HAL
open science

Accelerometer and Magnetometer Based Gyroscope Emulation on Smart Sensor for a Virtual Reality Application

Baptiste Delporte, Laurent Perroton, Thierry Grandpierre, Jacques Trichet

To cite this version:

Baptiste Delporte, Laurent Perroton, Thierry Grandpierre, Jacques Trichet. Accelerometer and Magnetometer Based Gyroscope Emulation on Smart Sensor for a Virtual Reality Application. Sensors & Transducers., 2012, 14-1 (Special Issue ISSN 1726-5479), p32-p47. hal-00826243

HAL Id: hal-00826243

https://hal.science/hal-00826243v1

Submitted on 27 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sensors & Transducers Journal, Vol.0,Issue 0, Month 2011,pp.

Accelerometer and Magnetometer Based Gyroscope Emulation on Smart Sensor for a Virtual Reality Application

Baptiste Delporte, Laurent Perroton, Thierry Grandpierre and Jacques Trichet[1]
Université Paris Est, ESIEE Engineering, Marne-la-vallée, France
[1]Freescale Semiconductor, Inc. Toulouse, France

Received: /Accepted: /Published:

Abstract: In this paper, we propose two methods based on quaternions to compute the angles of inclination and the angular velocity with 6 degrees of freedom using the measurements of a 3-axis accelerometer and a 3-axis magnetometer. Each method has singularities which occur during the computation of the orientation of the device in the 3-dimensional space. We propose solutions to avoid these singularities. Experimental results are given to compare our model with a real gyroscope.

Keywords: smart sensor; sensor fusion; accelerometer; magnetometer; angular velocity; gyroscope

1. Introduction

The computation of the angles of inclination of a device and its angular velocity has many applications for aeronautics, transportation systems, human motion tracking, games and virtual reality. Classical methods use accelerometers, magnetometers and gyroscopes. For some particular angles, there are singularities for which it is impossible to compute neither the orientation of the device in the 3-dimensional space nor its angular velocity [8, page 407].

Our goal is to design a smart sensor magnetometer based virtual gyroscope, i.e. a method to compute the angular velocity based on the measurements of a 3-axis accelerometer and a 3-axis magnetometer, without any gyroscope, and with 6 degrees of freedom: 3 degrees of freedom are provided by the

accelerometer and the others are provided by the magnetometer. It is easier to implement, less expensive and has lower power consumption than the classical gyroscope solutions. Our target is small motion tracking with embedded devices like cellular phones, with application fields like virtual or augmented reality. Moreover, it is possible to create a virtual gyroscope with a magnetometer and an accelerometer, whereas it is not possible to create a virtual magnetometer or a virtual accelerometer using only a gyroscope. Methods with accelerometers only have been already proposed in [2, 3, 6, 7].

A well-known method to compute a strap down gyroscope output simply consists in differentiating the angles of inclination of the device, but we want to compute the total angular velocity, which is the addition of the angular velocities about the three axes of the fixed frame.

Two methods with two different approaches have been developed. They are proposed in this paper. The method that uses the angles of inclination of the device has been implemented. The method that uses the rotation matrix will be implemented and the two methods will be compared in order to find the method which offers the best precision on the target architecture. This work is a collaboration project between Freescale and ESIEE Engineering School which started in June 2010.

In Section 2, we introduce the platform and the sensors. In Section 3, a first method to compute the angular velocity using the absolute angles of inclination is presented. In Section 4, a second method to compute the angular velocity using the rotation matrix is presented. In Section 5, experimental results are given and a virtual reality application is presented.

2. Hardware and Smart Sensors

We use the new Freescale MMA9550L smart sensor. This motion sensing platform can manage multiple sensor inputs. It includes a 3-axis accelerometer and a ColdFire V1 32-bit microcontroller unit (MCU) with an integrated Multiply and ACcumulate module (MAC module) for DSP-like operations. An additional Honeywell HMC5843 3-axis magnetometer is mounted on the MMA9550L board so that the two sensors are strictly parallel and their frames are aligned.

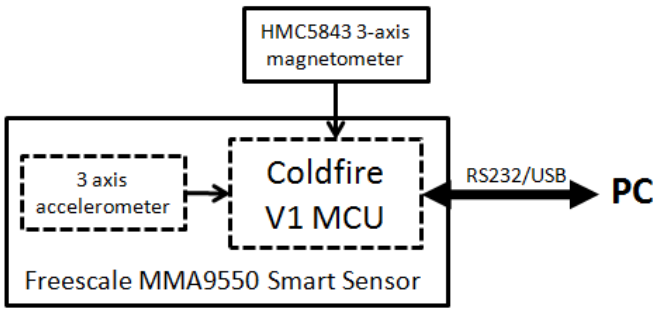

Figure 1 Hardware board

This paper focuses on the mathematical model which provides the angular velocity and the angles of inclination of the device in the 3-dimensional space. The algorithms have been implemented in the form of MATLAB scripts for testing purposes and the curves show the results of these implementations. Once validated, the algorithms have been implemented in C on a Windows 32bits platform for the virtual reality application presented at the last section of the article. In the future, the algorithms will be directly implemented on the MMA9550L, since it includes its own microprocessor.

3. Virtual Gyroscope Based on the Angles of Inclination of the Device

In this section, the angles of inclination and the angular velocity are computed from the accelerometer and the magnetometer measurements using Tait-Bryan angles and quaternions.

3.1 Parameterization of Rotations with Tait-Bryan Angles

In order to describe the orientation of the device in the 3-dimensional space, 2 right-handed Cartesian coordinate systems are used: a fixed reference frame with X_r =North, Y_r =East and Z_r = Down (NED convention), and denoted by the subscript r, and a moving frame attached to a mobile device, denoted by the subscript d. The reference frame and the device frame are aligned when the device is flat and aligned with the X_d axis pointed to magnetic North. Rotation angles are positive when clockwise viewed along the relevant axis vector in the positive direction.

The orientation of the device in the reference frame can be described by Tait-Bryan angles: ϕ, θ and ψ. ψ is the angle of rotation about the Z_r axis (yaw). θ is the angle of rotation about the Y_r axis (pitch). ϕ is the angle of rotation about the X_r axis (roll). Any rotation of the device can be expressed as a composition of these three rotations in the reference frame, as shown in Fig. 2.

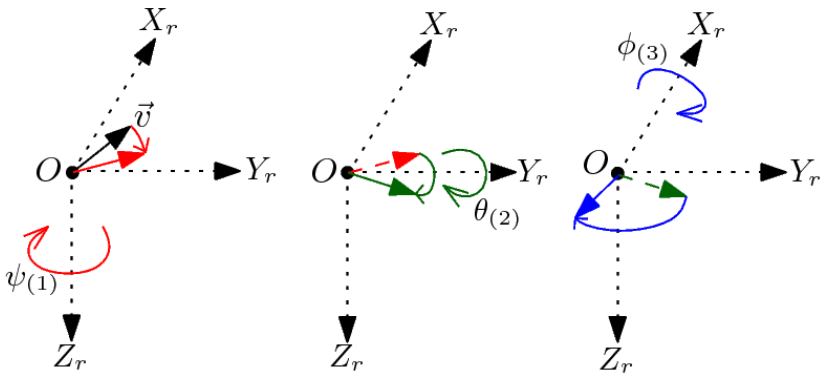

Figure 2. Tait-Briant angles.

A rotation about the Z_r axis, the Y_r axis or the X_r axis can be respectively described by a rotation matrix $R_z(\psi)$, $R_y(\theta)$ or $R_x(\phi)$:

$$R_z(\psi) = \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix}$$

$$R_z(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix}$$

The composition of the 3 rotations about the Z_r axis, then the Y_r axis and finally the X_r axis, is described by the rotation matrix $R(\phi,\theta,\psi) = R_x(\phi).R_y(\theta).R_z(\psi)$.

It is possible to compute ϕ, θ, ψ and the angular velocity $\vec{\omega}$ from the Earth's magnetic field $\overrightarrow{B_d}$, expressed in the device frame, and the Earth's gravitational field $\overrightarrow{g_d}$, expressed in the device frame. The magnetic field is measured by the magnetometer. On the other hand, the accelerometer measures the total acceleration including the gravitational field, the acceleration provided by the user and the acceleration due to the Coriolis force. Consequently, an extraction of the gravitational field $\overrightarrow{g_d}$ needs to be performed with a filter.

The expression of the Earth's magnetic field in the reference frame is given by $\overrightarrow{B_r} = (B \cdot \cos(\delta) \quad 0 \quad B \cdot \sin(\delta))^T$ where B denotes the strength of the magnetic field (in Teslas), δ denotes the angle of inclination of the magnetic field, which depends on the location on the Earth, and $()^T$ denotes the transpose of ().

The expression of the Earth's gravitational field in the reference frame is given by $\overrightarrow{g_r} = (0 \quad 0 \quad g)^T$ where g denotes the strength of the gravitational field, i.e. the acceleration (in Newtons).

The computation process is shown in Fig. 3.

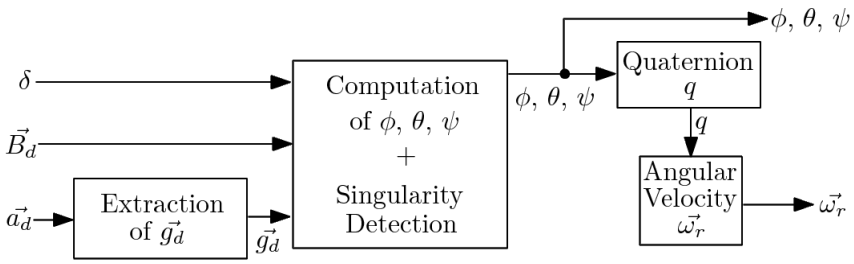

Figure 3. Computation process 1

3.2 Extraction of $\overrightarrow{g_d}$

Since $\overrightarrow{g_d}$ is a constant offset in the measurement of $\overrightarrow{a_d}$, it can be extracted with a low-pass filter. The resulting vector $\overrightarrow{g_d}$ contains sensor medium frequencies and spurious noise. In order to keep only $\overrightarrow{g_d}$, a sliding median filter and a sliding average filter are used, as shown in Fig. 4. The same delay is applied to $\overrightarrow{B_d}$ to make sure they are in phase.

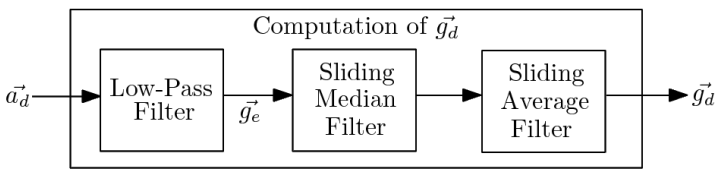

Figure 4. Computation of $\overrightarrow{g_d}$

3.2.1 Low-Pass Filter

The frequency of $\overrightarrow{g_d}$ equals 0. Consequently, the gravitational field can be extracted with a first-order Butterworth low-pass filter. The Z-transform transfer function of the filter is given by:

$$\frac{g_e(z)}{a_d(z)} = \frac{b_0 + b_1.z^{-1}}{1 + a_1.z^{-1}}$$

The default coefficients have been computed with MATLAB by synthesizing a low-pass filter with an experimentally determined cut-off frequency $F_c = 0.02.F_e$, where F_e denotes the sampling frequency. They are given by $[b_0, b_1, a_1] = [0.0305, 0.0305, -0.9391]$. If a variation of the norm $\|\overrightarrow{g_d}\|$ exceeds a threshold, the cut-off frequency of the Butterworth filter increases of $0.05.F_e$ and the coefficients $[b_0, b_1, a_1]$ are computed again. If the cut-off frequency reaches $F_c = 0.4.F_e$, the filter waits for the norm$\|\overrightarrow{g_d}\|$ to stabilize. Then, F_c decreases of $0.05.F_e$ until it reaches $0.02.F_e$. Then, F_c is kept, until the norm $\|\overrightarrow{g_d}\|$exceeds again the threshold. A threshold of $\Delta_{gd} = \frac{1}{100}\|\overrightarrow{g_d}\|$ has been experimentally determined.

3.2.2 Sliding Median Filter

A sliding median filter is used in order to eliminate the highest frequencies sensor spurious noise, which creates variations of the norm of $\overrightarrow{g_d}$. Since this norm should be constant, we need to eliminate the samples that have an erroneous norm. As we will see in section 3.3, $\overrightarrow{g_d}$ directly impacts the accuray of the entire computation process, hence the need to get $\overrightarrow{g_d}$with the least error. The sliding median filter uses a sliding window of n norms. At the beginning, the window contains the first n norms of the first n samples. Then, the norms of the window are sorted. Finally, the median value of the window is extracted, and the sample whose norm is the median value is output from the filter, as shown in Fig. 5.

Let $\vec{g} = (g_x \quad g_y \quad g_z)^T$ be the input vector,$\overrightarrow{g_f} = (g_{fx} \quad g_{fy} \quad g_{fz})^T$ the filtered vector and i the index of the sample. The expression of the filter is given by:

$$\overrightarrow{g_f}(i) = \vec{g}(k)\text{such that } \|\vec{g}(k)\| = median(\|\vec{g}(i-n+1 \dots i)\|)$$

Then, the window slides to the right and the norms of $\vec{g}(i-n+2 \dots i+1)$ are extracted. A sliding median filter creates a delay of$n-1$samples.

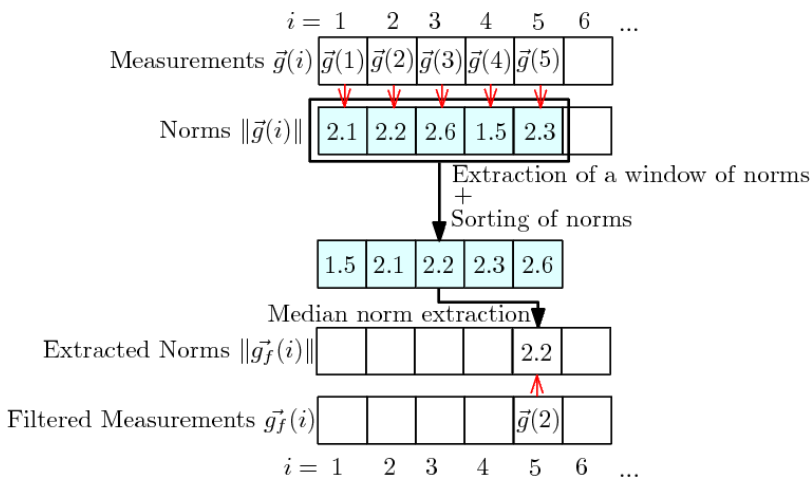

Figure 5. Sliding Median Filter

The gravitational field filtered with the sliding median filter still has variations, a sliding average filter is used to smooth it.

3.2.3 Sliding Average Filter

The sliding average filter uses a sliding window of n samples. At the beginning, the window contains the first $n$samples. Then, the average value of the window is extracted and output from the filter.

Let $\vec{g} = (g_x \quad g_y \quad g_z)^T$ be the input vector,$\overrightarrow{g_f} = (g_{fx} \quad g_{fy} \quad g_{fz})^T$ the filtered vector and $i$the index of the sample. The expression of the filter is given by:

$$\overrightarrow{g_f}(i) = \frac{1}{n}.\sum_{k=i-n+1}^{i} \vec{g}(k)$$

Then, the window slides to the right and filters the values$\vec{g}(i-n+2 \dots i+1)$. A sliding average filter creates a delay of $n-1$samples.

3.3 Computation of the Angles of Inclination

The Earth's magnetic field$\overrightarrow{B_d}$, expressed in the device frame, results from the rotation of the magnetic field $\overrightarrow{B_r}$, expressed in the reference frame.

$$\overrightarrow{B_d} = R_x(\phi).R_y(\theta).R_z(\psi).\overrightarrow{B_r} \tag{1}$$

The Earth's gravitational field ,$\overrightarrow{g_d}$, expressed in the device frame, results from the rotation of the gravitational field $\overrightarrow{g_r}$, expressed in the reference frame. Since $\overrightarrow{g_r}$ remains unchanged after a rotation about the Z_r axis, $R_z(\psi).\overrightarrow{g_r} = \overrightarrow{g_r}$. It follows:

$$\overrightarrow{g_d} = R_x(\phi).R_y(\theta).\overrightarrow{g_r} \tag{2}$$

It is possible to compute the roll angle ϕ from the gravitational field by developing Eq. 2:

$$\phi_g = \text{arctan2}\left(\frac{g_{dy}}{g_{dz}}\right) \tag{3}$$

arctan2 denotes the arctangent on the domain $[-\pi, \pi]$.

Once ϕ is known, it is possible to compute θ:

$$\theta = \arctan\left(\frac{-g_{dx}}{g_{dy}.\sin(\phi)+g_{dz}.\cos(\phi)}\right) \tag{4}$$

arctan denotes the arctangent on the domain $[-\pi/2, \pi/2]$.

If $\overrightarrow{g_d}$ is aligned with the X_d axis, the denominator in Eq. 4 becomes 0. Please see Tab. 1 for

the detection of this singularity.

Once ϕ and θ are known, it is possible to compute ψ by developing Eq. 1:

$$\psi = \arctan2\left(\frac{B_{dz}.\sin(\phi) - B_{dy}.\cos(\phi)}{B_{dx}.\cos(\theta) + B_{dy}.\sin(\phi).\sin(\theta) + B_{dz}.\sin(\theta).\cos(\phi)}\right)$$

3.4 Singularity Detection

A table of the singularities is given in Tab. 1. The normalized gravitational and magnetic field in the device frame are denoted respectively by $\overrightarrow{g_{dN}} = (g_{dNx} \quad g_{dNy} \quad g_{dNz})^T = \frac{\overrightarrow{g_d}}{\|\overrightarrow{g_d}\|}$ and $\overrightarrow{B_{dN}} = (B_{dNx} \quad B_{dNy} \quad B_{dNz})^T = \frac{\overrightarrow{B_d}}{\|\overrightarrow{B_d}\|}$. If a singularity is detected, several compositions of rotations give the same result. Consequently, there are two methods. The first method consists in keeping the previous values of ϕ, θ and ψ. The second method consists in finding the appropriate case that allows the accurate determination of ϕ, θ and ψ.

g_{dNx}	g_{dNy}	B_{dNx}	B_{dNy}	ϕ	θ	ψ
1	0	sin(δ)	cos(δ)	$-\pi/2$ 0 $\pi/2$ π	$-\pi/2$ $-\pi/2$ $-\pi/2$ $-\pi/2$	0 $-\pi/2$ π $\pi/2$
1	0	sin(δ)	-cos(δ)	$-\pi/2$ 0 $\pi/2$ π	$-\pi/2$ $-\pi/2$ $-\pi/2$ $-\pi/2$	0 $\pi/2$ π $-\pi/2$
0	-1	cos(δ)	-sin(δ)	$-\pi/2$ $\pi/2$	0 π	0 π
0	-1	-cos(δ)	-sin(δ)	$-\pi/2$ $\pi/2$	0 π	π 0
-1	0	-sin(δ)	-cos(δ)	$-\pi/2$ 0 $\pi/2$ π	$\pi/2$ $\pi/2$ $\pi/2$ $\pi/2$	0 $\pi/2$ π $-\pi/2$
-1	0	-sin(δ)	-cos(δ)	$-\pi/2$ 0 $\pi/2$ π	$\pi/2$ $\pi/2$ $\pi/2$ $\pi/2$	π $-\pi/2$ 0 $\pi/2$
0	-1	-cos(δ)	sin(δ)	$-\pi/2$ $\pi/2$	π 0	0 π
0	-1	cos(δ)	sin(δ)	$-\pi/2$ $\pi/2$	π 0	0 π

Table 1. Table of Singularities

3.5 Parameterization of Rotations with Quaternions

The quaternions are hypercomplex numbers, i.e. 4-dimensional mathematical objects, used to describe rotations in the 3-dimensional space [5].

3.5.1 Definition and Properties of a Quaternion

A quaternion q has 4 coordinates in a 4-dimensional vector space and is denoted by $q = (q_1 \quad q_2 \quad q_3 \quad q_4)^T$ with $(q_1 \quad q_2 \quad q_3 \quad q_4) \in \mathbb{R}^4$. It consists of a vector part $q_v = (q_1 \quad q_2 \quad q_3)^T$ and a scalar part $q_s = q_4$. It can be expressed in the following form:

$$q = q_1 \cdot i + q_2 \cdot j + q_3 \cdot k + q_4 \tag{5}$$

In Eq. 5, i, j and k are imaginary numbers: $i^2 = j^2 = k^2 = -1$, and $i \cdot j = -j \cdot i = k$, $j \cdot k = -k \cdot j = i$, $k \cdot i = -i \cdot k = j$. Therefore, it is possible to compute the product of two quaternions $q = (q_1 \quad q_2 \quad q_3 \quad q_4)^T$ and $q' = (q'_1 \quad q'_2 \quad q'_3 \quad q'_4)^T$, denoted by $q \cdot q'$, using the properties of the hypercomplex numbers. It can be noticed that the product between 2 quaternions is not commutative: $q \cdot q' \neq q' \cdot q$.

The inverse of a quaternion $q = (q_1 \quad q_2 \quad q_3 \quad q_4)^T$ is denoted by $q^{-1} = (-q_1 \quad -q_2 \quad -q_3 \quad q_4)^T$.

3.5.2 Euler-Rodrigues Parameters

A quaternion $q = (q_1 \quad q_2 \quad q_3 \quad q_4)^T$ can be used to describe a rotation by an angle α about a unit vector $\vec{a} = (a \quad b \quad c)^T$ that is the axis. $\vec{a}$ is a unit vector, so $\|\vec{a}\| = 1$. The Euler-Rodrigues parameters corresponding to the rotation are given by:

$$q_1 = a \cdot \sin\left(\frac{\alpha}{2}\right) \quad q_2 = a \cdot \sin\left(\frac{\alpha}{2}\right) \quad q_3 = a \cdot \sin\left(\frac{\alpha}{2}\right) \quad q_4 = \cos\left(\frac{\alpha}{2}\right)$$

3.5.3 Rotation

Let $\vec{v} = (x \quad y \quad z)^T$ be a vector. The quaternion q transforms $\vec{v}$ into another vector $\overrightarrow{v_f} = (x_f \quad y_f \quad z_f)^T$ by rotating it by angle α about an $\vec{a}$ axis. A fourth null coordinate is added to $\vec{v}$, so it becomes $\overrightarrow{v_q} = (x \quad y \quad z \quad 0)^T$. The rotated vector $\overrightarrow{v_f}$ corresponds to the vector part of $\overrightarrow{v_{fq}}$ given by:

$$\overrightarrow{v_{fq}} = q \cdot \overrightarrow{v_q} \cdot q^{-1} \tag{6}$$

The scalar part of $\overrightarrow{v_{fq}}$ is 0, since $\overrightarrow{v_{fq}}$ is a pure vector in the 3-dimensional space.

3.5.4 Composition of Two Rotations

Let q_α be a quaternion describing a rotation by an angle α about an axis $\vec{a}$ and q_β a quaternion describing a rotation by an angle β about an axis $\vec{b}$. The composition of the rotations about the $\vec{b}$axis, then the $\vec{a}$ axis, is given by the quaternion $q_{\alpha,\beta} = q_\alpha \cdot q_\beta$. Let $\vec{v} = (x \quad y \quad z)^T$ be a vector. The quaternion $q_{\alpha,\beta}$ transforms $\vec{v}$ into another vector $\overrightarrow{v_f} = (x_f \quad y_f \quad z_f)^T$ by rotating it by angle β about an axis $\vec{b}$, then by angle α about an axis $\vec{a}$. With $\overrightarrow{v_q} = (x \quad y \quad z \quad 0)^T$, the expression of Eq. 6 becomes $\overrightarrow{v_{fq}} = \left(q_\alpha \cdot q_\beta\right) \cdot \overrightarrow{v_q} \cdot \left(q_\beta{}^{-1} \cdot q_\alpha{}^{-1}\right)$. The rotated vector $\overrightarrow{v_f}$ corresponds to the vector part of $\overrightarrow{v_{fq}}$. The scalar part of $\overrightarrow{v_{fq}}$ is 0, since $\overrightarrow{v_{fq}}$ is a pure vector in the 3-dimensional space.

3.5.5 Computation of the Angular Velocity

The instantaneous angular velocity $\overrightarrow{\omega_r}(t)$ of the device at the instant t, expressed in the reference frame, corresponds to the vector part of $\overrightarrow{\omega_{rq}}(t)$ given by [4]:

$$\overrightarrow{\omega_{rq}}(t) = 2 \cdot q^{-1}(t) \cdot \frac{dq(t)}{dt}$$

The scalar part of $\overrightarrow{\omega_{rq}}(t)$ is 0, since $\overrightarrow{\omega_{rq}}(t)$ is a pure vector in the 3-dimensional space, which finally gives:

$$\overrightarrow{\omega_r}(t) = \begin{pmatrix} \cos(\theta) \cdot \cos(\psi) \cdot \dot{\phi} + \sin(\psi) \cdot \dot{\theta} \\ -\cos(\theta) \cdot \sin(\psi) \cdot \dot{\phi} + \cos(\psi) \cdot \dot{\theta} \\ \sin(\theta) \cdot \dot{\phi} + \dot{\psi} \end{pmatrix}$$

3.6 Computation of the Quaternion q From the Angles of Inclination

A rotation by angle ψ about the Z_r axis, by angle θ about the Y_r axis or by angle ϕ about the X_r axis can be respectively described by the quaternion $q_\psi = \left(0 \quad 0 \quad \sin\left(\frac{\psi}{2}\right) \quad \cos\left(\frac{\psi}{2}\right)\right)^T$, $q_\theta = \left(0 \quad \sin\left(\frac{\theta}{2}\right) \quad 0 \quad \cos\left(\frac{\theta}{2}\right)\right)^T$, or $q_\phi = \left(\sin\left(\frac{\phi}{2}\right) \quad 0 \quad 0 \quad \cos\left(\frac{\phi}{2}\right)\right)^T$. The quaternion describing the composition of the rotations about the Z_r axis, then the Y_r axis, and finally the X_r axis, is given by $q = q_\phi \cdot q_\theta \cdot q_\psi$.

The method described above has 8 singularities. Consequently, the computation of the angles ϕ, θ and ψ cannot be accurate if the detection of singularities is not efficient enough.

4 Virtual Gyroscope Based on the Rotation Matrix

In this section, the angles of inclination of the device and the angular velocity are computed from the accelerometer and the magnetometer measurements using the rotation matrix and

quaternions. Although its computation cost is higher, the major advantage of this method is that it reduces the number of singularities to only 2. Furthermore, this method does not require the explicit computation of the angles. The computation process is shown in Fig. 6.

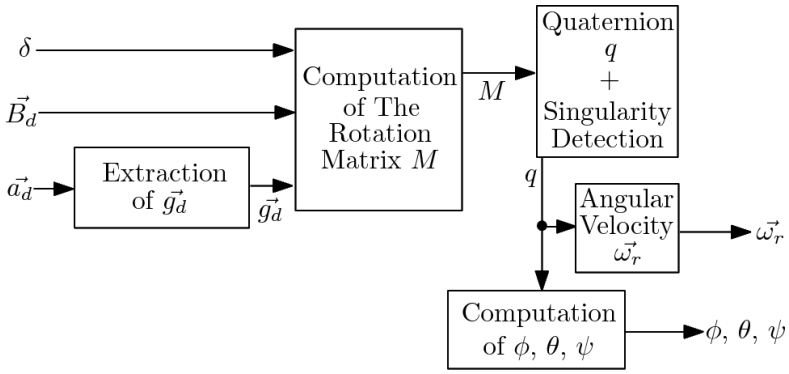

Figure 6. Computation Process 2

4.1 Computation of the Rotation Matrix M

Let $\vec{v} = (x \quad y \quad z)^T$ be a vector. The rotation matrix M transforms $\vec{v}$ into another vector $\overrightarrow{v_f} = (x_f \quad y_f \quad z_f)^T$ by rotating it by an unknown angle α about an unknown axis $\vec{a}$. The coordinates of the resulting $\overrightarrow{v_f}$ are given by:

$$\overrightarrow{v_f} = M \cdot \vec{v} \tag{7}$$

Once $\vec{v}$ and $\overrightarrow{v_f}$ are known, it is possible to compute M. Consequently, we will be able to deduce α and $\vec{a}$.

Let $\overrightarrow{g_{dN}} = \frac{\overrightarrow{g_d}}{\|\overrightarrow{g_d}\|}$ be the normalized gravitational field in the device frame, $\overrightarrow{B_{dN}} = \frac{\overrightarrow{B_d}}{\|\overrightarrow{B_d}\|}$ the normalized magnetic field in the device frame, $\overrightarrow{C_d} = \overrightarrow{g_d} \times \overrightarrow{B_d}$ the cross product between the gravitational field and the magnetic field in the device frame, $\overrightarrow{C_{dN}} = \frac{\overrightarrow{C_d}}{\|\overrightarrow{C_d}\|}$, $\overrightarrow{T_{Bd}} = \overrightarrow{B_{dN}} \times \overrightarrow{C_{dN}}$ and $\overrightarrow{T_{gd}} = \overrightarrow{g_{dN}} \times \overrightarrow{C_{dN}}$.

On the other hand, let $\overrightarrow{g_{rN}} = \frac{\overrightarrow{g_r}}{\|\overrightarrow{g_r}\|}$ be the normalized gravitational field in the reference frame, $\overrightarrow{B_{rN}} = \frac{\overrightarrow{B_r}}{\|\overrightarrow{B_r}\|}$ the normalized magnetic field in the reference frame, $\overrightarrow{C_r} = \overrightarrow{g_r} \times \overrightarrow{B_r}$ the cross product between the gravitational field and the magnetic field in the reference frame, $\overrightarrow{C_{rN}} = \frac{\overrightarrow{C_r}}{\|\overrightarrow{C_r}\|}$, $\overrightarrow{T_{Br}} = \overrightarrow{B_{rN}} \times \overrightarrow{C_{rN}}$ and $\overrightarrow{T_{gr}} = \overrightarrow{g_{rN}} \times \overrightarrow{C_{rN}}$.

The expressions of $\overrightarrow{g_{rN}}$ and $\overrightarrow{B_{rN}}$ are respectively given by $\overrightarrow{g_{rN}} = (0 \quad 0 \quad 1)^T$ and $\overrightarrow{B_{rN}} = (\cos(\delta) \quad 0 \quad \sin(\delta))^T$. Consequently, $\overrightarrow{C_r} = (0 \quad g \cdot B \cdot \cos(\delta) \quad 0)^T$, $\overrightarrow{C_{rN}} = (0 \quad 1 \quad 0)^T$, $\overrightarrow{T_{Br}} = (-\sin(\delta) \quad 0 \quad \cos(\delta))^T$ and $\overrightarrow{T_{gr}} = (-1 \quad 0 \quad 0)^T$.

The general expression of the matrix M is given by:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

Since $\overrightarrow{T_{gd}} = M \cdot \overrightarrow{T_{gr}}$, $\overrightarrow{C_{dN}} = M \cdot \overrightarrow{C_{rN}}$ and $\overrightarrow{g_{dN}} = M \cdot \overrightarrow{g_{rN}}$, the matrix M can be deduced from $\overrightarrow{T_{gd}}$, $\overrightarrow{C_{dN}}$ and $\overrightarrow{g_{dN}}$:

$$M = \begin{bmatrix} -T_{gdx} & C_{dNx} & g_{dNx} \\ -T_{gdy} & C_{dNy} & g_{dNy} \\ -T_{gdz} & C_{dNz} & g_{dNz} \end{bmatrix}$$

Since $\overrightarrow{T_{Bd}} = M \cdot \overrightarrow{T_{Br}}$ and $\overrightarrow{B_{dN}} = M \cdot \overrightarrow{B_{rN}}$, there is another method, which is to deduce the matrix M from $\overrightarrow{T_{Bd}}$, $\overrightarrow{C_{dN}}$ and $\overrightarrow{B_{dN}}$:

$$M = \begin{bmatrix} B_{dNx} \cdot \cos(\delta) - T_{Bdx} \cdot \sin(\delta) & C_{dNx} & B_{dNx} \cdot \sin(\delta) + T_{Bdx} \cdot \cos(\delta) \\ B_{dNy} \cdot \cos(\delta) - T_{Bdy} \cdot \sin(\delta) & C_{dNy} & B_{dNy} \cdot \sin(\delta) + T_{Bdy} \cdot \cos(\delta) \\ B_{dNz} \cdot \cos(\delta) - T_{Bdz} \cdot \sin(\delta) & C_{dNz} & B_{dNz} \cdot \sin(\delta) + T_{Bdz} \cdot \cos(\delta) \end{bmatrix}$$

4.2 Computation of the Quaternion q

Once the matrix M is known, it becomes possible to compute q. The four possible cases are given in Tab. 2 [1, page 15]. The comparison of M_{11}, M_{22} and M_{33} gives the appropriate case that allows the computation of q .

$M_{22} < -M_{33}$ $M_{11} > M_{22}$ $M_{11} > M_{33}$	$q = \frac{1}{2}.\begin{pmatrix} \sqrt{1 + M_{11} - M_{22} - M_{33}} \\ (M_{12} + M_{21})/\sqrt{1 + M_{11} - M_{22} - M_{33}} \\ (M_{31} + M_{13})/\sqrt{1 + M_{11} - M_{22} - M_{33}} \\ (M_{23} - M_{32})/\sqrt{1 + M_{11} - M_{22} - M_{33}} \end{pmatrix}$
$M_{22} > M_{33}$ $M_{11} < -M_{22}$ $M_{11} < -M_{33}$	$q = \frac{1}{2}.\begin{pmatrix} (M_{12} + M_{21})/\sqrt{1 - M_{11} + M_{22} - M_{33}} \\ \sqrt{1 - M_{11} - M_{22} - M_{33}} \\ (M_{23} + M_{32})/\sqrt{1 + M_{11} - M_{22} - M_{33}} \\ (M_{31} - M_{13})/\sqrt{1 + M_{11} - M_{22} - M_{33}} \end{pmatrix}$

$M_{22} < M_{33}$ $M_{11} < -M_{22}$ $M_{11} > M_{33}$	$q = \frac{1}{2}.\begin{pmatrix} (M_{31}+M_{13})/\sqrt{1+M_{11}-M_{22}-M_{33}} \\ (M_{23}+M_{32})/\sqrt{1+M_{11}-M_{22}-M_{33}} \\ \sqrt{1-M_{11}-M_{22}+M_{33}} \\ (M_{12}-M_{21})/\sqrt{1+M_{11}-M_{22}-M_{33}} \end{pmatrix}$
$M_{22} > -M_{33}$ $M_{11} > -M_{22}$ $M_{11} > -M_{33}$	$q = \frac{1}{2}.\begin{pmatrix} (M_{23}-M_{32})/\sqrt{1+M_{11}-M_{22}-M_{33}} \\ (M_{31}-M_{13})/\sqrt{1+M_{11}-M_{22}-M_{33}} \\ (M_{12}-M_{21})/\sqrt{1+M_{11}-M_{22}-M_{33}} \\ \sqrt{1+M_{11}+M_{22}+M_{33}} \end{pmatrix}$

Table 2. Computation of q From M

4.3 Computation of the Angles of Inclination

Once q is known, ϕ, θ and ψ can be computed:

$$\phi = \text{arctan2}\left(\frac{2\cdot(q_1\cdot q_4 + q_2\cdot q_3)}{1-2\cdot({q_1}^2+{q_2}^2)}\right)$$

$$\theta = \arcsin\left(2\cdot(q_2\cdot q_4 - q_3\cdot q_1)\right)$$

$$\psi = \text{arctan2}\left(\frac{2\cdot(q_3\cdot q_4 + q_1\cdot q_2)}{1-2\cdot({q_2}^2+{q_3}^2)}\right)$$

With this method, there are only 2 singularities left: $\theta = \pm\pi/2$. They are known as the gimbal lock. If such a singularity is detected, the previous value of θ is kept.

5 Experimental Results

5.1 Extraction of $\overrightarrow{\mathbf{g_d}}$

The Fig. 7 shows the results of the computation of $\overrightarrow{g_d}$. First, the X_d axis has been aligned with the Z_r axis, then the device has been shaken by the user, who created accelerations of about 2 g . Then, the Y_d axis has been aligned with the Z_r axis and, finally, the Z_d axis has been aligned with the Z_r axis.

We can notice that the norm of the extracted gravitational field equals 1 g and that the highest frequencies due to the fast shakes of the user have been eliminated.

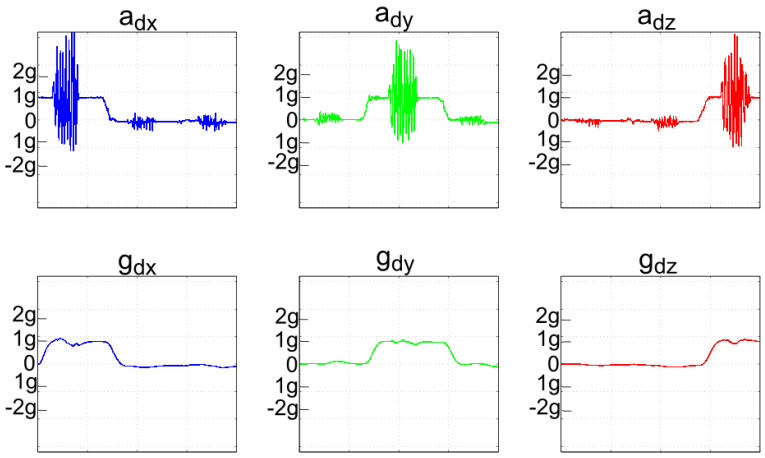

Figure 7. Extraction of $\overrightarrow{g_d}$

5.2 Angular Velocity

Experimental results of the angular velocity computed with our first method virtual gyroscope (top) compared to the one from a real gyroscope (bottom) are given in Fig. 8. The real gyroscope is tied to the accelerometer and the magnetometer and their frames are aligned to get a 9 degree of freedom system. The similarity of the two measures confirms the accuracy of our model.

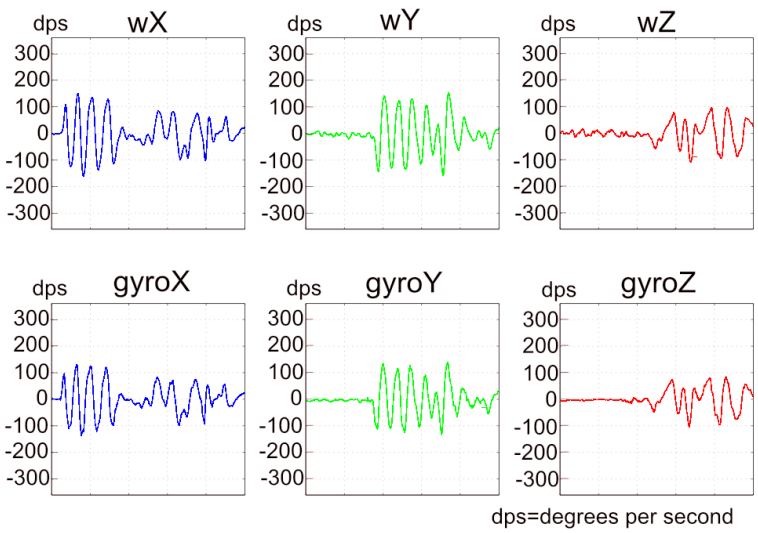

Figure 8. Angular Velocity Computed With our Virtual Gyroscope (top) vs. a Real one (bottom)

5.3 Virtual Reality Application

In order to evaluate the precision, the latency and finally the usability of our smart sensor we have developed a basic virtual reality application. This application has been developed using

OGRE 3D (Object Oriented Graphics Rendering Engine) [9] which is an open source library toolkit. It allows building quickly virtual environment where you can move your view through the control of the camera position. The 3D scene has to be created previously with a classical modeler tool such as 3DS Studio Max or Blender for example. In our implementation, the smart sensor values are used to modify the orientation of the camera or any virtual object of a simulated environment. Hence it replaces the keyboard/mouse. The raw data of the 6 sensors (3 for magnetometer, 3 for accelerometer) are collected by the Freescale MMA9550L hardware (presented in section 2) which forwards these data on a serial over USB connection to the host PC without any processing. Then we implement the equations of section 4 in a processing thread that continuously takes these raw data as input, computes and produces the 3 gyroscope angles as output. Finally these 3 computed values are continuously sent to the OGRE application as shown on figure 9.

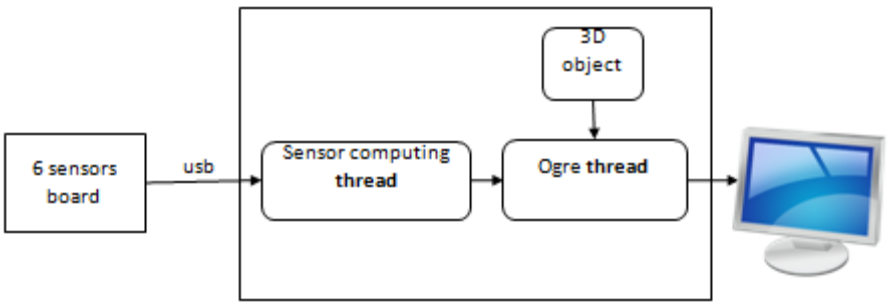

Figure 9. Software architecture

On the photo of Figure 10, we can see the sensor board behind the 3D scene computed and displayed in real-time. Moving the board on the 3 axis makes the virtual object (here a model of a plane) moving the same way. This experiment proves the usability of our virtual gyroscope algorithm. Moreover, angles computation can be executed in real-time leading to an almost instantaneous interaction with the virtual environment.

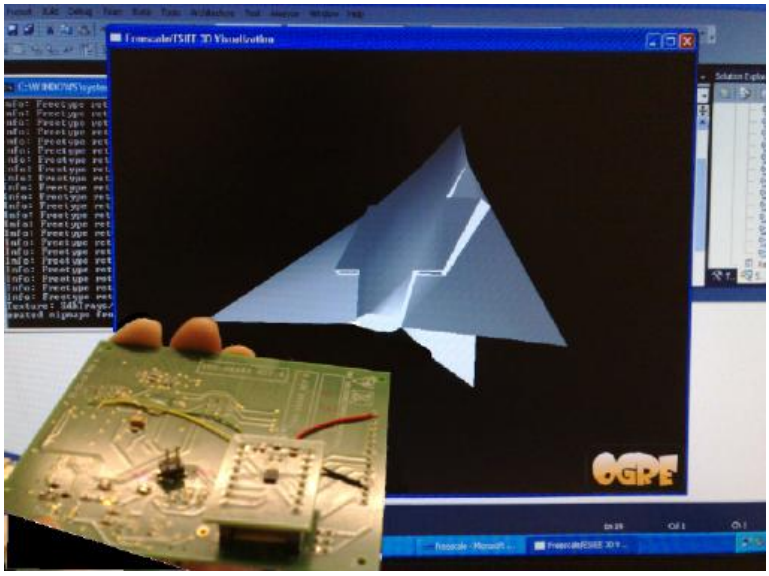

Figure 10. Software architecture

6 Conclusion and Future Works

In this paper, we have presented two methods to implement a virtual gyroscope that only uses the measurements of an accelerometer and a magnetometer, with 6 degrees of freedom.

The two methods have their own advantages and drawbacks. The method which uses the angles of inclination is easier to implement, but there are 8 singularities, which need to be solved. Moreover, the computation of ψ depends on the computation of θ, which in turns depends on ϕ. If there is a singularity on ϕ, the computation of the angles is not possible. On the other hand, the method with the rotation matrix has only two singularities but its computation cost is higher. The second method has not been completely implemented and validated yet; this is our current work.

The precision of both methods and their limitations must be investigated and will be our main future work.

Finally, we plan to optimize the implementation of both methods on the MMA9550L. This will allow us to provide the angular velocity and the angles of inclination of the device and use them for several applications, like a 3-dimensional mouse, a virtual joystick, a human motion tracker. The MMA9550L board can communicate with the PC with a Bluetooth connection. Consequently, the board can become a portable device with its own power supply.

Acknowledgment

The authors would like to thank Freescale for their support, the platform, and Mr. Mark Pedley whose work is the base of this project.

References

[1] J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Technical report, Stanford University, California 94301-9010, october 2006.

[2] T. Liu, G.-R. Zhao, and S. Pan. New calculating method of angular velocity in gyroscope-free strapdown inertial navigation systems. Systems Engineering and Electronics, 32(1):162–165, January 2010.

[3] P. Schopp, L. Klingbeil, C. Peters, A. Buhmann, and Y. Manoli. Sensor fusion algorithm and calibration for a gyroscope-free imu. In Proceedings of the Eurosensors 23rd Conference, volume 1, pages 1323–1326, 2009.

[4] A. L. Schwab. Quaternions, finite rotation and Euler parameters, 2002.

[5] D. Stahlke. Quaternions in classical mechanics, 2007.

[6] C. Wang, J.-X. Dong, S.-H. Y., and X.-W. Kong. Hybrid algorithm for angular velocity calculation in a gyroscope-free strapdown inertial navigation system. Journal of Chinese Inertial Technology, 18(4):401–404, 2010.

[7] X.-N. Wang, S.-Z. Wang, and H.-B. Zhu. Study on models of gyroscope-free strap-down inertial navigation system. Binggong Xuebao/Acta Armamentarii, 27(2):288–292, 2006.

[8] J. R. Wertz. Spacecraft Attitude Determination and Control. D. Reidel Publishing Company, Dordrecht, Holland, 1978.

[9] http://www.ogre3d.org