.

The state vector is defined as a 7-dimensional vector with the first three components being the angular rates and the last four being the elements of the quaternion. The process model expressed in terms of state equations is characterized as follows:

$$\dot{x}_i=-\frac{1}{\tau_i}x_i+\frac{1}{\tau_i}w_i(t) \quad i=1,2,3 \tag{2}$$

for the angular rates, and

$$\dot{x}_4=-\frac{1}{2}\left[x_1x_5+x_2x_6+x_3x_7\right], \tag{3}$$

$$\dot{x}_5=\frac{1}{2}\left[x_1x_4-x_2x_7+x_3x_6\right], \tag{4}$$

$$\dot{x}_6=\frac{1}{2}\left[x_1x_7+x_2x_4-x_3x_5\right], \tag{5}$$

$$\dot{x}_7=\frac{1}{2}\left[-x_1x_6+x_2x_5+x_3x_4\right] \tag{6}$$

for the quaternion components.

The MARG sensor provides a 9-dimensional measurement vector, consisting of three elements of the linear acceleration vector, three elements of the local magnetic field, and three elements of the angular rate vector. If this nine-dimensional measurement vector is provided directly to the Kalman filter as measurements, the measurement equations are nonlinear and the resulting Kalman filter becomes complex and computationally expensive. An alternative approach to the Kalman filter design was suggested in [1]. This approach uses the Newton method or a reduced-order Gauss-Newton method to find a quaternion corresponding to each set of accelerator and magnetometer measurements. These computed quaternion and angular rate measurements are then presented to the Kalman filter as measurements. As a result, the measurement equation for the Kalman filter is linear and is given by:

$$z=Hx+v(t) \tag{7}$$

where z is the seven-dimensional measurement vector, H is a 7×7 identity matrix, and v is the vector of measurement noises.

Although the reduced-order Gauss-Newton method presented in [1] was considerably more efficient than the full-order Gauss-Newton method, it still is an iterative method that needs to be executed several times before convergence occurs. Following additional work, the reduced-order Gauss-Newton method was replaced by the QUEST Algorithm [3,6], and more recently by the Factored Quaternion Algorithm [4]. Both the QUEST Algorithm and Factored Quaternion Algorithm take a set of the accelerometer and magnetometer measurements and produce an orientation quaternion. They are appropriate for orientation estimation in static or slow moving applications where linear acceleration does not comprise a significant part of the total acceleration measurements. The Factored Quaternion Algorithm is computationally about 25% more efficient than the QUEST Algorithm.

III. Kalman filter implementation

In this section, the implementation of the Kalman filter based on the process model presented in the previous section will be described. It is noted that although Equation (2) is linear, Equations (3) to (6) are nonlinear. As a result, an extended Kalman filter must be used. Additionally, these continuous equations must be discretized for digital implementations.

A. Discrete Extended Kalman Filter

Equations (2) to (6) can be written in vector form as follows:

$$\dot{x}=f(x)+w(t). \tag{8}$$

This nonlinear state equation is linearized along the currently estimated trajectory $\hat{x}$:

$$\Delta\dot{x}=\left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}}\Delta x+w(t), \tag{9}$$

where the actual trajectory, x , is the sum of estimated trajectory $\hat{x}$ and the small increment Δx

$$x=\hat{x}+\Delta x\,. \tag{10}$$

Equation (9) is linear, but it is still in the continuous time domain. The next step is to discretize it to obtain a discrete

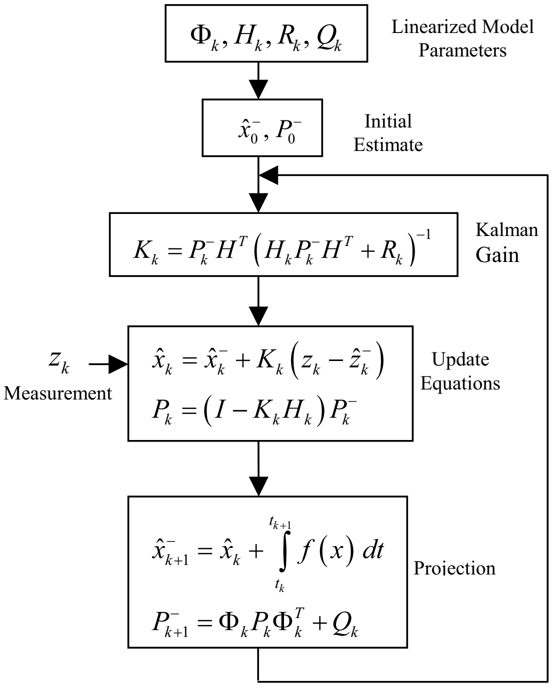
Φ_k, H_k, R_k, Q_k
Linearized Model Parameters
$\hat{x}_0^-, P_0^-$
Initial Estimate
$K_k = P_k^- H^T \left(H_k P_k^- H^T + R_k\right)^{-1}$
Kalman Gain
z_k
Measurement
$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - \hat{z}_k^-\right)$
$P_k = \left(I - K_k H_k\right) P_k^-$
Update Equations
$\hat{x}_{k+1}^- = \hat{x}_k + \int_{t_k}^{t_{k+1}} f(x)\, dt$
$P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k$
Projection

Figure 2. Diagram of the extended Kalman filter.

time process model. Let Δt be the sampling interval. Then the difference equation corresponding to the differential equation (9) is given by:

$$\Delta x(t_{k+1}) = \Phi_k \Delta x(t_k) + w(t_k) \tag{11}$$

where the discrete state transition matrix is:

$$\Phi_k = \begin{bmatrix} e^{-\frac{\Delta t}{\tau_1}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{-\frac{\Delta t}{\tau_2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{-\frac{\Delta t}{\tau_3}} & 0 & 0 & 0 & 0 \\ -\frac{\hat{x}_5\Delta t}{2} & -\frac{\hat{x}_6\Delta t}{2} & -\frac{\hat{x}_7\Delta t}{2} & 1 & -\frac{\hat{x}_1\Delta t}{2} & -\frac{\hat{x}_2\Delta t}{2} & -\frac{\hat{x}_3\Delta t}{2} \\ \frac{\hat{x}_4\Delta t}{2} & -\frac{\hat{x}_7\Delta t}{2} & \frac{\hat{x}_6\Delta t}{2} & \frac{\hat{x}_1\Delta t}{2} & 1 & \frac{\hat{x}_3\Delta t}{2} & -\frac{\hat{x}_2\Delta t}{2} \\ \frac{\hat{x}_7\Delta t}{2} & \frac{\hat{x}_4\Delta t}{2} & -\frac{\hat{x}_5\Delta t}{2} & \frac{\hat{x}_2\Delta t}{2} & -\frac{\hat{x}_3\Delta t}{2} & 1 & \frac{\hat{x}_1\Delta t}{2} \\ -\frac{\hat{x}_6\Delta t}{2} & \frac{\hat{x}_5\Delta t}{2} & \frac{\hat{x}_4\Delta t}{2} & \frac{\hat{x}_3\Delta t}{2} & \frac{\hat{x}_2\Delta t}{2} & -\frac{\hat{x}_1\Delta t}{2} & 1 \end{bmatrix}$$

and the elements of discrete white noises are given by:

$$w_i\left(t_k\right) = \begin{cases} \int_{t_k}^{t_{k+1}} e^{-\frac{\left(t_{k+1}-\gamma\right)}{\tau_i}} w_i\left(\gamma\right) d\gamma & i = 1,2,3 \\ 0 & i = 4,5,6,7 \end{cases} \tag{12}$$

Equation (7) is linear. Thus, linearization is not needed. The corresponding discrete process model equation is simply given as:

$$z_k = H_k x_k + v_k \tag{13}$$

A standard discrete Kalman filter may now be designed for the discrete process equation (11) and the discrete measurement equation (13). A complete diagram of the extended Kalman filter is depicted in Figure 2.

B. Modeling of Process and Measurement Noises

In order to implement the Kalman filter described above, it is necessary to determine values of the process noise covariance matrix Q_k and the measurement noise covariance matrix R_k. These matrices represent the confidence in the system model and the measurement data, respectively.

The process noise matrix Q_k is given by:

$$Q_k = E\left[w\left(t_k\right) w\left(t_k\right)^T\right] \tag{14}$$

where E is the expectation operator, and $w(t_k)$ is the discrete white noise of Equation (12).

It is noted that $w_i\left(\gamma\right)$ in Equation (12) is the continuous, independent white noise process of Equations (2) to (6), with zero mean and variance D_i. Therefore,

$$E\left[w_i\left(t\right) w_j\left(\tau\right)\right] = \begin{cases} D_i \delta\left(t - \tau\right) & i = j \\ 0 & i \neq j \end{cases} \tag{15}$$

This implies that the process noise matrix is a diagonal matrix with non-zero elements only in the first three positions of the main diagonal, and can be computed using Equations (14) to (15) as

$$Q_k = \begin{bmatrix} q_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_{33} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

where q_{11}, q_{22}, and q_{33} are given by:

$$q_{11} = E\left[w_1\left(t_k\right) w_1\left(t_k\right)\right] = \frac{D_1}{2\tau_1}\left(1 - e^{-\frac{2\Delta t}{\tau_1}}\right), \tag{17}$$

$$q_{22} = E\left[w_2\left(t_k\right) w_2\left(t_k\right)\right] = \frac{D_2}{2\tau 2}\left(1 - e^{-\frac{2\Delta t}{\tau_2}}\right), \tag{18}$$

and

$$q_{33} = E\left[w_3\left(t_k\right) w_3\left(t_k\right)\right] = \frac{D_3}{2\tau_3}\left(1 - e^{-\frac{2\Delta t}{\tau_3}}\right). \tag{18}$$

Up to this point, the variance of the white noise processes D_i and the time constants of the process model τ_i have been assumed known. To implement the Kalman filter, these parameters must be determined.

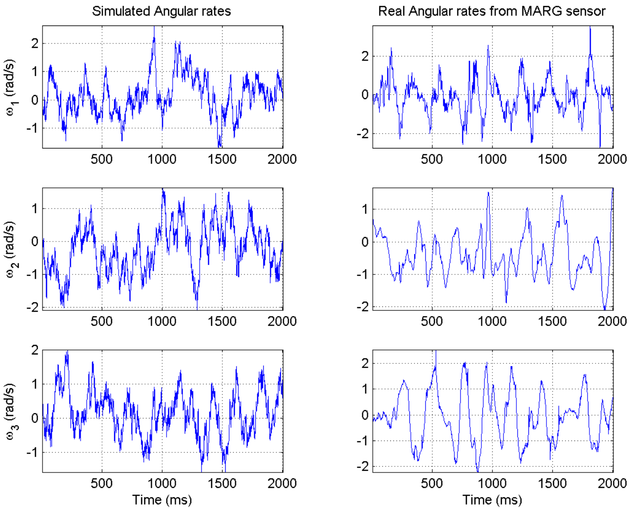

Figure 3. Simulated angular rate (left) and actual angular rate measurements (right).

Using measurement data available from the MARG sensors, the variances and time constants can be found using a simulated process model for the angular rates, where the variance and time constants are adjusted until the output of the simulated model closely matched the real data collected from the MARG sensors. For this purpose, a sensor was attached to the arm of a person and typical arm motion data was collected.

The resultant variances and time constants are shown in Table 1, where ω_1, ω_2 and ω_3 are respectively the angular rates about the *x*, *y*, and *z* body coordinate axes.

Table 1. White noise variance and the time constant of the linear system.

Angular rate	Variance D_i (rad^2/s^2)	Time constant τ_i (s)
ω_1	50	0.5
ω_2	50	0.5
ω_3	50	0.5

Figure 3 shows a comparison between the simulated angular rates and the actual angular rates obtained from a MARG III sensor for typical arm motions. The graphs to the left represent the angular rates generated by the simulation model. The graphs to the right are the angular rates measured by a MARG sensor. It can be observed that the two sets of data exhibit similar characteristics.

The measurement noise covariance matrix R_k represents the level of confidence placed in the accuracy of the measurements, and is given by:

$$R_k = E\left[v(t_k)v(t_k)^T\right]. \tag{19}$$

Assuming that measurements are uncorrelated, Equation (19) leads to the following expression for the measurement noise covariance matrix:

$$R_k = \begin{bmatrix} r_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_{22} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_{33} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_{44} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{55} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_{77} \end{bmatrix}. \tag{20}$$

The diagonal elements are the variances of the individual measurements, which can be determined experimentally using measurement data from the MARG sensors. For this purpose, the measurements from a static MARG sensor were collected. Table 2 summarizes the values derived from experimental measurements.

Table 2. Elements of the measurement noise covariance matrix.

r_{11}	r_{22}	r_{33}	r_{44}	r_{55}	r_{66}	r_{77}
0.01	0.01	0.01	0.0001	0.0001	0.0001	0.0001

Table 3. Convergence of the quaternion estimate.

Sample	$\hat{q}_0$	$\hat{q}_1$	$\hat{q}_2$	$\hat{q}_3$
1	0.99985	0.0082135	0.0066032	0.01357
2	0.99991	0.0057585	0.0049037	0.011901
3	0.9999	0.0055983	0.0048826	0.011882
4	0.9999	0.005288	0.0046884	0.011784
5	0.9999	0.0052297	0.0046353	0.011506

IV. MATLAB IMPLEMENTATION AND TESTNG

After deriving all the required parameters to initialize the Kalman filter, it was implemented using MATLAB to test the performance and accuracy of the quaternion orientation estimates produced by the extended Kalman filter. Real world data recorded using a MARG sensor was used in these tests.

Since the Kalman gain was determined such that the sum of squared errors is minimized, one way to measure the performance of the Kalman filter is through examination of

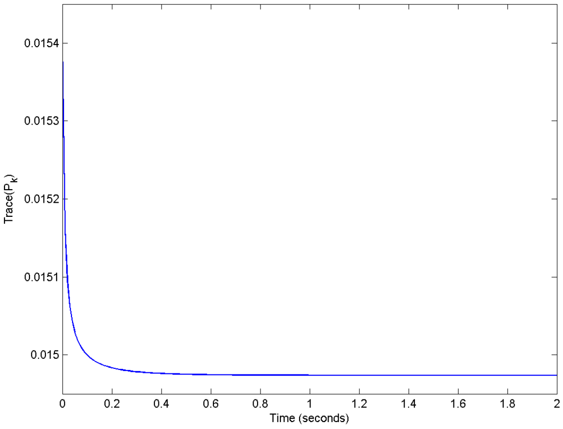

Figure 4. Trace of the error covariance matrix.

the trace of the error covariance matrix P_k. Figure 4 shows the trace of P_k for the first 200 samples of data obtained with the sensor in its reference position (*x* axis pointing north, *y* axis pointing East, and *z* axis point down). It is noted, that the sum of squared errors reaches a steady state after approximately 60 iterations.

Table 3 shows the elements of the quaternion for the first five samples. The initial estimate was chosen to be the unit quaternion *(0.5, 0.5, 0.5, 0.5)*. The actual position of the sensor in the reference position is represented by the quaternion *(1, 0, 0, 0)*. The data shown in Table 3 indicates that the Kalman filter estimate converged to the actual position in a single iteration.

While both the QUEST Algorithm and the Factored Quaternion Algorithm worked well for static orientation and slow movements, the objective of the Kalman filter is to blend angular rate measurements with the estimates produced using magnetometer and accelerometer data during periods in which the sensor module is subjected to motions involving high angular rates and large linear accelerations. To verify the estimation accuracy during such periods, the orientation estimates of the Kalman filter were compared to the estimates produced using only the Factored Quaternion Algorithm with no rate measurement. Two kinds of experiments were conducted for this test. The first used controlled rotations produced by a HAAS precision tilt table. The second used a random motion pattern produced while the sensor was attached to the arm of a person.

In the first set of experiments, the sensor was initially placed at the end of a 1-meter pole attached to the rotating table with its *xyz* axes aligned with West-North-Down directions. The sensor was rotated 90º about the *y*-axis at a rate of 60º/s and then rotated –90º at the same rate (in the reverse direction). Figure 5 shows the performance of the Kalman filter in estimating the orientation of the sensor. The graphs to the left show the orientation estimated by the Factored Quaternion Algorithm, and the graphs to the right show the orientation estimated by the Kalman filter. It can be seen that the Factored Quaternion Algorithm was able to correctly estimate the pitch angle before the first (positive) rotation, between the first and second rotations, and after the second (negative) rotation, but it is not able to correctly estimate orientation during the rotational motions. Large errors in roll and yaw were also produced by the Factored Quaternion Algorithm. On the other hand, it can be seen from the right-center plot that the Kalman filter was able to correctly estimate the pitch angle throughout the duration of the experiment. The small roll and yaw motions seen in the top-right and bottom-right plots are due to misalignment of the individual sensor components within the MARG sensor module.

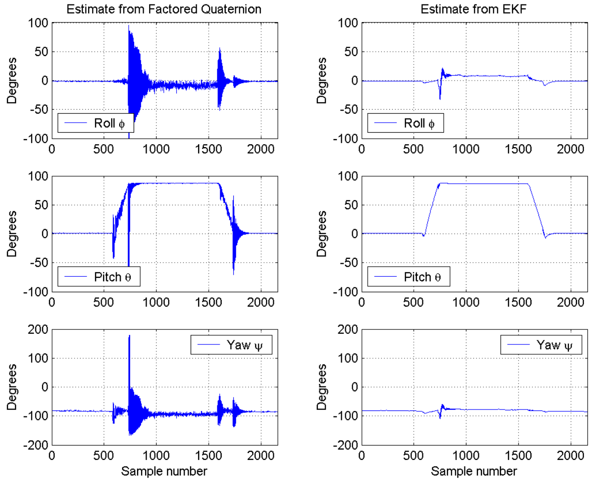

Figure 5. Quaternion estimates produced by the Factored Quaternion Algorithm (left) and Kalman filter (right) with a 90-degree rotation in pitch axis.

Figure 6 shows the results of an experiment in which the sensor was rotated randomly while attached to the arm of a person. Although there is no true reference in this case, it can be seen that the Kalman filter eliminated the jittering and spiking contained in the orientation estimates produced by using the Factored Quaternion Algorithm alone.

V. Real-time Testing results

After initial testing of the extended Kalman filter with the MATLAB implementation, the Factored Quaternion Algorithm and extended Kalman filter algorithm were implemented in Java for real-time testing and evaluation. The real-time quaternion produced by the Kalman filter was visualized by a human-like avatar called "Andy" as seen in Figure 7. Two MARG sensors were used to track the motion of a human arm, one sensor being attached to the upper arm and the other attached to the lower arm.

The Factored Quaternion Algorithm was able to track the motion of the human arm under slow moving conditions where linear acceleration was not significant. However, when the arm motion became faster, the algorithm was not able to follow the arm motion resulting in observable lag as well as overshoots.

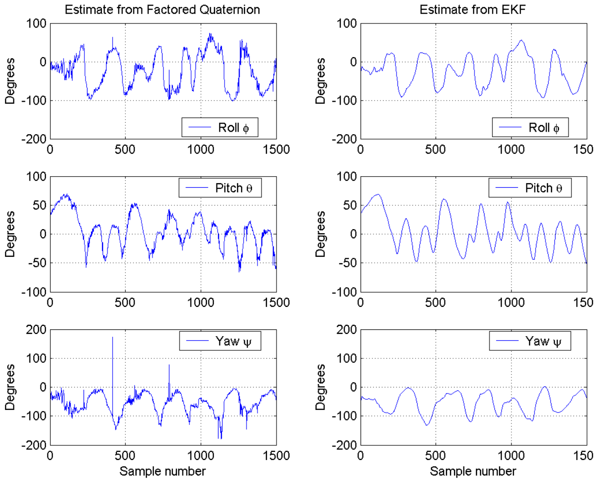

Figure 6. Quaternion estimates produced by the Factored Quaternion Algorithm (left) and Kalman filter (right) with random arm movements.

When the extended Kalman filter was integrated with the Factored Quaternion Algorithm, the avatar was able to successfully track the human arm motion in real time under all conditions. Furthermore, the filtering process did not produce any noticeable lag. Movement of the human arm and the avatar was synchronized.

V. CONCLUSIONS

The paper presents implementation and experimental results for a quaternion-based Kalman filter designed for real-time human body motion tracking using the MARG sensors. A simple process model designed for human body motion tracking was first introduced. The model was then linearized and discretized. Experimental determination of error covariance matrices was described. An extended Kalman filter was implemented, first in MATLAB for offline evaluation and finally in Java for real-time testing and evaluation. The estimated orientation quaternion was visualized using a human avatar. Testing results indicated that the Kalman filter performed satisfactorily for tracking motions of a human arm in real time under all conditions.

ACKNOWLEDGMENT

This research was supported in part by Army Research Office (ARO), Navy Modeling and Simulation Management Office (N6M). Authors would like to thank James Calusdian for his technical support during the course of this project, and to thank Doug McKinney for collaboration on the MARG sensors and Control Interface Unit (CIU).

DISCLAIMER

The views expressed in this paper are solely those of the authors, and do not reflect the opinions of the Naval Postgraduate School, Department of Navy, Department of Defense, or U.S. Government.

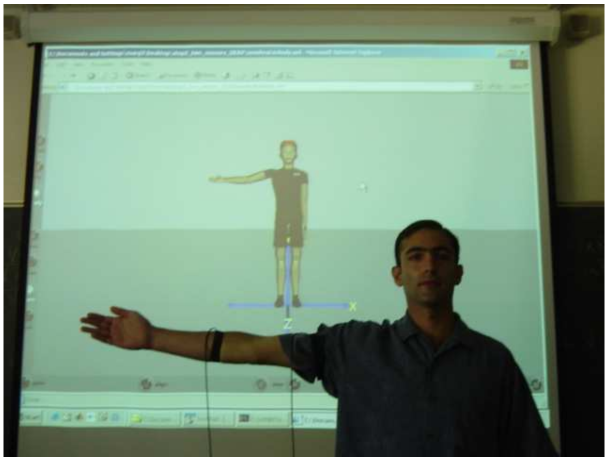

Figure 7. A snapshot of real-time testing.

REFERENCES

[1] Xiaoping Yun, Mariano Lizarraga, Eric R. Bachmann, and Robert B. McGhee, “An Improved Quaternion-Based Kalman Filter for Real-Time Tracking of Rigid Body Orientation,” Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1074-1079, Las Vegas, Nevada, October 2003.

[2] E.R. Bachmann, Xiaoping Yun, D. McKinney, R.B. McGhee, M.J. Zyda, “Design and Implementation of MARG Sensors for 3-DOF Orientation Measurements of Rigid Bodies,” Proceedings of 2003 IEEE International Conference on Robotics and Automation, pp. 1171-1178, Taipei, Taiwan, September 14-19, 2003.

[3] M.D. Shuster and S.D. Oh, “Three-Axis Attitude Determination from Vector Observations,” Journal of Guidance and Control, Vol. 4, No. 1, pp. 70-77, January-February 1981.

[4] Robert B. McGhee, “The Factored Quaternion Algorithm for Orientation Estimation from Measured Earth Gravity and Magnetic Field,” Technical Memorandum, MOVES Institute, Naval Postgraduate School, Monterey, CA, June 2004. http://www.users.muohio.edu/bachmaer/Papers/Factored%20Quaternion.pdf

[5] Conrado Aparicio, “Implementation of a Quaternion-Based Kalman Filter for Human Body Motion Tracking Using MARG Sensors,” Master’s Thesis, Naval Postgraduate School, Monterey, CA, September 2004.

[6] F. Landis Markley and Daniele Mortari, “Quaternion Attitude Estimation Using Vector Observations,” The Journal of the Astronautical Sciences, Vol. 48, No. 2-3, pp. 359-380, 2000.

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

[8] K. Meyer, H. Applewhite, and F. Biocca, “A Survey of Position Trackers,” Presence: Teleoperators and Virtual Environments, Vol. 1, No. 2, pp. 173-200, 1992.

[9] D.M. Gavrila, “The Visual Analysis of Human Movement: A Survey,” Computer Vision and Image Understanding, Vol. 73, No. 1, January 1999, pp. 82-98.

[10] Michael Macedonia, “Games Soldiers Play,” IEEE Spectrum, March 2003, pp. 32-37.