Implementing a Sensor Fusion Algorithm for 3D Orientation Detection with Inertial/Magnetic Sensors

Fatemeh Abyarjoo, Armando Barreto, Jonathan Cofino,
and Francisco R. Ortega

Abstract

In this paper a sensor fusion algorithm is developed and implemented for detecting orientation in three dimensions. Tri-axis MEMS inertial sensors and tri-axis magnetometer outputs are used as input to the fusion system. A Kalman filter is designed to compensate the inertial sensors errors by combining accelerometer and gyroscope data. A tilt compensation unit is designed to calculate the heading of the system.

Keywords

Sensors • 3D • Detection • Algorithms • Filters

Introduction

Orientation tracking has a wide range of applications including military, surgical aid, navigation systems, mobile robots, gaming, virtual reality and gesture recognition [1, 2]. So far, orientation detections are mostly done by using “externally referenced” [3] motion sensing technologies, such as video, radar, infrared or acoustic tracking.

Although these methods achieve good results in an indoor environment, they suffer from some limitations, like shadows, light interruptions, distance limitations and interference [4, 5].

An alternative approach is to use inertial sensors. Inertial sensors detect physical quantities of the moving object regardless of external references, environment lighting or friction. This detected movement is directly related to the object that has the sensors attached. Furthermore, inertial sensors are self-contained technologies, which do not need external devices, like cameras or emitters. These sensors have been used in submarines, spacecraft and aircrafts for many years [6].

Micro-Electro-Mechanical-System (MEMS) based inertial sensors have emerged during the last decade. Due to their miniature size, low power consumption, and light weight [7], the use of inertial MEMS sensors has developed rapidly in recent years.

In this paper, an algorithm is proposed to detect orientation in three dimensions. An inertial measurement unit (IMU) is composed of a tri-axis gyroscope, a tri-axis accelerometer, and a tri-axis magnetometer. A Kalman filter is implemented to yield a reliable estimate of the orientation. Tilt compensation is applied to compensate the tilt error present in the raw measurement.

Data Acquisition

The IMU system we utilized is composed of a tri-axis gyroscope, a tri-axis accelerometer and a tri-axis magnetometer. The sampling rate is 8.96 samples per second. The gyroscope resolution is 16 bits and the sensitivity is 0.007°/s/digit. The accelerometer sensitivity is 0.00024 g/digit. Raw data were acquired while the sensors were stationary on the desk. In Fig. 1, the raw data extracted from sensors are shown.

F. Abyarjoo (✉) • A. Barreto • J. Cofino
Electrical and Computer Engineering Department,,
Florida International University, Miami, FL 33174, USA
e-mail: fabya001@fiu.edu; barretoa@fiu.edu; jcofi001@fiu.edu

F.R. Ortega
School of Computing and Information Science,
Florida International University, Miami, FL 33174, USA
e-mail: forte007@fiu.edu

T. Sobh and K. Elleithy (eds.), *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*, Lecture Notes in Electrical Engineering 313,
DOI 10.1007/978-3-319-06773-5_41, © Springer International Publishing Switzerland 2015

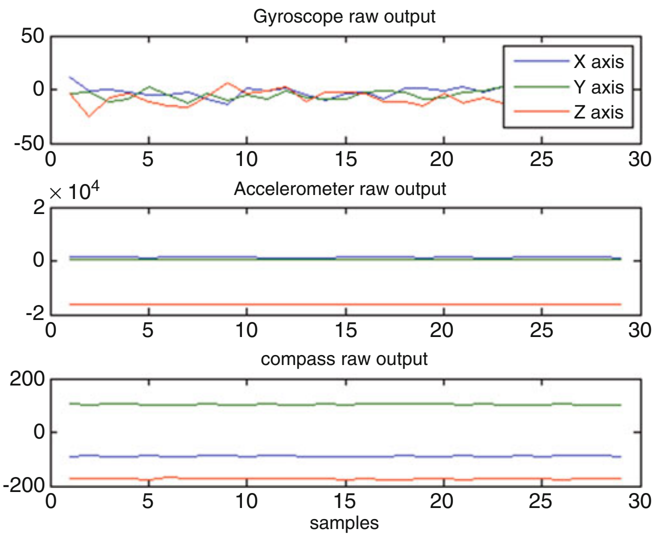

Fig. 1 Raw data

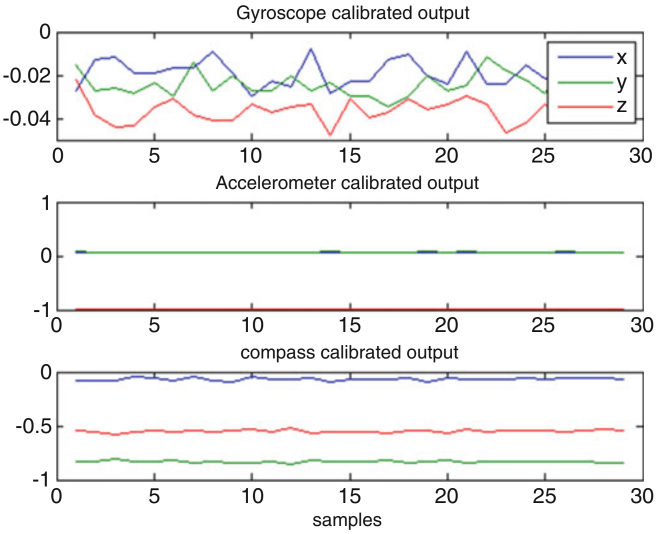

Fig. 2 Calibrated data

The raw data should not be used, as they need to be calibrated. To calibrate these data, scale and bias must be taken into account. The bias represents how far the center of sensor data is from zero. The scale means how much larger the range of data from the sensor is than the real values of the physical quantity.

Figure 2 presents the calibrated data from the gyroscope, the accelerometer and the magnetometer respectively. It can be observed that in the accelerometer calibrated data, X and Y axes are approximately zero and the Z-axis is −1. The axes X and Y are zero because there is no acceleration in these axes. In fact, the only acceleration present is the earth's gravity, which is along the Z-axis pointing downward. This is the reason for measuring a negative number in the Z-axis.

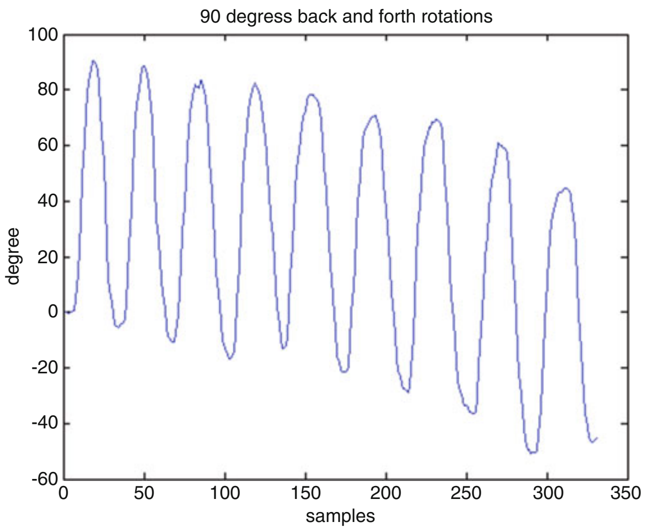

Fig. 3 Drifting rotation angle calculated by the gyroscope integration

The hardware was stationary when the data were recorded and no rotational movement was applied to the system. Therefore, the gyroscope did not record a rate of rotation. The fluctuations which are seen are random noise. This noise is inseparable from the gyroscope data; it will cause drift in the rotational angle, which is obtained based on the gyroscope's data.

Method

MEMS gyroscopes use the Coriolis acceleration effect on a vibrating mass to detect angular rotation. The gyroscope measures the angular velocity, which is proportional to the rate of rotation. They respond quickly and accurately and the rotation can be computed by time-integrating the gyroscope output. Figure 3 depicts the rotational angle, obtained by the trapezoidal integration from the gyroscope signal, for multiple 90° back and forth rotations.

The trapezoidal integration method [8] is shown in Eq. (1), for f(x) between interval a and b.

$$\int_a^b f(x)dx = (b-a)f(a) + \frac{1}{2}(b-a)[f(b)-f(a)] \qquad (1)$$

The computed result drifts over time and after approximately 30 s it drifts down about 50°. The explanation for this phenomenon is that the integration accumulates the noise and offsets over time and turns them into the drift, which yields unacceptable results.

In fact, the integration result is less noisy than the gyroscope signal but there is more drift present. However one good aspect of the gyroscope is that it is not affected by earth's gravity.

Accelerometers measure acceleration based on the forces associated with the Newton's second law. The problem with accelerometers is that they measure both acceleration due to the device's linear movement and acceleration due to earth's gravity, which is pointing toward the earth. Since it cannot distinguish between these two accelerations, there is a need to separate gravity and motion acceleration by filtering. Filtering makes the response sluggish and it is the reason why the accelerometer has to be processed with information from the gyroscope.

By utilizing the accelerometer output, rotation around the X- axis (roll) and around the Y-axis (pitch) can be calculated. If Accel_X, Accel_Y, and Accel_Z are accelerometer measurements in the X-, Y- and Z-axes respectively, Eqs. (2) and (3) show how to calculate the pitch and roll angles:

$$\text{Pitch} = \arctan\left(\frac{\text{Accel\_X}}{(\text{Accel\_X})^2 + (\text{Accel\_Z})^2}\right) \tag{2}$$

$$\text{Roll} = \arctan\left(\frac{\text{Accel\_Y}}{(\text{Accel\_Y})^2 + (\text{Accel\_Z})^2}\right) \tag{3}$$

These equations provide angles in radians and they can be converted to degrees later. Figure 4 presents the rotation angle, which is computed by using the accelerometer signal. Despite recording this signal in a much longer interval, contrary to Fig. 3, no drift is observed in Fig. 4, but it is noisier.

In order to measure rotation around the Z-axis (yaw), the other sensors need to be incorporated with the accelerometer.

It has now been observed that neither the accelerometer nor the gyroscope provides accurate rotation measurements alone. This is the reason to implement a sensor fusion algorithm to compensate for the weakness of each sensor by utilizing other sensors.

System Configuration

The applied sensor fusion system is depicted in Fig. 5. The calibrated accelerometer signal is used to obtain roll* and pitch* by Eqs. (2) and (3). Roll* and pitch* are noisy calculations and the algorithm combines them with the gyroscope signal through a Kalman filter to acquire clean

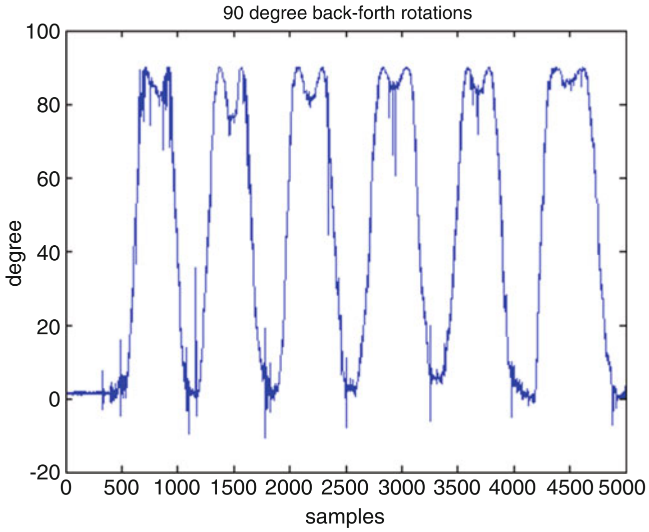

Fig. 4 Noisy rotation angle calculated by the accelerometer

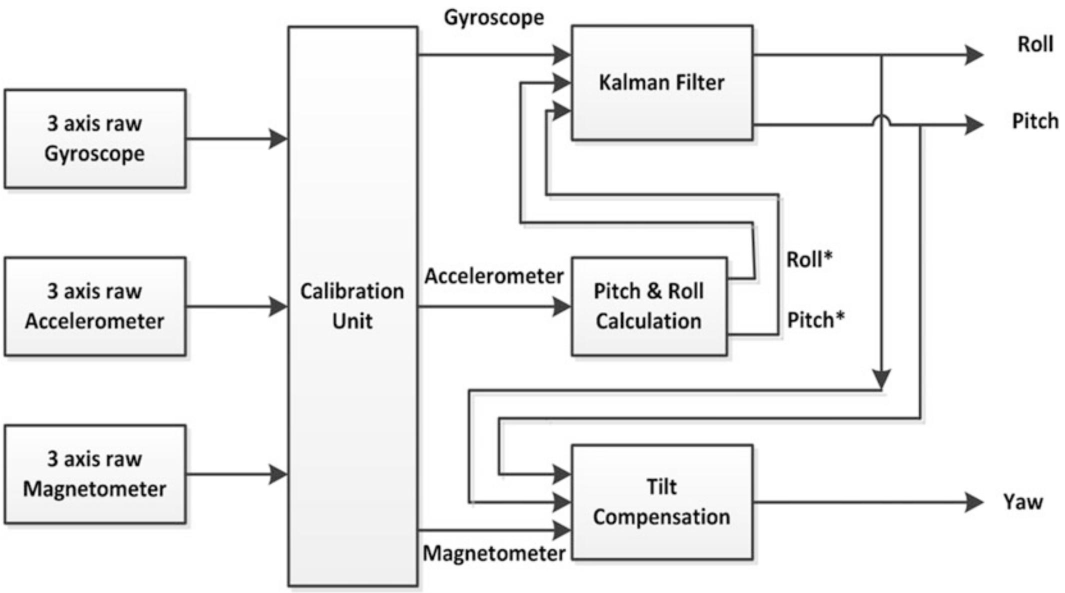

Fig. 5 System structure

and not-drifting roll and pitch angles. On other hand, a tilt compensation unit is implemented, which uses a magnetometer signal in combination with roll and pitch to calculate the challenging yaw rotation.

Kalman Filter

Kalman filtering is a recursive algorithm which is theoretically ideal for fusion processing of noisy data. Implementation of the Kalman filter calls for knowledge of the physical properties of the system. Kalman filter estimates the state of system at a time (t) by using the state of system at time (t−1). The system should be described in a state space form, like the following:

$$x_{k+1} = Ax_k + w_k \tag{4}$$

$$z_k = Hx_k + v_k \tag{5}$$

Where; x_k is the state vector at time k, A is the state transition matrix, w_k is the state transition noise, z_k is measurement of x at time k, H is the observation matrix and v_k is the measurement noise. State variables are the physical quantities of the system like velocity, position, etc.

Matrix A describes how the system changes with time and matrix H represents the relationship between the state variable and the measurement. In our Kalman filter the input vector x, and A and H are:

$$\mathrm{x} = \begin{bmatrix} \omega \\ \varphi \end{bmatrix} \tag{6}$$

$$\mathrm{A} = \begin{bmatrix} 1 & -\Delta \mathrm{t} \\ 0 & 1 \end{bmatrix} \tag{7}$$

$$\mathrm{H} = [1 \quad 0] \tag{8}$$

Where ω is the angular velocity from the gyroscope, and φ is the rotation angle, which is calculated by the accelerometer signal. To implement the Kalman filter, the steps in algorithm 1 should be executed [10]. The A, H, Q and R should be calculated before implementing the filter. Q and R are covariance matrices of w_k and w_k respectively, which are diagonal matrices. Z_k is the system measurement vector and $\hat{x}_k$ is the filter output.

Algorithm 1

1. Set initial values,

$$P_0 = 0, \;\; \hat{x}_0 = 0$$

2. State prediction; the superscript ‘’means predicted value. This step uses the state from the previous time point to estimate the state at the current time point:

$$\hat{x}_k^- = A\hat{x}_{k-1}$$

3. Error covariance prediction; this step uses the error covariance from the previous time point to estimate the error covariance at the current time point:

$$P_k^- = AP_{k-1}A^T + Q$$

4. Kalman gain computation; H and R are computed outside the filter, and P_k^- comes from the previous step. Kalman gain is the weight used for the computation of the estimate and it updates for each time step based on error covariance:

$$K_k = P_k^- H^T \left(HP_k^- H^T + R \right)^{-1}$$

5. Estimate computation; in this step, the algorithm compensates the difference between measurement and prediction. This is the output of the filter:

$$\hat{x}_k = \hat{x}_k^- + K_k H^T \left(z_k - H\hat{x}_k^- \right)$$

6. Error covariance computation; error covariance indicates the degree of estimation accuracy. Larger P_k shows bigger error in estimation:

$$P_k = P_k^- - K_k HP_k^-$$

7. Loop to step 2;

Tilt Compensation

As mentioned earlier, computing the rotation around the Z-axis is challenging (the Z-axis is perpendicular to the earth’s surface). This angle is also called the heading or azimuth. If the gyroscope is used to calculate the heading, not only is the drift problem encountered, but the initial heading must be known [11].

The earth’s magnetic field is parallel to the earth’s surface. Therefore, while the tri-axis magnetometer is parallel with the earth’s surface, it can measure the heading accurately through the direction of the earth’s magnetic field [12]. However, in most applications, the magnetometer is attached to the object and it moves with the object and goes out of the horizontal plane.

By tilting the magnetometer, the direction of axial sensitivity will change [13]. Consequently, it will be difficult to measure the heading. Depending on how much the magnetometer tilts, different amounts of error appear in the calculations.

The tilt compensation process maps the magnetometer data to the horizontal plane and provides the accurate heading calculation regardless of the position of the magnetometer.

The roll and pitch angles are utilized in combination with magnetometer data to correct the tilt error, regardless of the magnetometer's position.

As Fig. 5 shows, the roll and pitch angles come from the output of the Kalman filter.

If m_x, m_y, and m_z are calibrated and normalized magnetometer outputs, and α, β and γ present roll, pitch and yaw respectively, the heading is calculated by Eq. (9). Equations (7) and (8) are used to transform the magnetometer reading to the horizontal plane. When magnetometer data is mapped to the horizontal plane, Eq. (9) obtains a reliable calculation.

$$\mathrm{XH} = \mathrm{m_x}\cos(\beta) + \mathrm{m_y}\sin(\beta)\sin(\alpha) + \mathrm{m_z}\sin(\beta)\cos(\alpha) \tag{7}$$

$$\mathrm{YH} = \mathrm{m_y}\cos(\alpha) + \mathrm{m_z}\sin(\alpha) \tag{8}$$

$$\gamma = \mathrm{atan2}\left(\frac{-\mathrm{YH}}{\mathrm{XH}}\right) \tag{9}$$

The difference between the regular inverse tangent and the MATLAB's command "atan2" is that the first one returns the results in the range of [−π/2, π/2], while "atan2" calculates the results in the range of [−π, π].

Experimental Results

In order to evaluate the performance of the proposed system, some experiments were performed to measure the Euler orientation.

Initially, the hardware was manually moved back and forth in the horizontal plane. It was observed that roll and pitch angles remained constant during this movement. A small fluctuation has been observed in the yaw angles measured, which is because of hand unsteadiness while the hardware was moved. This shows the system can detect even small fluctuations.

The experiment was repeated in both planes, which are perpendicular to the X-axis and perpendicular to the Y-axis as well. Observations proved that the system could track both roll and pitch angles accurately. In both roll and pitch movements, the hand fluctuations can be observed.

The Kalman filter was designed to estimate the orientation. To evaluate the performance of the Kalman filter, an experiment was carried out. Back and forth movements around X-axis were applied to the hardware. The roll angle was obtained by integrating the gyroscope output and then it was compared with the result from the Kalman filter. The results are depicted in Fig. 6.

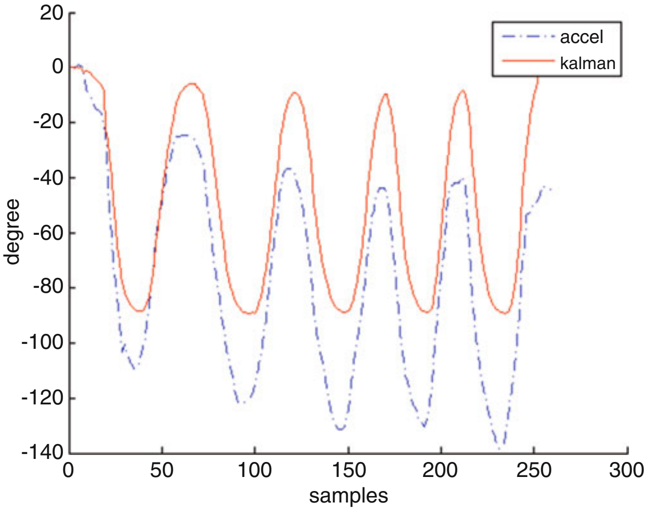

Fig. 6 Comparison between the Kalman filter's output and the gyroscope integration result

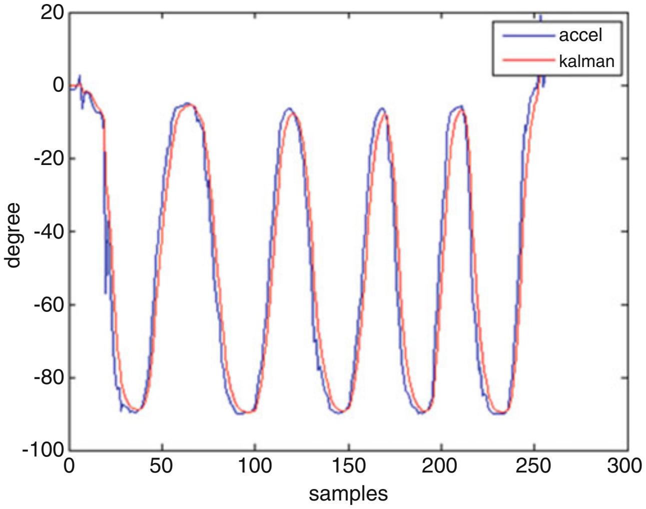

Fig. 7 Comparison between Kalman filter output and the accelerometer result

The dashed line shows the results come from the gyroscope integration by the trapezoidal integration method and the solid line shows the output from the Kalman filter. Noticeable down-drift is clearly seen in the result from integration while this drift is eliminated in the Kalman filter results.

Evaluation of the performance of the Kalman filter continued by comparing the rotation angle from the accelerometer with the Kalman filter output. In Fig. 7, the red solid line presents the filter output and the blue solid line is the angle calculated by the accelerometer output. It is clearly observed that all fluctuations, seen in the accelerometer output, are eliminated successfully by the filter.

Conclusion

In this paper, a method was proposed to detect the orientation in three dimensions by utilizing microelectromechanical sensors.

An efficient algorithm was proposed to deal with the limitation of inertial sensors based on the Kalman filter implementation. Heading compensation is applied to the system to provide accurate orientation around the Z-axis in any position. The experimental results confirmed the appropriate performance of the proposed algorithm.

Our next step will be expanding the algorithm such that it can measure the position in three dimensions.

Acknowledgments This work was sponsored in part by NFS grants HRD-0833093, and CNS-0959985.

References

1. Ji-Hwan Kim, 3-D Hand Motion Tracking and Gesture Recognition Using a Data Glove, IEEE International Symposium on Industrial Electronics, Seol, Korea, 2009
2. Clare Chen, Grace Li, Peter Ngo, Connie Sun, Motion Sensing Technology, Management of Technology – E 103, 2011
3. C.Verplaetes, Inertial Proprioceptive devices: Self-motion-sensing toys and tools, IBM SYSTEMS JOURNAL, VOL 35, NOS 3&4, 1996
4. I-K. Park, J-H. Kim, H-S. Hong, An Implementation of an FPGA-Based Embedded Gesture Recognizer Using a Data Glove, Conference On Ubiquitous Information Management And Communication Processing of the 2nd international conference on Ubiquitous information management and communication 2008, Suwon,Korea, January 31 - February 01, 2008, pp.496-500
5. Fg A. M. Khan, T-S. Kim, Accelerometer Signal-Based Human Activity Recognition using Augmented Autoregressive Model Coefficients and Artificial Neural Nets, " IEEE EMBC 2008, pp. 5172-5175.
6. Oliver J.Woodman, An Introduction to Inertial Navigation", Technical Report, University of Cambridge, 2007
7. Walid Abdel-Hamid, Accuracy Enhancement of Integrated MEMS-IMU/GPS Systems for Land Vehicular Navigation Applications, University of CAGARY, January 2005
8. Thilakshan Kanesalingam, Motion Tracking Glove for Human-Machine Interaction: Inertial Guidance, Mc Master University, Hamilton, Ontario, Canada
9. Doug Vargha, Motion Processing Technology Driving New Innovations in Consumer Products, Invensense
10. Phil Kim, Kalman Filter for Beginners with Matlab Examples", CreateSpace Independent Publishing Platform, 2011
11. Seong Yun, Chan Gook Park, A Calibration Technique for a Two-Axis Magnetic Compass in Telematics Devices, ETRI Journal, Volume 27, Number 3, June 2005
12. M.J. Caruso, Application of Magnetometer Sensors in Navigation Systems, Sensors and Actuators, SAE SP-1220, Feb. 1997, pp. 15-21
13. Adam N.Bingaman, Tilt-Compensated Magnetic Field Sensor, Master Dissertation, Virginia Polytechnic Institute and State University, May 2010
14. H.J. Luinge, P.H. Veltik, Measuring orientation of human body segments using miniature gyroscopes and accelerometer, Medical & Biological Engineering & Computing 2005, Vol.43
15. Rong Zhu, Zhaoying Zhou, A Real-Time Articulated Human Motion Tracking Using Tri-Axis Inertial/Magnetic Sensors Package, IEEE TRANSACTION ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, VOL.12, NO. 2, JUNE 2004