國立交通大學

資訊學院資訊科技（IT）產業研發碩士班

碩士論文

用於物體追蹤的加速度計星群

G-Sensor Constellations for Object Tracking

研究生：蔡文添

指導教授：黃俊龍　教授

易志偉　教授

中華民國九十八年六月

用於物體追蹤的加速度計星群
G-Sensor Constellations for Object Tracking

研究生：蔡文添　　Student：Wen-Tien Tsai

指導教授：黃俊龍　　Advisor：Jiun-Long Huang

易志偉　　Chih-Wei Yi

國立交通大學
資訊學院資訊科技（IT）產業研發碩士班
碩士論文

A Thesis
Submitted to College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Industrial Technology R & D Master Program on
Computer Science and Engineering

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

用於物體追蹤的加速度計星群

學生：蔡文添　　　　　　　　　　指導教授：黃俊龍
易志偉

國立交通大學資訊學院產業研發碩士班

摘　　要

在多數的慣性追蹤系統中，物體旋轉角度通常都是由陀螺儀或是磁力計所測量得來，在這篇論文中，我們提出一個新的物體移動軌跡追蹤系統，稱作加速度計星群，主要是純用多顆加速度計來做物體的追蹤，並不需要使用陀螺儀或是磁力計來做角度的測量。我們所提出的加速度星群技術除了可以用在導航的輔助外、車禍的分析或是新的人機介面也都是我們可以應用的範圍。

G-Sensor Constellations for Object Tracking

Student：Wen-Tien Tsai

Advisors：Dr. Jiun-Long Huang
Dr. Chih-Wei Yi

Industrial Technology R & D Master Program of
Computer Science College
National Chiao Tung University

ABSTRACT

In most inertial tracking systems, motion directions are measured by gyroscopes and/or magnetometers. In this work, we propose a new tracking system, called g-sensor constellations, that is composed of only accelerometers. Gyroscopes and magnetometers are not necessary to detect the change of motion directions. The proposed technique can be used to improve the accuracy of dead reckoning systems, help the analyzing of traffic accidents, and develop new human-computer interfaces. A g-sensor constellation that is composed of three accelerometers is built to verify the proposed technique.

誌謝

首先我要感謝我的家人跟朋友對我的支持與照顧，由於他們在我背後的支持，才能讓我在研究及學習上無後顧之憂，希望他們也能跟我一同分享完成這篇論文的喜悅。再來我要感謝的是我的指導教授黃俊龍教授以及共同指導易志偉教授，在研究學習過程中，給予我的指導和協助並且分享他們的經驗，使我能夠順利的完成論文，還有我要感謝實驗室學長姐、同學及學弟，經過與他們不斷的討論以及適時的鼓勵及兩年來的陪伴，讓我在研究學習之餘，能得到壓力的紓解及不斷往前的動力，最後我要再一次感謝所有人，讓我在研究所的這兩年得到許多美好的回憶。

Contents

1 Introduction 1

2 Preliminaries-Rotation Matrix 5

2.1 Unit Quaternion Representation of Rotation 5

2.2 Rotation Matrix by a Unit Vector 8

2.3 Rotation Matrix by Yaw Pitch Roll(ZYX) 9

3 Related Works 11

3.1 Accelerometer + Gyroscope 11

3.2 Accelerometer + Magnetometer 12

3.3 Accelerometer + Gyroscope + Magnetometer 13

3.4 IMU + Kalman filter 14

3.5 Six Linear Accelerometers 15

4 Sensor Calibration 17

4.1 Single Sensor Calibration 17

4.1.1 Linear Calibration 17

4.1.2 Least Square Calibration 18

4.2 Multiple Sensors Calibration 19

5 G-Sensor Constellations 20

5.1 Framework 20

5.2 The Object Tracking Algorithm 23

5.2.1 Extrapolation 24

5.2.2 Least Square Method 24

5.2.3 Adjust Velocity 26

5.2.4 Object Rotation 27

6 Experiments **29**

6.1 Static Experiment 29

6.2 Static Rotation Experiment 32

6.3 Dynamic Experiment in a Straight Line 35

6.4 Dynamic Experiment in a S-curve 36

7 Conclusions **37**

List of Figures

2.1 Rotation about an arbitrary unit vector $\mathbf{u}(u_x, u_y, u_z)$ by an angle θ. 8

2.2 Yaw-Pitch-Roll. 9

2.3 Rotation by X, Y, Z axis. 10

3.1 A simplified strapped down system [From "Simplified strapped down inertial navigation utilizing bang-bang gyro torquing"]. 12

3.2 Six linear accelerometer system [From "Gyroscope free Strapdown inertial measurement unit by six linear accelerometers"]. 15

4.1 Alignment between sensors. 19

5.1 Constellations with four g-sensors. 20

5.2 Constellations with five g-sensors. 21

5.3 A g-sensor constellation. 21

5.4 System framework of the g-sensor constellation. 22

5.5 Object tracking by extrapolation and least square method. 23

5.6 Flowchart of the object tracking algorithm. 28

6.1 The vibratility of sensor 1 in a static state. 30

6.2 The vibratility of sensor 2 in a static state. 30

6.3 The vibratility of sensor 3 in a static state. 31

6.4 Displacement and angle with different thresholds. 32

6.5 Rotation along time. 33

6.6 Rotate 30°by Z-axis with average method. 33

6.7 Rotate 30° by Z-axis with threshold method. 33

6.8 Rotate 60° by Z-axis with average method. 34

6.9 Rotate 60° by Z-axis with threshold method. 34

6.10 Rotate 90° by Z-axis with average method. 34

6.11 Rotate 90° by Z-axis with threshold method. 34

6.12 The track of a striaght line movement by average method. 35

6.13 The track of a striaght line movement by threshold method. 35

6.14 The track of a S-curve movement by average method. 36

6.15 The track of a S-curve movement by threshold method. 36

List of Tables

6.1 The drifting distance of a static constellation after 30 seconds. 31

6.2 The Yaw-Pitch-Roll angle by rotating 30°, 60° and 90°. 32

6.3 The distance measured after the constellation moves 45m in a straigt line. 35

Chapter 1

Introduction

The Micro-Electro-Mechanical System (MEMS) is a popular technique which allows both electronic circuits and mechanical devices to be manufactured on a silicon chip. MEMS promises to revolutionize nearly every product category by bringing together silicon-based microelectronics and makes possible the realization of complete systems-on-a-chip. Due to MEMS technique is not only more and more practice but also it has advantages of low cost and small size. New applications are emerging as the existing technology is applied to the miniaturization and integration of conventional devices. Examples of MEMS device applications include gyroscopes, accelerometers, press sensors, DLP and wireless RF sensors. In recently years it is even applied on bioengineering, technique of memory and game platforms.

MEMS-based inertial measurement units (IMUs) sensors are now becoming pervasive. Inertial measurement units are usually composed of an accelerometer, a gyroscope, and a magnetometer. The accelerometer can measure the acceleration, the gyroscope can measure the rotation speed, and the magnetometer can give the orientation. Through accelerometers the distance of motion can be got by integrating acceleration, the rotation and orientation of motion also can get through gyroscopes or magnetometers.

Inertial measurement units (IMUs) are widely used in various applications. In [1], a triaxial accelerometer was used to monitor human movements and postures. The technique can be used for tracking patients in hospitals or detecting the falls of elders. In [2], IMUs were used for detecting personal activity, and then small-scale location information was revealed by comparing with a pre-built knowledge database. That is useful to improve the accuracy of GPSs. In [3], accelerometer-based estimators for pedestrian step lengths were developed. However, positions and motion trajectory were not studied. In [4], [5], and [6], authors used IMUs to implement pedometers to calculate step lengths, walking distances and velocities, but there was a gyroscope in their IMUs to detect motion directions. In [7], a position tracing system for human movements are developed. In the system, accelerometers and magnetometers were used, but the accuracy was dropping as metals or electronic devices around. Besides these, IMUs or inertial sensors still have a lot of other applications, including

- Electronic products: On game controllers it can get the player's movement or motion of hand for controlling operation of games. It also can apply for pedometers by gathering information from IMUs.

- Vehicular applications: Car alarms can use it to detect open of a car door and turn on an alarm when a door is opened. Besides on rollover protection or airbag activation it can be detected by IMUs and adopt some protections when a car occurs an accident.

- Geographic applications: On motion tracking the accelerometer can be used to calculate the distance of an object and the gyroscope or magnetometer can be used to calculate the rotation or orientation of an object. On earthquake monitoring is also useful for using IMUs.

- Others: some notebooks will integrate 3D g-sensor for HDD protection, it can automatic lock the HDD for protecting the data loss when g-sensor detects the fall of notebooks.

Although IMUs can be used on a lot scopes, most and popular applications are used on navigation system, called Inertial Navigation Systems. These INSs compose of accelerometers and gyroscopes and/or magnetometers. Inertial Navigation Systems combine with GPS for position and attitude determination are widely. They provide many complimentary characteristics that overcome the limitations experienced when using them individually. The integration of GPS and INS is always achieved using a Kalman filter for example [8] [9] [10]. An inertial navigation system includes at least a computer and a platform or module containing accelerometers, gyroscopes, or other motion-sensing devices. They are in general very expensive because they demand precision requirements, high computational efforts, and last but not least sufficient reliability under harsh environmental conditions.

Most IMUs are composed of accelerometers, gyroscopes, and magnetometers. In the past, IMU-based object tracking techniques need gyroscopes and/or magnetometers to detect the change in motion directions or simply the motion directions. However, the price of IMUs is relatively high for some daily applications. In addition, magnetometers are interfered by other electronic devices.

In this work, we propose a pure accelerometer-based solution, called g-sensor constellations. A g-sensor constellation is a collection of accelerometers that has a rigid geometric topology and can be used for object tracking in the 3D space. Due to no need for the help of magnetometers, the interference caused by electronic devices is avoided and gyro-free which is cheaper than gyro-based IMUs. We apply extrapolation methods and least square methods (LSM) to track the locations of sensors. First, we apply the extrapolation method to get an estimation of sensor positions, and then the topology information of the constellation is utilized to calibrate positions by the LSM. Later, the LSM outcomes are applied to correct accumulation error of velocities and used in the next iteration. Besides the displacement, the rotation information is also obtained.

Whether accelerometers have high accuracy, the acceleration value of three axes still has slight vibration even sensors at static state. This small and slight value will increase error of displacement or velocity and become bigger over the time when we use integration to get this information. In this work a method can reduce the error by using the characteristic of fixed graph and LSM. The g-sensor constellation system can use the rigid geometric topology to calculate the distance and the rotation of an object with extrapolation, LSM and transformation matrix based on coordination and vector methods. The error can be corrected when an object is static or moving dynamic. The motion of straight line or curved line even 3D also can be tracked.

The rest of this thesis is organized as follows. Chapter 2 will describe the related work in recently years including how they to do, what differences with our work. In chapter 3 some preliminary proofs will also describe here. Chapter 4 the main approach will be described including the g-sensor constellation architecture, sensors alignment and the object tracking algorithm. The extrapolation and LSM will be described in this chapter. The experimental condition and devices will be introduced in chapter 5. The result of experiment also will be described. Chapter 6 is our conclusion for g-sensor constellations system and what performance we achieve what problems we observe.

Chapter 2

Preliminaries-Rotation Matrix

In this chapter three rotation matrixes will be described after we use them. In first section we will show how a unit quaternion is used to represent a rotation of a vector in 3D space. In second section rotation in 3 dimensional space by a unit vector will be introduced. Finally we will describe a matrix expression for any frame given its Euler angles. Using the $Z - Y - X$ convention, a matrix can be constructed that transforms every vector of the given reference frame in the corresponding vector of the referred frame. Before we start to introduce those rotation methods, some rules will be describes. The lowercase letters (ex. $\mathbf{u}$) are marked bold to represent a vector and the uppercase letters (ex. M, T) are marked italic to represent the matrix. Other lowercase letters represent a variable or an element.

2.1 Unit Quaternion Representation of Rotation

Quaternions were first defined by W.R. Hamilton in the 1800's. Quaternions can be used in describing rotations in 3 dimensional real vector space. First we will define properties and operations of quaternions, and then show the relationships with the rotation matrix.

Using complex number notation, a quaternion q can be represented by the following :

$$\widehat{q} = q_0 + q_x + q_y + q_z$$

where q, q_x, q_y and q_z are real numbers. The basic of the imaginary components i, j and

k have the following properties :

$$i^2 = -1, j^2 = -1, k^2 = -1$$
$$ij = k, jk = i, ki = j$$
$$ji = -k, kj = -i, ik = -j.$$

If $\widehat{r} = r_0 + ir_x + jr_y + kr_z$, quaternion multiplication becomes

$$\begin{aligned}\widehat{r}\widehat{q} &= (r_0q_0 - r_xq_x - r_yq_y - r_zq_z) + i(r_0q_x + r_xq_0 + r_yq_z - r_zq_y)\\ &+ j(r_0q_y - r_xq_z + r_yq_0 + r_zq_x) + k(r_0q_z + r_xq_y - r_yq_x + r_zq_0).\end{aligned}$$

It can be easily verified that the multiplication operation is not commutative.

The conjugate of q has the same real part with negated imaginary part , $\widehat{q}^* = q_0 - iq_x - jq_y - kq_z$. The conjugate of a product of two quaternions $(\widehat{q}\widehat{r})^*$ is $\widehat{r}^*\widehat{q}^*$. The dot product of two quaternions is the sum of products of corresponding components.

$$\widehat{p} \cdot \widehat{q} = \widehat{p}_0\widehat{q}_0 + \widehat{p}_x\widehat{q}_x + \widehat{p}_y\widehat{q}_y + \widehat{p}_z\widehat{q}_z = \frac{(\widehat{p}\widehat{q}^* + \widehat{q}\widehat{p}^*)}{2}$$

It is obvious that the dot product is a real number. The square of magnitude of a quaternion is the dot product of the quaternion with itself.

$$|\widehat{q}|^2 = \widehat{q} \cdot \widehat{q} = \widehat{q}\widehat{q}^* = \widehat{q}^*\widehat{q}$$

A multiplicative inverse of a nonzero quaternion is

$$\widehat{q}^{-1} = (\frac{1}{\widehat{q} \cdot \widehat{q}})\widehat{q}^*$$

A vector in 3D space $\mathbf{t} = (t_x\ t_y\ t_z)^T$ can be represented by a purely imaginary quaternion

$$\widehat{t} = 0 + it_x + jt_y + kt_z.$$

We will denote $\mathbf{t} = \widehat{t}$, if the corresponding components are the same between a 3D vector and a pure imaginary quaternion. Let s be vector in 3D space, and $\widehat{s}$ be its pure imaginary representation; then we can verify the following properties, which will be used in the following proofs.

$$\mathbf{s} \cdot \mathbf{t} = \widehat{s} \cdot \widehat{t} = \frac{(\widehat{s}\widehat{t}^* + \widehat{t}\widehat{s}^*)}{2}$$
$$\mathbf{s} \times \mathbf{t} = \frac{(\widehat{s}\widehat{t} - \widehat{t}\widehat{s})}{2}$$

We will the claim that $\widehat{t'} = \widehat{r}\widehat{t}\widehat{r}^*$ is a rotation operation when $\widehat{t}$ is a pure imaginary quaternion equivalent to a 3D vector $\mathbf{t}$, and $\widehat{r}$ is a unit quaternion. The inverse mapping is $\widehat{r}^*\widehat{t'}\widehat{r}$ and the mapping is bijective. First it will be shown that $\widehat{t'}$ is pure imaginary by checking if $\widehat{t'} + \widehat{t'}^* = 0$.

$$\begin{aligned}
\widehat{t'} + \widehat{t'}^* &= \widehat{r}\widehat{t}\widehat{r}^* + (\widehat{r}\widehat{t}\widehat{r}^*)^* = \widehat{r}\widehat{t}\widehat{r}^* + \widehat{r}\widehat{t}^*\widehat{r}^* \\
&= \widehat{r}(\widehat{t} + \widehat{t}^*)\widehat{r}^* = 0
\end{aligned}$$

from the fact that t is pure imaginary.

The rotation operation must preserve the dot product.

$$\begin{aligned}
s' \cdot t' &= \frac{(\widehat{s'}\widehat{t'}^* + \widehat{t'}\widehat{s'}^*)}{2} \\
&= \frac{\left[\widehat{r}\widehat{s}\widehat{r}^*(\widehat{r}\widehat{t}\widehat{r}^*)^* + \widehat{r}\widehat{t}\widehat{r}^*(\widehat{r}\widehat{s}\widehat{r}^*)^*\right]}{2} \\
&= \frac{\left[\widehat{r}\widehat{s}(\widehat{r}^*\widehat{r})\widehat{t}\widehat{r}^* + \widehat{r}\widehat{t}(\widehat{r}^*\widehat{r})\widehat{s}\widehat{r}^*\right]}{2} \\
&= (\frac{\widehat{r}\widehat{s}\widehat{t}^*\widehat{r}^* + \widehat{r}\widehat{t}\widehat{s}^*\widehat{r}^*}{2}) = \widehat{r}(\frac{\widehat{s}\widehat{t}^* + \widehat{t}\widehat{s}^*}{2})\widehat{r}^* \\
&= \widehat{r}(\widehat{s} \cdot \widehat{t})\widehat{r}^* = (\widehat{s} \cdot \widehat{t})(\widehat{r}\widehat{r}^*) = (\widehat{s} \cdot \widehat{t}) = \mathbf{s} \cdot \mathbf{t}
\end{aligned}$$

Here we use the fact the dot product $\widehat{s} \cdot \widehat{t}$ is a real number, and the product of a real number with quaternion is commutative. It is a special case of the above that $\widehat{t} \cdot \widehat{t}$ is the same as $\mathbf{t} \cdot \mathbf{t}$, which means the length of a vector is invariant after the rotation.

If t is taken from t rotated by an angle θ around the $\mathbf{u} = (u_x, u_y, u_z)$ axis is a unit vector, then we represent a unit quaternion $\widehat{r} = r_0 + ir_x + jr_y + kr_z$ in the following form

$$\cos(\frac{\theta}{2}) + \sin(\frac{\theta}{2})\mathbf{u}$$

,where $r_0 = \cos(\frac{\theta}{2})$, $\begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T = \sin(\frac{\theta}{2}) \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T$ then $\widehat{r}\widehat{t}\widehat{r}^*$ reduce to following equation. A component by component equation for $\widehat{t'} = \widehat{r}\widehat{t}\widehat{r}^*$ with matrix notation becomes

$$\begin{bmatrix} t'_x \\ t'_y \\ t'_z \end{bmatrix} = \begin{bmatrix} (r_0^2 + r_x^2 - r_y^2 - r_z^2) & (-2r_0 r_z + 2r_x r_y) & (2r_0 r_y + 2r_x r_z) \\ (2r_0 r_x + 2r_x r_y) & (r_0^2 - r_x^2 + r_y^2 - r_z^2) & (-2r_0 r_x + 2r_y r_z) \\ (-2r_0 r_y + 2r_x r_z) & (2r_0 r_x + 2r_y r_z) & (r_0^2 - r_x^2 - r_y^2 + r_z^2) \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.1}$$

The more detail description can obtain from [11].

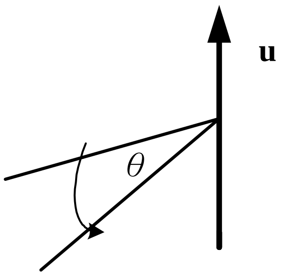


Figure 2.1: Rotation about an arbitrary unit vector $\mathbf{u}(u_x, u_y, u_z)$ by an angle θ.

2.2 Rotation Matrix by a Unit Vector

Let a unit vector $\mathbf{u} = (u_x, u_y, u_z)$, where $u_x^2 + u_y^2 + u_z^2 = 1$, the transformation matrix M for a counterclockwise rotation by an angle of θ about an axis the direction of $\mathbf{u}$ showed as Figure 2.1. We can get $r_0 = \cos(\frac{\theta}{2})$, $\begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T = \sin(\frac{\theta}{2}) \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T$ and substitute to Eq. 2.1, which will use following proof :

$$\sin^2\frac{\theta}{2} = \frac{1-\cos\theta}{2}, \cos^2\frac{\theta}{2} = \frac{1+\cos\theta}{2}$$

$$\sin\theta = 2\sin\frac{\theta}{2}\cos\frac{\theta}{2}$$

$$\begin{bmatrix} (r_0^2 + r_x^2 - r_y^2 - r_z^2) & (-2r_0r_z + 2r_xr_y) & (2r_0r_y + 2r_xr_z) \\ (2r_0r_x + 2r_xr_y) & (r_0^2 - r_x^2 + r_y^2 - r_z^2) & (-2r_0r_x + 2r_yr_z) \\ (-2r_0r_y + 2r_xr_z) & (2r_0r_x + 2r_yr_z) & (r_0^2 - r_x^2 - r_y^2 + r_z^2) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1+\cos\theta}{2} + (\frac{1-\cos\theta}{2})(2u_x^2 - 1) & (-\sin\theta)u_z + 2(\frac{1-\cos\theta}{2})u_xu_y & \sin\theta u_y + 2(\frac{1-\cos\theta}{2})u_xu_z \\ \sin\theta u_x + 2(\frac{1-\cos\theta}{2})u_xu_y & \frac{1+\cos\theta}{2} + (\frac{1-\cos\theta}{2})(2u_y^2 - 1) & (-\sin\theta)u_x + 2(\frac{1-\cos\theta}{2})u_yu_z \\ (-\sin\theta)u_y + 2(\frac{1-\cos\theta}{2})u_xu_z & \sin\theta u_x + 2(\frac{1-\cos\theta}{2})u_yu_z & \frac{1+\cos\theta}{2} + (\frac{1-\cos\theta}{2})(2u_z^2 - 1) \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta + (1-\cos\theta)u_x^2 & (-\sin\theta)u_z + (1-\cos\theta)u_xu_y & \sin\theta u_y + (1-\cos\theta)u_xu_z \\ \sin\theta u_x + (1-\cos\theta)u_xu_y & \cos\theta + (1-\cos\theta)u_y^2 & (-\sin\theta)u_x + (1-\cos\theta)u_yu_z \\ (-\sin\theta)u_y + (1-\cos\theta)u_xu_z & \sin\theta u_x + (1-\cos\theta)u_yu_z & \cos\theta + (1-\cos\theta)u_z^2 \end{bmatrix}$$

It can also be proofed from [12].

Theorem 1 *Rotation about an arbitrary unit vector* $\mathbf{u}\,(u_x, u_y, u_z)$ *by an angle* θ *is given by a matrix*

$$M(\mathbf{u},\theta) = \begin{bmatrix} c+(1-c)u_x^2 & (1-c)u_xu_y-(s)u_z & (1-c)u_xu_z+(s)u_y \\ (1-c)u_yy_x+(s)u_z & c+(1-c)u_y^2 & (1-c)u_yu_z-(s)u_x \\ (1-c)u_zu_x-(s)u_y & (1-c)u_zu_y+(s)u_x & c+(1-c)u_z^2 \end{bmatrix} \tag{2.2}$$

,where $c = \cos\theta$, $s = \sin\theta$

2.3 Rotation Matrix by Yaw Pitch Roll(ZYX)

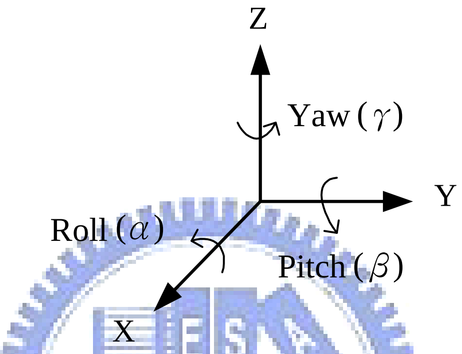


Figure 2.2: Yaw-Pitch-Roll.

As Figure 2.2 we define the Euler angles for our system. Yaw, pitch and roll are counterclockwise by Z, Y and X axis.

Considering three basic rotation matrices are in three dimensions. These matrices represent counterclockwise rotations of an object relative to fixed coordinate axes, by an angle of α, β, γ around the X, Y and Z axes, respectively. The direction of the rotation is determined by the right-hand rule: Rx rotates the Y-axis towards the Z-axis, Ry rotates the Z-axis towards the X-axis, and Rz rotates the X-axis towards the Y-axis. The transpose of the above matrices represents positive (right-hand sense) rotation of the coordinate axes relative to a fixed object. As Figure 2.3 (a) (b) (c) we can get the Yaw-Pitch-Roll $(Z-Y-Z)$ transform matrix. From Eq. 2.2, the matrix of $R_x(\alpha)$ describes rotation about an arbitrary unit vector $(1,0,0)$ by an angle α. The matrix of $R_y(\beta)$ describes rotation about an arbitrary unit vector $(0,1,0)$ by an angle β and $R_z(\gamma)$ describes rotation about an arbitrary unit vector $(0,0,1)$ by an angle γ. Then we can get

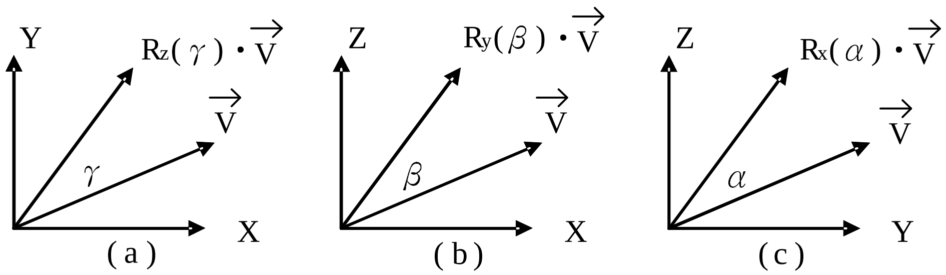


Figure 2.3: Rotation by X, Y, Z axis.

the following equations (those equations can also be proofed in [13]):

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix},$$

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

The product $T = R_x(\alpha)\cdot R_y(\beta)\cdot R_z(\gamma)$ represents a rotation matrix whose Yaw, Pitch, and Roll are γ, β, and α (using the $Z-Y-X$ convention for Euler angles).

Theorem 2 *Rotation by Yaw-Pitch-Roll($Z-Y-X$ axes)*

$$T = \begin{bmatrix} c_1c_2 & -c_2s_1 & s_2 \\ c_3s_1 + c_1s_2s_3 & c_1c_3 - s_1s_2s_3 & -c_2s_3 \\ s_1s_3 - c_1c_3s_2 & c_3s_1s_2 + c_1s_3 & c_2c_3 \end{bmatrix}$$

,where $c_1 = \cos\gamma, c_2 = \cos\beta, c_3 = \cos\alpha$ and $s_1 = \sin\gamma, s_2 = \sin\beta, s_3 = \sin\alpha$

Chapter 3

Related Works

In this chapter, we will introduce some tracking systems with IMUs. Related works are about tracking or motion by accelerometers, gyroscopes and magnetometer. Finally the six linear accelerometer system will be introduced. It describes a tracking system which composes of six linear accelerometers without any gyroscope or magnetometer. It designs one accelerometer at the center of face of a cube and the sensing axis of each accelerometer is along the respective cube face diagonal.

The research for tracking, motion and navigation system is currently carried out in many laboratories with using of accelerometers, gyroscopes and magnetometers. This is the basic idea for tracking and navigation by using accelerometers, gyroscopes and magnetometers. They usually use accelerometers to get acceleration of objects and integrate the acceleration to calculate the moving distance. Gyroscopes and magnetometers are usually used to calculate the rotation or change of orientation.

3.1 Accelerometer + Gyroscope

In [14], it represented a simplified strapped down system for navigation. A strapped down navigation system comprised of gyroscopes and accelerometers without magnetometers. They could be mounted on body or vehicle. Gyroscopes provided attitude angle and angular rate signals along three axes. Accelerometers meant provide signals representative of the acceleration along three independent axes. First transformation matrix connected to the attitude angle output of the gyroscopes to accelerometers transforms the gyroscope and the accelerometer signals from body coordinates to gyroscope coordinates. Second transformation matrix connected to the output of the gyros, transforms the gyro coordinates into navigation coordinates. The algorithm flow is shown as Figure 3.1. It is a simplified and basic system for tracking or motion. Because of gyroscopes had expensive price and error increased with double integration by the time. Attitude determination systems that used inexpensive sensors had been discussed.

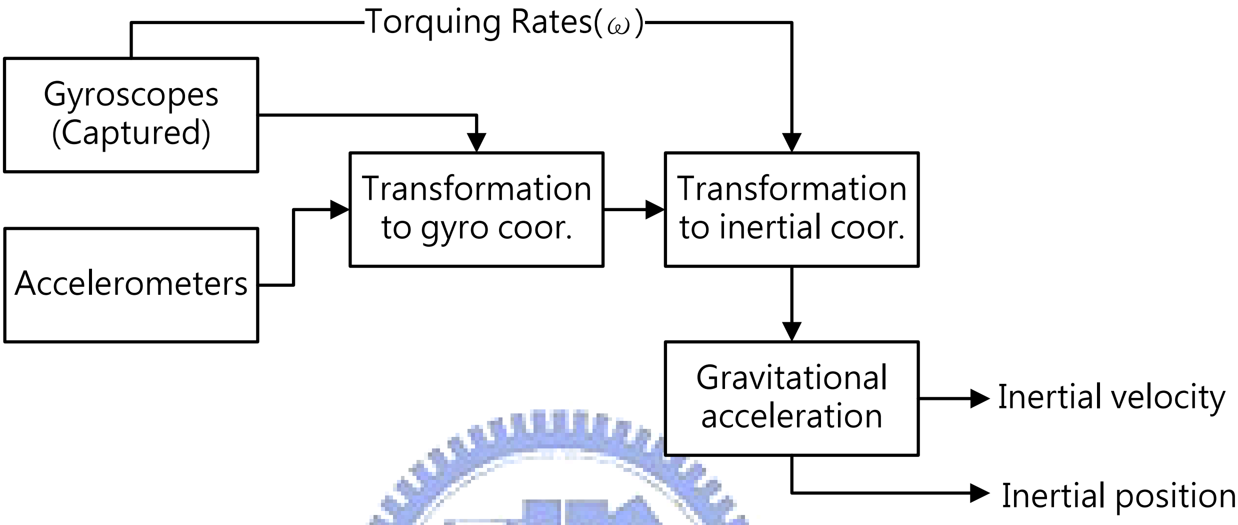


Figure 3.1: A simplified strapped down system [From "Simplified strapped down inertial navigation utilizing bang-bang gyro torquing"].

3.2 Accelerometer + Magnetometer

In [15], it replaced expensive gyroscopes with magnetometers. This IMU composed of accelerometers and magnetometer. It used the earth's magnetic field and gravity as the two measured quantities, a low-cost attitude determination system is proposed. It used the magnetic field vector and the acceleration, a unique plane containing the two vectors can be defined, and if the components of those two vectors can be measured in two non-aligned coordinate frames, then the rotation needed to align the two coordinate frames can be determined. In [16], the author also used a tri-accelerometer and a tri-magnetometer to determine the orientation of a static or slow-moving rigid body. It presented a geometrically intuitive 3-degree-of-freedom (3-DOF) orientation estimation algorithm with physical meaning which is called the factored quaternion algorithm (FQA). It is defined orientation by rotating it about its z-axis by an angle ψ (azimuth or yaw rotation), then about its y-axis by angle θ (elevation or pitch rotation), and finally about its x-axis by angle φ (bank or roll rotation). The elevation and roll rotation are determined from accelerometers. The azimuth rotation is determined from magnetometers. The value of a_x, a_y, a_z is from a tri-accelerometer. And the value of ${}^b m_x$, ${}^b m_y$, ${}^b m_z$ is from a tri-magnetometer in body coordinate system. The ${}^e m_x$, ${}^e m_y$, ${}^e m_z$ are magnetic vector in Earth coordinate system. The initial magnetic vector is $\begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^T$.

Elevation Quaternion :

$$q_e = \cos\frac{\theta}{2}(1\ 0\ 0\ 0) + \sin\frac{\theta}{2}(0\ 0\ 1\ 0)$$
$$\sin\theta = \frac{a_x}{g},\quad \cos\theta = \sqrt{1-\sin^2\theta}$$

Roll Quaternion :

$$q_r = \cos\frac{\varphi}{2}(1\ 0\ 0\ 0) + \sin\frac{\varphi}{2}(0\ 1\ 0\ 0)$$
$$\sin\varphi = \frac{-a_y}{g\cos\theta},\quad \cos\varphi = \frac{-a_z}{g\cos\theta}$$

Azimuth Quaternion :

$$q_a = \cos\frac{\psi}{2}(1\ 0\ 0\ 0) + \sin\frac{\psi}{2}(0\ 0\ 0\ 1)$$
$${}^e m = q_e\ q_r\ {}^b m\ q_r^{-1}\ q_e^{-1},\quad \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} {}^e m_x \\ {}^e m_y \end{bmatrix}$$

Having obtained all three rotation quaternion, the quaternion estimates representing the orientation of rigid body is finally given by

$$\widehat{q} = q_a\ q_e\ q_r$$

Although they are cheaper than the gyroscope, the magnetometer is easily interfered by other electronic device. When a ferrous object is close to those systems, they will get bigger noise and lost their accuracy.

3.3 Accelerometer + Gyroscope + Magnetometer

In order to improve the accuracy or reduce the error of orientation, much research had focused on using IMUs system that composed of accelerometers, gyroscopes and magnetometer. They not only used gyroscopes to calculate orientation but also magnetometers had been used to enhance accuracy and reduce error. In [7], it described a self-contained method for relative position tracking of a human engaged in various types of motion involving discrete steps. This method is based on the use of the inertial and magnetometer sensor module attached to the foot. This modules contained three orthogonally mounted angular rate sensors, three orthogonal linear accelerometers and three orthogonal magnetometers. In generally, the output of an accelerometer will be integrated twice to obtain displacement information. However it is susceptible to drift errors. A drift correction can be applied to the accelerometer with double integration method so that the final estimated velocity should be zero when a foot instant contacts with the ground. These inertial and magnetic sensor modules are primarily designed for tracking orientation. Although this system is high accuracy, it is not cheap and also is easily interfered by electronic devices.

3.4 IMU + Kalman filter

Not only using multi kinds of sensors but also applying Kalman filter to improve accuracy or reduce drift error on IMUs system. The Kalman filter is the most widely used state estimator for tracking applications. The filter is the general solution to the recursive linear minimum mean square estimation problem. In [17], it used two different adaptive Kalman filter. Each individual process noise covariance value will provide optimum estimates of the target's states only when underlying model, which is represented by that specific process noise covariance, is correct. For example a low level process noise covariance will match the target dynamics during an almost straight line motion period but fail to track when the target moves into a turn manoeuvre. The first filter is adjusted at each time step according to the estimated turn rate. The turning rate is estimated from acceleration divided by the estimated speed of target. Second uses a scale filter which is estimated from the available data of sensors reading. The first algorithm exploits the idea "the choice of the process covariance level must be made according to the expected turn rate" and utilizes an empirical turn rate-process noise covariance level curve. And the second algorithm introduces a scale factor which represents the current magnitude of process noise, i.e., target unpredictability, at time t as estimated from the available data. But a proper Kalman filter is complex and hard to design.

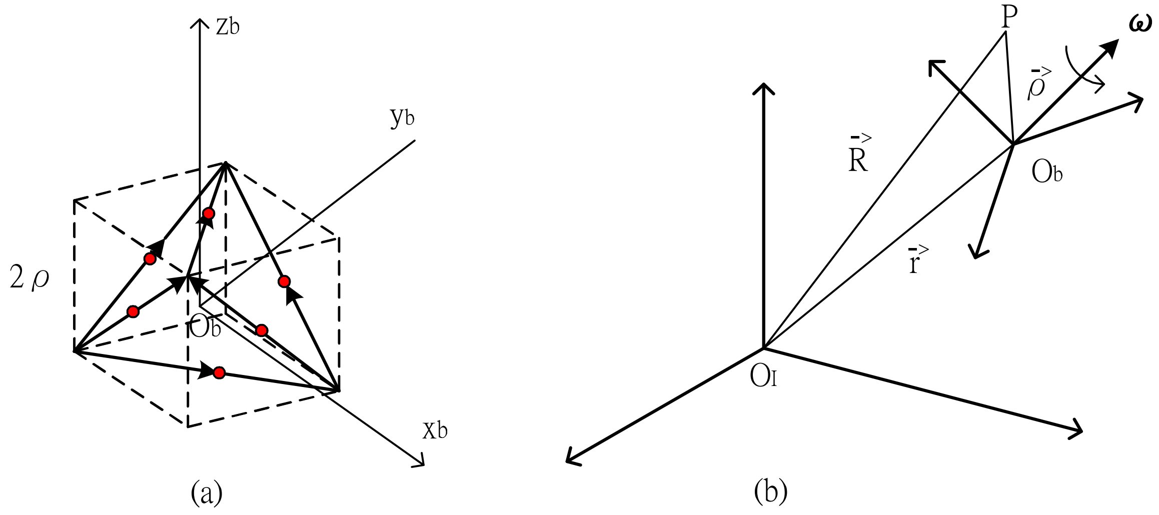


Figure 3.2: Six linear accelerometer system [From ”Gyroscope free Strapdown inertial measurement unit by six linear accelerometers”].

3.5 Six Linear Accelerometers

Gyroscopes are not inexpensive and magnetometers are easily interfered by other electronic devices. In 1994 Jeng-Heng [18] presented a method to determine the kinematics of a rigid body by using only linear accelerometers. In their work, six linear accelerometers are used for a complete description of a rigid body motion. They design one accelerometer at the center of each face of a cube and the sensing axis of each accelerometer is along the diagonal of respective cube face, shows as Figure 3.2 (a). The line of cube side is 2ρ and x_b, y_b, z_b are body frame of cube. O_b is the center of this cube. Figure 3.2 (b) shows relation of the inertial frame(O_I) and a rotating moving frame (O_b). The $\overrightarrow{R} = \overrightarrow{r} + \overrightarrow{\rho}$ then the accelerometer of point P can be calculated by double integration $\overrightarrow{R}$. S and T are 3×6 matrices After calculating, the formula will be produced :

$$\begin{bmatrix} \partial\overrightarrow{\omega} \\ \partial\overrightarrow{v} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} S/\rho \\ T \end{bmatrix} A + \rho \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_y\omega_z \\ \omega_x\omega_z \\ \omega_x\omega_y \end{bmatrix}$$

,where A is the acceleration from accelerometers, $\partial\overrightarrow{\omega}$ is the angular acceleration, the ω is the angular velocity by integrating $\partial\overrightarrow{\omega}$, $\partial\overrightarrow{v}$ is inertial acceleration.

$$S = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 & 0 & 0 & 1 & -1 \\ -1 & 0 & 1 & -1 & 0 & -1 \\ 0 & 1 & -1 & -1 & 1 & 0 \end{bmatrix}, T = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & -1 & -1 \\ 1 & 0 & 1 & -1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Although this system only uses accelerometer, it needs six accelerometer, fixed graph and fixed direction when installs the accelerometer. Because of the high accurate at the installing position is difficult, it needs to lathe to reach this request If the direction of accelerometer does not aligned, it will cause error. The major request is high accuracy for those accelerometers.

Some related works usually use gyroscopes or magnetometers to calculate change of orientation and accelerometer to get distance information. But they have the problem of expensive price and easily are influenced by electric/ferrous devices. Although the system of six linear accelerometers can also use pure accelerometers to tracking objects, it is hard to design and lathe is needed. In this work the g-sensor constellation system not only considers the price and is influenced by other device but also considers the sensor error itself. The pure accelerometers can reduce the cost and does not be influence by electric devices. And the LSM is a method to correct the error itself in this rigid graph system.

Chapter 4

Sensor Calibration

After each sensor leaves the factory, although the hardware specification is the same, the hardware itself has the trifle error to cause the reading to have the offset. In this chapter we will introduce some calibrated methods to correct offset of the sensor, including, single sensor calibration and multiple sensors calibration.

4.1 Single Sensor Calibration

Before using the inertial sensor to implement on all kinds of applications, it is basic and the most important first step to calibrate the sensor itself. In this section two single calibration methods will be introduced.

4.1.1 Linear Calibration

The linear calibration uses the linear function $y = a(x - b)$ to correct offset of the sensor. It means the reading of the sensor maybe have translation or scaling. Using linear function can efficient reduce this offset. The major steps are to measure the maximal and minimal reading of a sensor with X, Y, Z axes, respectively. Here we define the $f_y(i)$ to present the reading of the sensor which it should be. The $f_x(i)$ presents the measured reading of the sensor from the axis, where i presents the X, Y or Z axis. In order to measure the maximal and minimal reading of a sensor, each axis (X, Y, Z) must put toward to down and up, respectively which the reading should only be effect from gravity. When the axis puts toward to down, it means we can obtain the maximal reading $(f_{x-\max}(i))$ and the $f_{y-\max}(i)$ should be $1g$. On the other hand, the axis puts toward to up to get the minimal reading ($f_{x-\min}(i)$) and the $f_{y-\min}(i)$ should be $-1g$. The simultaneous equation is following :

$$f_{y-\max}(i) = 1 = a(f_{x-\max}(i) - b) \tag{4.1}$$

$$f_{y-\min}(i) = -1 = a(f_{x-\min}(i) - b) \tag{4.2}$$

from Eq. 4.1 and Eq. 4.2, we can solve variables a and b.

4.1.2 Least Square Calibration

Assuming the sensor has an offset at each X, Y and Z axis which are c_1, c_2 and c_3. The measured reading of a sensor at three axes is define to x_i, y_i, z_i, where i is the i-th sample. It can be presented to least square methods to find the minimum of $(x_i - c_1)^2 + (y_i - c_2)^2 + (z_i - c_3)^2$. Assuming the minimum is r^2, the formula can be written as :

$$(x_i - c_1)^2 + (y_i - c_2)^2 + (z_i - c_3)^2 = r^2. \tag{4.3}$$

This problem is similar to determine how close those points (you can think that x_i, y_i, z_i is a point in 3D space) are to being circular, we can fit a least squares circle to the data and check to see how close the measured points are to the circle. To fit a circle to n sample pairs of measured readings $(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)$ in a static state, we must determine the center (c_1, c_2, c_3) and the radius r. Rewriting Eq. 4.3, we get

$$2x_ic_1 + 2y_ic_2 + 2z_ic_3 + r^2 - c_1^2 - c_2^2 - c_3^2 = x_i^2 + y_i^2 + z_i^2.$$

If we set $c_4 = r^2 - c_1^2 - c_2^2 - c_3^2$, then the equation takes the form

$$2x_ic_1 + 2y_ic_2 + 2z_ic_3 + c_4 = x_i^2 + y_i^2 + z_i^2.$$

Substituting each of the measured reading with n sample into this equation, we obtain the overdetermined system

$$\begin{bmatrix} 2x_1 & 2x_1 & 2x_1 & 1 \\ 2x_2 & 2x_2 & 2x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 2x_n & 2x_n & 2x_n & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} x_1^2 + y_1^2 + z_1^2 \\ x_2^2 + y_2^2 + z_2^2 \\ \vdots \\ x_n^2 + y_n^2 + z_n^2 \end{bmatrix}. \tag{4.4}$$

The Eq. 4.4 can be solve by using matrix system $\mathbf{Ax} = \mathbf{b}$ (see [13]) to find minimum.

4.2 Multiple Sensors Calibration

In our work we use multiple sensors to implement our system, so the multiple sensors calibration is need. As Figure 4.1, three axes of each sensor has deviation with other sensor. That means the $X/Y/Z$ axis of sensor 1 maybe be not aim at the $X/Y/Z$ axis of sensor 2 or sensor 3. So we must do the alignment between every sensor. The multiple sensors calibration (this also is called the sensors alignment method) is a method to correct this deviation. In our work the 1st sensor is denoted by S_1, is called as the body frame, and the 2nd and 3rd sensors, respectively denoted by S_2 and S_3, are aligned to the body frame. The transformation matrix $T^{S_i \rightarrow S_1}$ from the i-th sensor to the body frame, we need to sample the reading (in a static state) in three independent orientations. If $\overline{G}_{i,j}$, a 3×1 vector, is the j-th reading of the i-th sensor, we have $\overline{G}_{1,j} = T^{S_i \rightarrow S_1}\overline{G}_{i,j}$ for $i, j = 1, 2, 3$. The transformation matrices $T^{S_i \rightarrow S_1}$ can be obtained by solving linear systems. Note that $T^{S_1 \rightarrow S_1} = I$.

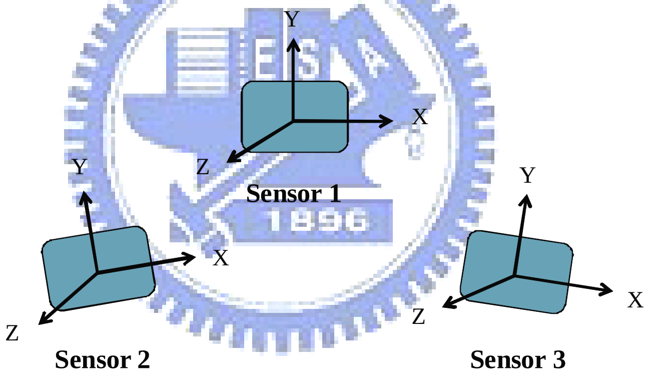


Figure 4.1: Alignment between sensors.

Chapter 5

G-Sensor Constellations

In this chapter we will introduce our g-sensor constellations architecture. In the beginning aligning between g-sensors is the first task in our work. The transformation matrix will also be described from initial coordinate to present coordinate. Finally the object tracking algorithm will be described, including extrapolation, LSM, adjust velocity and the object rotation.

5.1 Framework

A g-sensor constellation (abbreviated as constellation) is composed of at least three g-sensors with fixed topology. In a general case, a constellation may be composed of more than three sensors and the topology may not be a regular polygon. For example it can also compose of four/five or more g-sensors, as Figure 5.1, 5.2.

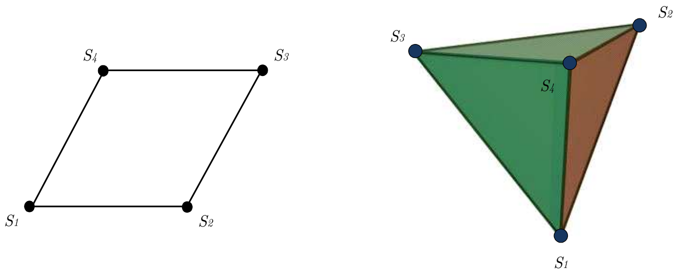


Figure 5.1: Constellations with four g-sensors.

In this work, a g-sensor constellation consisting of three Witilt V3 sensors are built to verify our tracking technique. The three sensors are stuck at the vertices of an equilateral triangle of 0.3m edge lengths. So, the constellation is depicted as in Figure 5.3. The

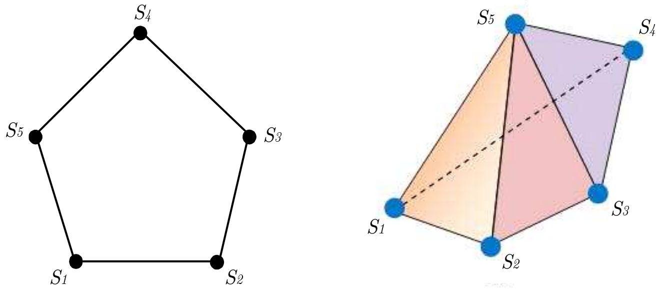


Figure 5.2: Constellations with five g-sensors.

constellation is output via Bluetooth interface. The receiver maybe is a computer or a phone which receive the data from the constellation. After receiving the data, the object tracing algorithm will be executed then the outcome will be input to the draw programming. The system framework is depicted as in Figure 5.4.

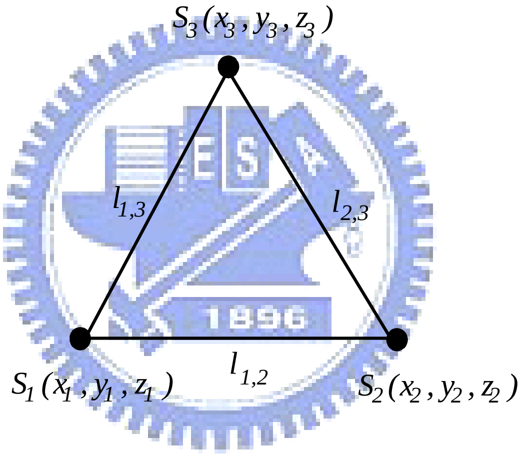


Figure 5.3: A g-sensor constellation.

Since the coordinates of three sensors may be not perfectly aligned, linear transformations are needed to align the coordinate systems of three sensors. The coordinate system of the 1st sensor, denoted by S_1, is called as the body frame, and the coordinate systems of the 2nd and 3rd sensors, respectively denoted by S_2 and S_3, are aligned to the body frame. To compute the transformation matrix $T^{S_i \to S_1}$ (or sometimes written as $T^{S_i \to B}$) from the i-th sensor to the body frame, we need to sample the reading (in a static state) in three independent orientations. If $\overline{G}_{i,j}$, a 3×1 vector, is the j-th reading of the i-th sensor, we have $\overline{G}_{1,j} = T^{S_i \to S_1}\overline{G}_{i,j}$ for $i, j = 1, 2, 3$. The transformation matrices $T^{S_i \to S_1}$ can be obtained by solving linear systems. Note that $T^{S_1 \to B} = T^{S_1 \to S_1} = I$. Let $\widetilde{G}_{i,j}$ denote the raw reading of the i-th sensor, i.e., measured in the coordinate system of S_i, and $G^B_{i,j}$ denote the reading of the i-th sensor measured in the body frame at time t_j.

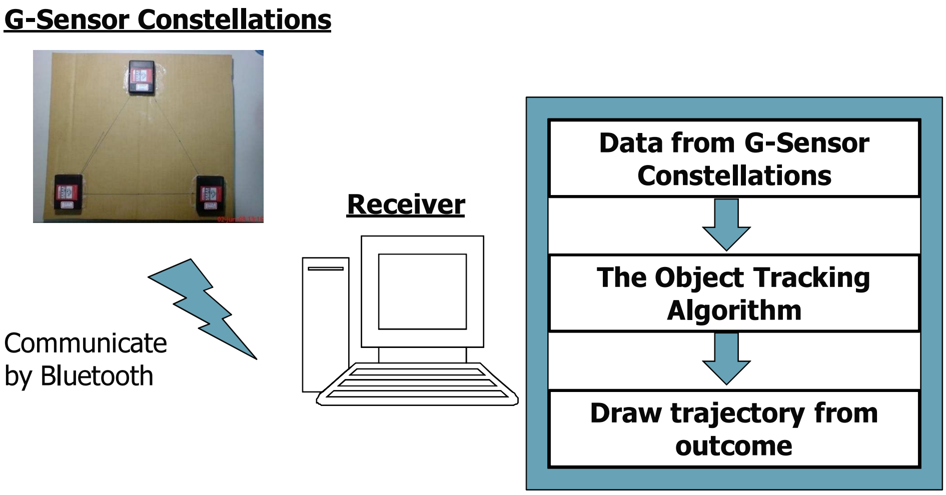

G-Sensor Constellations
Receiver
Communicate by Bluetooth
Data from G-Sensor Constellations
The Object Tracking Algorithm
Draw trajectory from outcome

Figure 5.4: System framework of the g-sensor constellation.

Then,

$$G^B_{i,j} = T^{S_1 \to B} \widetilde{G}_{i,j}.$$

Besides displacement, the constellation may rotate during the movement. A reference coordinate system is needed to record the movement of the constellation. We choose the body frame at time t_0 as the reference coordinate system and call it the earth frame. Let $T^{B\to E}_j$ be the linear transformation from the body frame to the earth frame at time t_j, and $G^E_{i,j}$ denote the reading of the i-th sensor measured in the earth frame at time t_j. Then,

$$G^E_{i,j} = T^{B\to E}_j G^B_{i,j}.$$

Let $S_{i,j}$ denote the position of the i-th sensor at time t_j measured in the earth frame. Then, the linear transformation matrix from the body frame to the earth frame at time t_j can be determined by the mapping from $\{S_{2,j} - S_{1,j}, S_{3,j} - S_{1,j}, (S_{2,j} - S_{1,j}) \times (S_{3,j} - S_{1,j})\}$ to $\{S_{2,0} - S_{1,0}, S_{3,0} - S_{1,0}, (S_{2,0} - S_{1,0}) \times (S_{3,0} - S_{1,0})\}$. Let $N_j = (S_{2,j} - S_{1,j}) \times (S_{3,j} - S_{1,j})$. Then, the transformation matrix $T^{B\to E}_j$ can be obtained by solving the following linear equation

$$\begin{aligned} &\begin{pmatrix} S_{2,0} - S_{1,0} & S_{3,0} - S_{1,0} & N_0 \end{pmatrix} \\ &= T^{B\to E}_j \begin{pmatrix} S_{2,j} - S_{1,j} & S_{3,j} - S_{1,j} & N_j \end{pmatrix}. \end{aligned} \tag{5.1}$$

Let $A_{i,j}$ be the acceleration of the i-th sensor at time t_j measured in the earth frame. Since $G^E_{i,j}$ is a composition of the gravity and the acceleration of the sensor, to obtain

$A_{i,j}$, the gravity must be deducted from $G^E_{i,j}$. If the constellation is in a static state at time t_0, then the gravity can be given by $G^E_{1,0}$, $G^E_{2,0}$, and $G^E_{3,0}$. So, for $i = 1, 2, 3$, we have

$$\begin{aligned} A_{i,j} &= G^E_{i,j} - G^E_{i,0} \\ &= T_j^{B\to E} T^{S_i \to B} \widetilde{G}_{i,j} - T_0^{B\to E} T^{S_i\to B} \widetilde{G}_{i,0} \\ &= T_j^{B\to E} T^{S_i\to B} \widetilde{G}_{i,j} - T^{S_i\to B} \widetilde{G}_{i,0}. \end{aligned} \tag{5.2}$$

5.2 The Object Tracking Algorithm

We iteratively keep tracking the positions and velocities of sensors by extrapolation based on the accelerations reported by g-sensors. However, accumulated error is a major concern and rotations of sensors can't be detected by sensors themselves. So, we develop a heuristic by utilizing the relative positions between sensors to adjust the displacements and velocities of sensors by least square methods (abbreviated as LSM). In the mean while, rotations can also be revealed.

In addition to position $S_{i,j}$ and acceleration $A_{i,j}$, let $V_{i,j}$ denote the velocity of the i-th sensor at time t_j measured in the earth frame, and $\widetilde{S}_{i,j+1}$ denote the estimated position of the i-th sensor at time t_{j+1} obtained from $S_{i,j}$ and $V_{i,j}$ by extrapolation. Based on the topology information of the constellation, $S_{i,j+1}$ is obtained and $V_{i,j+1}$ is also calculated. The process of object tracking from time t_j to time t_{j+1} is depicted in Figure 5.5. The process will repeat again and again.

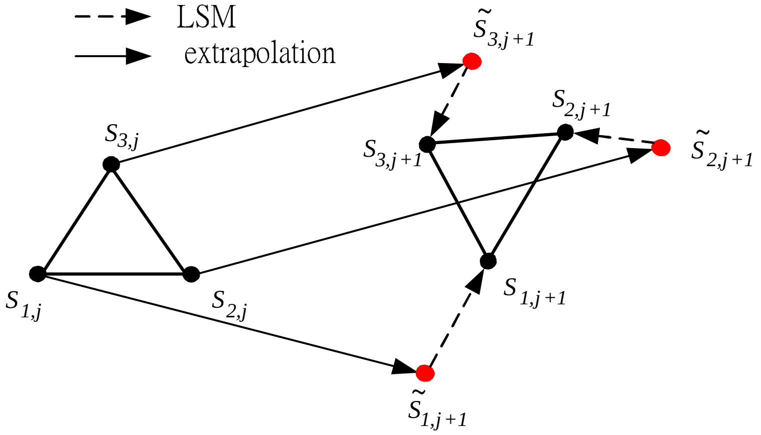


Figure 5.5: Object tracking by extrapolation and least square method.

5.2.1 Extrapolation

Let $\triangle t_j = t_{j+1} - t_j$, i.e., the time span between the j-th and $(j+1)$-th samples. Assume the motion from t_j to t_{j+1} has a constant acceleration. Then, $\widetilde{S}_{i,j+1}$ can be obtained by applying the constant acceleration motion formula

$$\widetilde{S}_{i,j+1} = S_{i,j} + V_{i,j}\triangle t_j + \frac{1}{2}A_{i,j}\left(\triangle t_j\right)^2. \tag{5.3}$$

This also is called the extrapolation method.

5.2.2 Least Square Method

Since the constellation is rigid, the topology metrics between sensors is fixed, especially the distance. However, these geometric properties may be broken by the extrapolation process. Therefore, the predicted positions can be adjusted based on topology information. We apply the least square method (abbrev. as LSM) given in Alg. 1 to do adjustment. For convenience, if it is not necessary to emphasize the time index j, we frequently suppress the index j and simply use S_i to denote the position of the i-th sensor.

Let $S_i = (x_i, y_i, z_i)$ for $i = 1, 2, 3$ be the position of the i-th sensor; and $\widetilde{S}_i = (\widetilde{x}_i, \widetilde{y}_i, \widetilde{z}_i)$ for $i = 1, 2, 3$ be the predicted position of the i-th sensor that is obtained by the extrapolation method. Due to the measurement error of acceleration and the accumulation error of velocity, the predicted position may not precise. Since the system is rigid, the topology metrics between S_1, S_2, S_3 is fixed, especially the distance. We have $\|S_2 - S_1\| = l_{1,2}$, $\|S_3 - S_2\| = l_{2,3}$, and $\|S_1 - S_3\| = l_{1,3}$. So, we can calibrate the predicted position under the topology constraints by the least square method.

Algorithm 1 [Least Square Problem (LSP)]

```
for all $\widetilde{S}_i$, $i = 1, 2, 3$, and $l_{i,k}$, $1 \le i < k \le 3$ such that $\|S_i - S_k\| = l_{i,k}$ do
    Measurement: Minimize $\sum_{i=1}^{3} \left\| S_i - \widetilde{S}_i \right\|^2$
end for
```

To solve the LSM, we apply Lagrange multipliers $\lambda_{i,j}$ for $1 \le i < j \le 3$. Let

$$\begin{aligned} f\left(S_1, S_2, S_3\right) &= \sum_{i=1}^{3} \left\| S_i - \widetilde{S}_i \right\|^2 \\ &= \sum_{i=1}^{3} \left(x_i - \widetilde{x}_i\right)^2 + \left(y_i - \widetilde{y}_i\right)^2 + \left(z_i - \widetilde{z}_i\right)^2, \end{aligned}$$

, for $1 \leq i < k \leq 3$

$$\begin{aligned} & g_{i,k}\left(S_1, S_2, S_3\right) \\ & = \left\|S_i - S_k\right\|^2 - \left(l_{i,k}\right)^2 \\ & = \left(x_i - x_k\right)^2 + \left(y_i - y_k\right)^2 + \left(z_i - z_k\right)^2 - \left(l_{i,k}\right)^2, \end{aligned}$$

and

$$\begin{aligned} & \Lambda\left(P_1, P_2, P_3, \lambda_1, \lambda_2, \lambda_3\right) \\ & = f\left(S_1, S_2, S_3\right) - \sum_{1 \leq i < k \leq 3} \lambda_{i,k} g_{i,k}\left(S_1, S_2, S_3\right). \end{aligned}$$

To obtain S_1, S_2, S_3, we solve the system

$$\begin{cases} \nabla_{x_i, y_i, z_i} \Lambda = \mathbf{0} \text{ for } i = 1, 2, 3; \\ \nabla_{\lambda_{1,2}, \lambda_{1,3}, \lambda_{2,3}} \Lambda = \mathbf{0}. \end{cases}$$

For $i = 1, 2, 3$, from $\frac{\partial}{\partial x_i}\Lambda = 0$, $\frac{\partial}{\partial y_i}\Lambda = 0$, and $\frac{\partial}{\partial z_i}\Lambda = 0$, we have $S_i - \widetilde{S}_i = \sum_{k \neq i} \lambda_{i,k}\left(S_k - S_i\right)$, or, more explicitly,

$$\begin{pmatrix} x_i - \widetilde{x}_i \\ y_i - \widetilde{y}_i \\ z_i - \widetilde{z}_i \end{pmatrix} = \begin{pmatrix} x_j - x_i & x_k - x_i \\ y_j - y_i & y_k - y_i \\ z_j - z_i & z_k - z_i \end{pmatrix} \begin{pmatrix} \lambda_{j,i} \\ \lambda_{k,i} \end{pmatrix}.$$

For $1 \leq i < j \leq 3$, from $\frac{\partial}{\partial \lambda_{i,j}}\Lambda = 0$, we have $\left\|S_i - S_j\right\| = l_{i,j}$, or, more explicitly,

$$\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(y_i - y_j\right)^2 = \left(l_{i,j}\right)^2.$$

Put all together and then we have

$$\begin{aligned} x_1 + x_2 + x_3 &= x_1' + x_2' + x_3', \\ y_1 + y_2 + y_3 &= y_1' + y_2' + y_3', \\ z_1 + z_2 + z_3 &= z_1' + z_2' + z_3'. \end{aligned}$$

Therefore, the barycenter of S_1, S_2, S_3 and $\widetilde{S}_1, \widetilde{S}_2, \widetilde{S}_3$ are overlapping, and we have the following theorem.

Theorem 3 *For the LSP, the barycenters of $S_1 S_2 S_3$ and $\widetilde{S}_1 \widetilde{S}_2 \widetilde{S}_3$ are overlapping.*

Put a constellation at the barycenter of $\widetilde{S}_1\widetilde{S}_2\widetilde{S}_3$, then based on Theorem 3, the LSP can be solved by finding a best rotation. In this work, we further assume that $S_1S_2S_3$ and $\widetilde{S}_1\widetilde{S}_2\widetilde{S}_3$ are in the same plane, and therefore, the solution can be found by a 2D rotation. A numerical method is used to find the best rotation. The numerical method is used to solve the LSM which described by following :

In the beginning, whose vertices are S_1, S_2 and S_3 and G is the center of mass of the triangle. Let $\angle S_1GS_2 = \theta_3$, $\angle S_2GS_3 = \theta_1$ and $\angle S_3GS_1 = \theta_2$, then we can use law of cosines to get θ_1, θ_2 and θ_3. After time $\triangle t$, the vertices of triangle become $\widetilde{S}_1$, $\widetilde{S}_2$ and $\widetilde{S}_3$, which are derived using the extrapolation method. So we can get $\widetilde{G}$ the center of mass whose vertices are $\widetilde{S}_1$, $\widetilde{S}_2$ and $\widetilde{S}_3$, which equals G by theorem 3. We also can decide the initial position of S_1. Assume the unit normal vector becomes $\mathbf{u} = (u_x, u_y, u_z)$ whose vertices are $\widetilde{S}_1$, $\widetilde{S}_2$ and $\widetilde{S}_3$. Now, we need to find the coordinates of S_2 and S_3. Then find the best possible coordinates of S_1, S_2 and S_3. Then rotation about an arbitrary unit vector (u_x, u_y, u_z) by an angle θ is given by a matrix by Theorem 1

$$M(\mathbf{u},\theta) = \begin{bmatrix} c + (1-c)u_x^2 & (1-c)u_xu_y - (s)u_z & (1-c)u_xu_z + (s)u_y \\ (1-c)u_yu_x + (s)u_z & c + (1-c)u_y^2 & (1-c)u_yu_z - (s)u_x \\ (1-c)u_zu_x - (s)u_y & (1-c)u_zu_y + (s)u_x & c + (1-c)u_z^2 \end{bmatrix}$$

,where $c = \cos\theta$, $s = \sin\theta$

Using the theorem above, we have $\overrightarrow{GS_2} = M(\mathbf{u},\theta_3)\overrightarrow{GS_1}\frac{\|\overrightarrow{G\widetilde{S}_2}\|}{\|\overrightarrow{G\widetilde{S}_1}\|}$ and $\overrightarrow{GS_3} = M(\mathbf{u},\theta_2)\overrightarrow{GS_1}\frac{\|\overrightarrow{G\widetilde{S}_3}\|}{\|\overrightarrow{G\widetilde{S}_1}\|}$. Now, to find the best possible coordinates of S_1, S_2 and S_3, calculate $\overrightarrow{GS_i^{'}} = M(\mathbf{u},\theta)\overrightarrow{GS_i}$ for $i = 1,2,3$ and $-\pi \leq \theta \leq \pi$. So, our goal is to find $\widehat{\theta}$ such that $\sum_i \left\|\overrightarrow{G\widetilde{S}_i} - \overrightarrow{GS_i^{'}}\right\|^2$ has the minimum. This means that

$$\widehat{\theta} = \arg\min\left(\sum_i \left\|\overrightarrow{G\widetilde{S}_i} - \overrightarrow{GS_i^{'}}\right\|^2, \theta\right).$$

Having $\widehat{\theta}$, the best possible coordinates can be found.

5.2.3 Adjust Velocity

After applying the LSM, we get the positions $S_{i,j+1}$. We can further to adjust the velocities of sensors according to the displacement. Let $D_{i,j} = S_{i,j+1} - S_{i,j}$ be the displacement of the i-th sensor in the period from the j-th sampling time to the $(j+1)$-the sampling time.

In addition, we also have

$$\begin{aligned} D_{i,j} &= V_{i,j}\Delta t_j + \frac{1}{2}A_{i,j}\left(\Delta t_j\right)^2 \\ &= \frac{1}{2}\Delta t_j\left(V_{i,j} + V_{i,j+1}\right). \end{aligned}$$

So, the velocity $V_{i,j+1}$ can be given by

$$V_{i,j+1} = \frac{2\left(S_{i,j+1} - S_{i,j}\right)}{\Delta t_j} - V_{i,j}. \tag{5.4}$$

5.2.4 Object Rotation

From Eq. 5.1 we can get the transformation matrix $T_j^{B\to E}$ and by Theorem 2 the $Z-Y-X$ rotation matrix is known.

$$T_j^{B\to E} = \begin{bmatrix} T_{00} & T_{01} & T_{02} \\ T_{10} & T_{11} & T_{12} \\ T_{20} & T_{21} & T_{22} \end{bmatrix} = \begin{bmatrix} c_1c_2 & -c_2s_1 & s_2 \\ c_3s_1 + c_1s_2s_3 & c_1c_3 - s_1s_2s_3 & -c_2s_3 \\ s_1s_3 - c_1c_3s_2 & c_3s_1s_2 + c_1s_3 & c_2c_3 \end{bmatrix}$$

,where $c_1 = \cos\gamma, c_2 = \cos\beta, c_3 = \cos\alpha$ and $s_1 = \sin\gamma, s_2 = \sin\beta, s_3 = \sin\alpha$.

Then $s_2 = \sin\beta = T_{02}, \beta$ can be obtained, α, γ can also be obtained from β and T_{01}, T_{12}.

Finally the Figure 5.6 shows the procedure of this algorithm. In the beginning the coordinate, velocity and transformation matrix will be initiated. The alignment between g sensors will also be calculated in this section. After initial and alignment have done, the new coordinates of g sensors will be calculated by extrapolation. Then the corrected coordinates will be calculated from new coordinates by LSM. Finally the velocity and transformation matrix can be determined by initial coordinates and corrected coordinates. Rotation information can also be got from the transformation matrix.

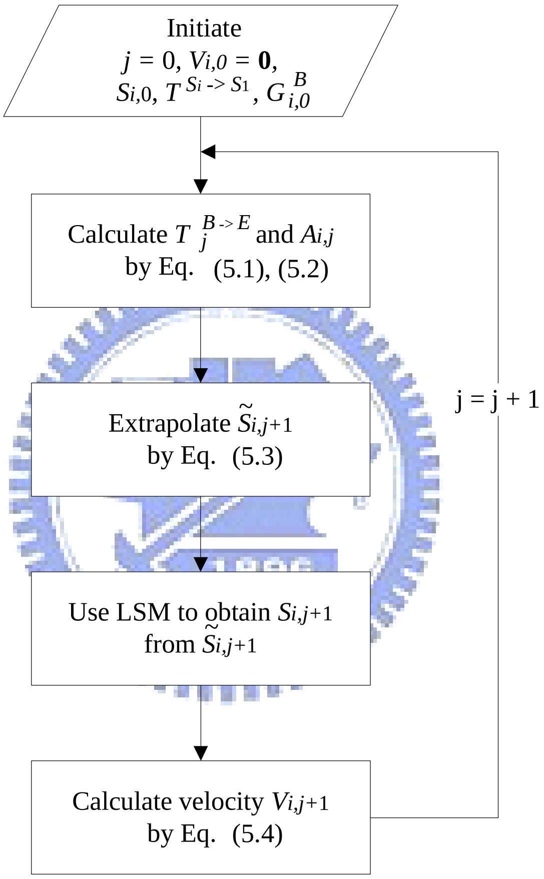

Initiate
j = 0, $V_{i,0}$ = **0**,
$S_{i,0}$, $T^{S_i \to S_1}$, $G^{B}_{i,0}$
Calculate $T^{B \to E}_{j}$ and $A_{i,j}$ by Eq. (5.1), (5.2)
Extrapolate $\tilde{S}_{i,j+1}$ by Eq. (5.3)
Use LSM to obtain $S_{i,j+1}$ from $\tilde{S}_{i,j+1}$
Calculate velocity $V_{i,j+1}$ by Eq. (5.4)
j = j + 1

Figure 5.6: Flowchart of the object tracking algorithm.

Chapter 6

Experiments

In our experiments, the constellation is composed of three Witilt V3 sensors which has a triaxial accelerometer and a single axis gyroscope. But only the accelerometer is used. The sampling rate is set up $90Hz$. The sampling range is around $\pm 6g$. Three sensors are fixed on a plastic board at the vertices of an equilateral triangle. The edge of the equilateral triangle is $0.3m$. The data of Witilt V3 sensors is output via a Bluetooth interface.

6.1 Static Experiment

The g-sensor is also with a certain level of inaccuracy after signal calibration (here, we use the linear calibration method). Even sensors are in a static state, the reading of sensors slightly vibrate. Figure 6.1, Figure 6.2 and Figure 6.3 show the vibratility of three sensors. The inaccuracy will accumulate and cause significant error on velocities and displacements. In the first experiment, the constellation keeps static for 30s, and the drifting that caused by accumulation error is calculated for three individual sensors and the constellation. The constellation is represented by its barycenter. The field with LSM by average in Table 6.1 describes a method for determining the output value of the sensor when the sensor is static. It is meaning a sensor is static for 30 seconds, we gather all output values in this interval and calculate the average of those values. When we do calculation by our object tracking algorithm, every reading value from sensors will be subtract this average value. For example if the average value is x when the objects keeps for an interval. When we get the reading value from a sensor is y. The really output value

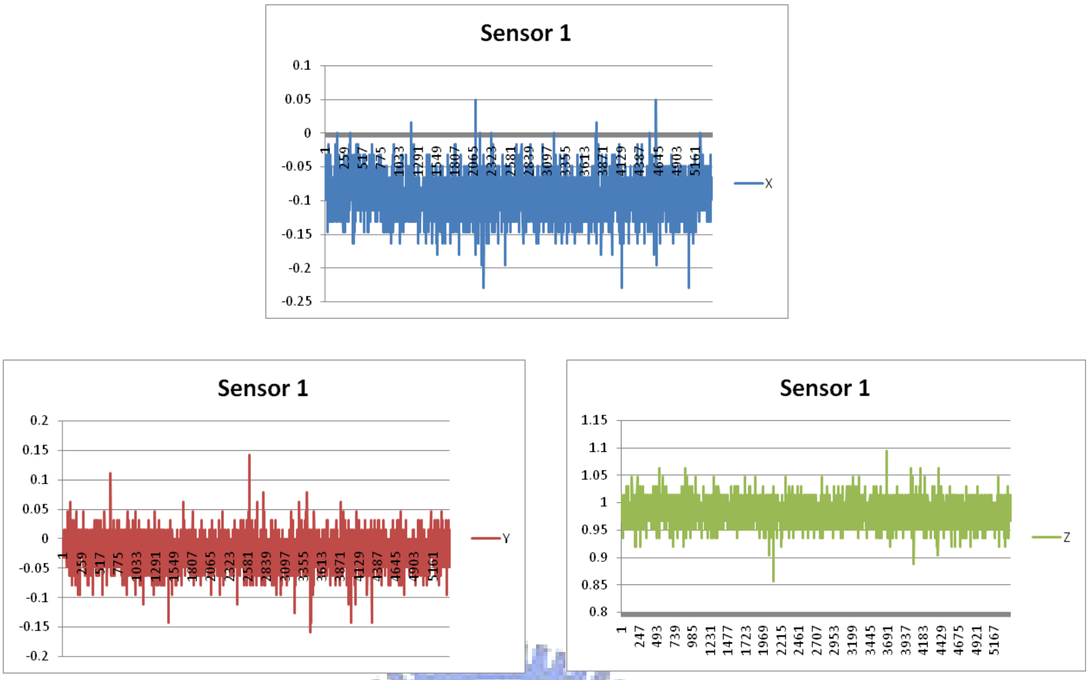


Figure 6.1: The vibratility of sensor 1 in a static state.

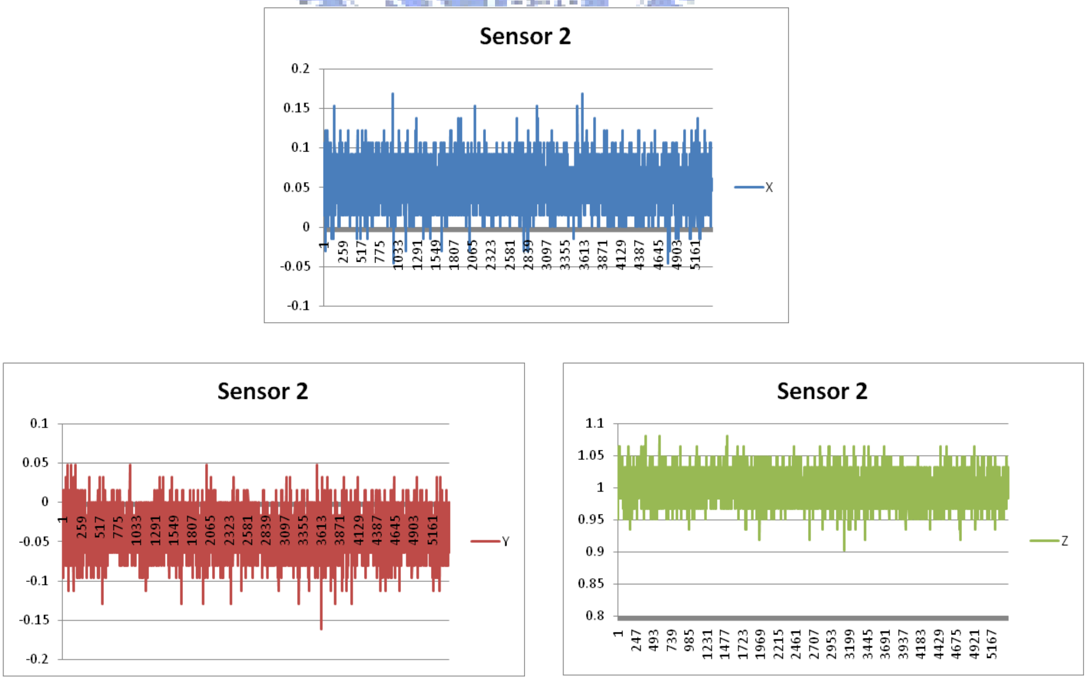


Figure 6.2: The vibratility of sensor 2 in a static state.

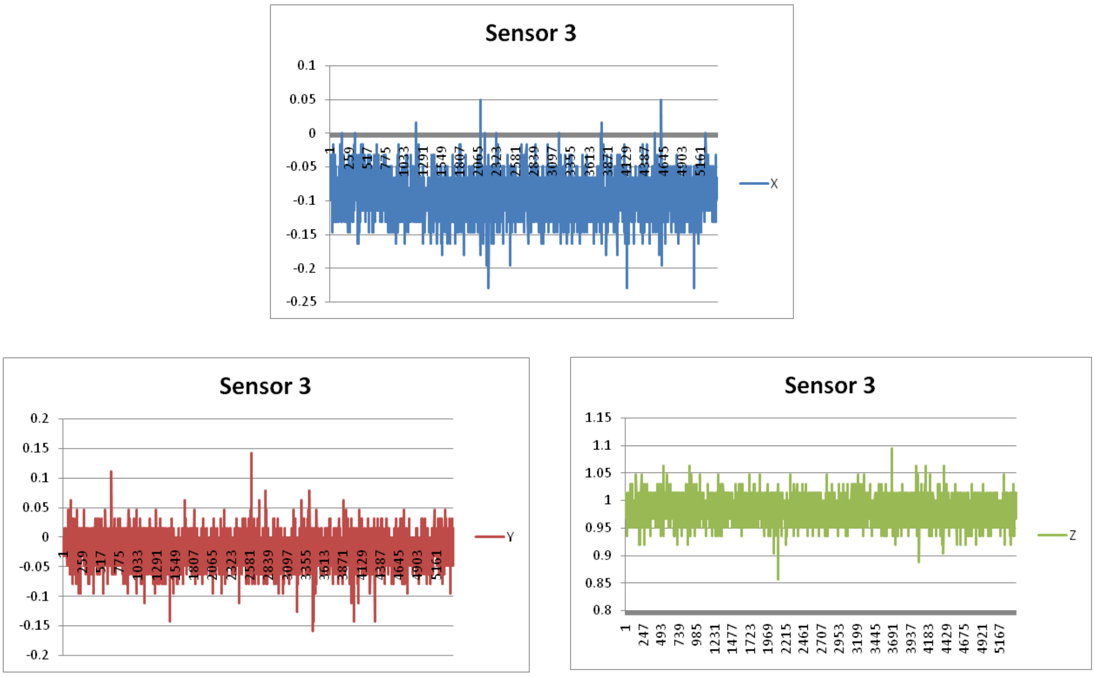


Figure 6.3: The vibratility of sensor 3 in a static state.

is $y - x$. This goal is to reduce the vibratility in sensor itself. The field with LSM by threshold. In Table 6.1 is another method to do this. This method is to set a threshold to reduce the vibratility in sensor itself. For example, if threshold sets as a value, the reading value from a sensor is b and the average is x. The really output value is $|b - x|$ if $|b - x| \geqq a$ else 0 if $|b - x| < a$. Figure 6.4 shows the displacement and angle what threshold is better. The threshold is set from between 0.01 and 0.1. Finally, the threshold is set to 0.03 in our experiments. Table 6.1 records the displacement of each sensors and the constellation with LSM by average and LSM by threshold methods.We can see the

Table 6.1: The drifting distance of a static constellation after 30 seconds.

	S1	S2	S3	LSM by average	LSM by threshold
X	12.3 m	3.6 m	3.9 m	6.8 m	3.2 m
Y	5.2 m	3.1 m	-0.8 m	2.6 m	2.5 m
Z	0.9 m	3.0 m	4.4 m	2.7 m	1.6 m

constellation is the stablest with different methods.

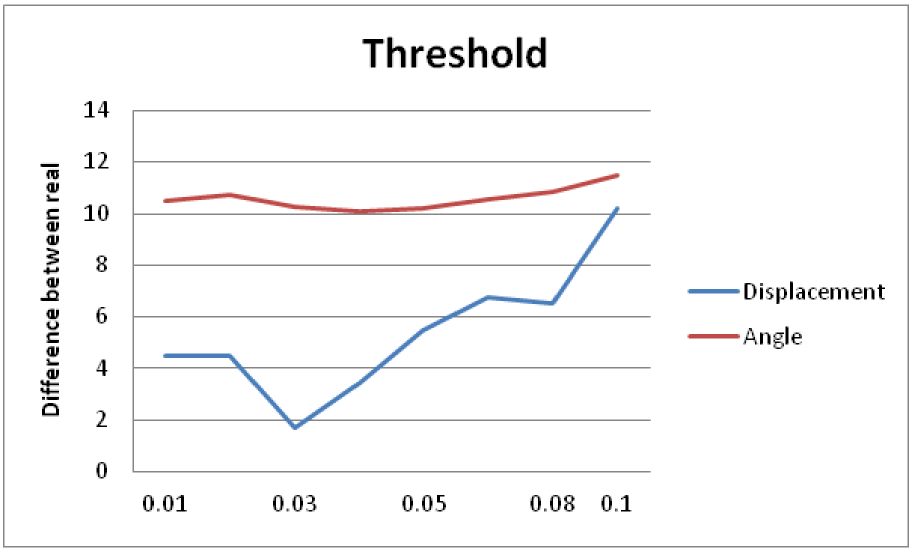


Figure 6.4: Displacement and angle with different thresholds.

6.2 Static Rotation Experiment

We also do the rotation tests. In a rotation test, rotate the constellation counterclockwise about the Z-axis by $30°, 60°$ and $90°$ at static state. Thus the z-component of each $A_{i,j}$ should be zero. Figure 6.6, Figure 6.8 and Figure 6.10 show the result of the rotation test with LSM by average We also do same tests with LSM by threshold Figure 6.7, Figure 6.9 and Figure 6.11 show the this result. Table 6.2 shows the rotational angle by $Z-Y-X$ axis (yaw,pitch,roll) for all methods. Figure 6.5 shows the relation between time and angle. We can see that, after being calibrated by the LSM, the angle between the orientation of the initial and the final constellation is better than without LSM.

Table 6.2: The Yaw-Pitch-Roll angle by rotating 30°, 60° and 90°.

	Yaw	Pitch	Roll
30° with average	22.2	0	0.09
60° with average	50.6	0	0.09
90° with average	79.6	0	0.09
30° with threshold	22.3	0	0.08
60° with threshold	51.7	0	0.08
90° with threshold	79.8	0	0.08

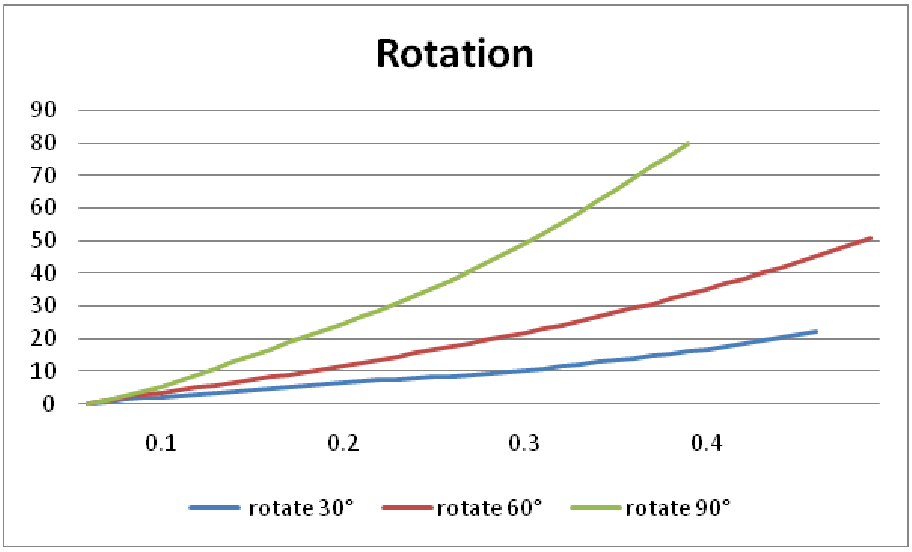
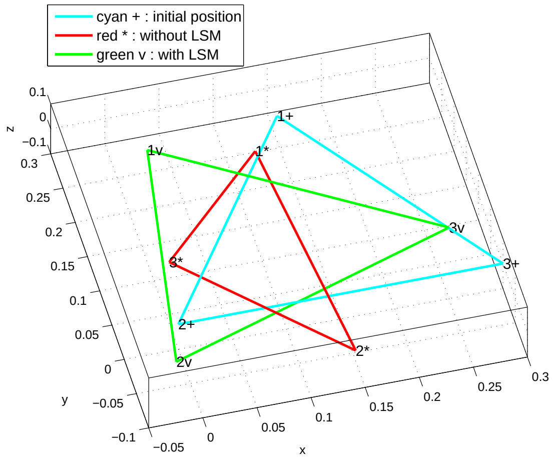


Figure 6.5: Rotation along time.



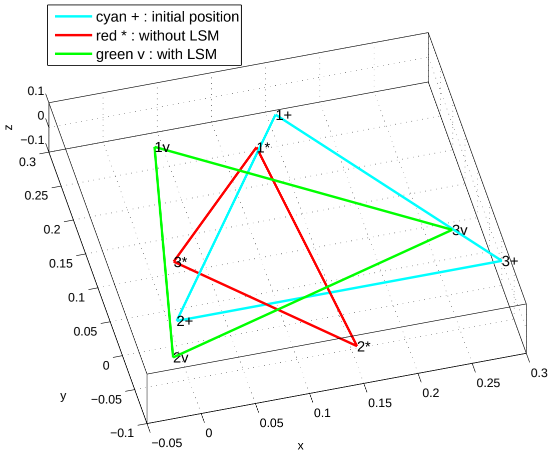


Figure 6.6: Rotate 30°by Z-axis with average method.

Figure 6.7: Rotate 30° by Z-axis with threshold method.

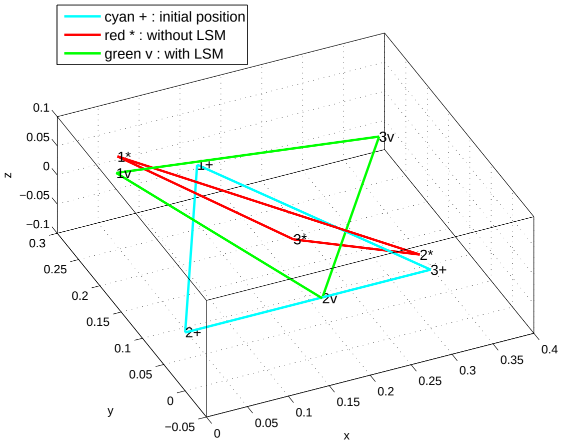


Figure 6.8: Rotate 60° by Z-axis with average method.

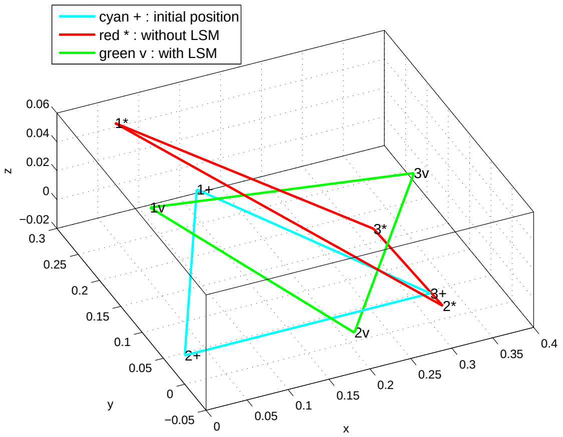


Figure 6.9: Rotate 60° by Z-axis with threshold method.

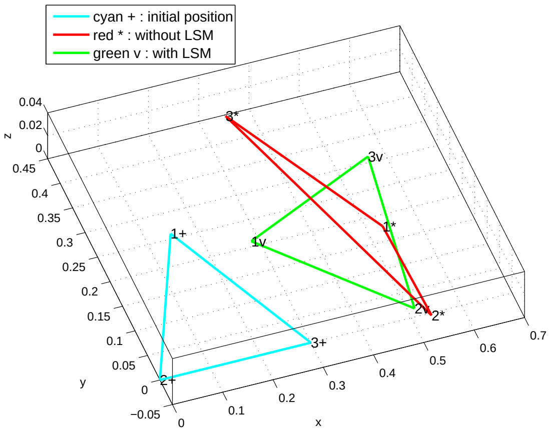


Figure 6.10: Rotate 90° by Z-axis with average method.

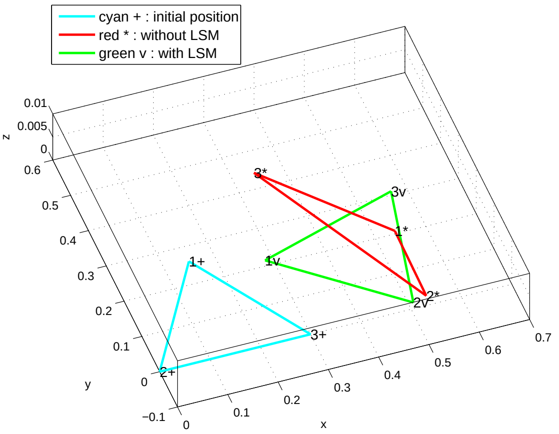


Figure 6.11: Rotate 90° by Z-axis with threshold method.

6.3 Dynamic Experiment in a Straight Line

Besides the static experiment, we also do some dynamic experiments. For example, the constellation moves about $45m$ in a straight line. The outcome of the straight line test is depicted in Figure 6.12 and Figure 6.13. Figure 6.12 shows the result of straight line with LSM by average method. Figure 6.13 shows the result of straight line with LSM by threshold method. From the reconstructed tracks, the constellation marked by the red curve moves almost in a straight line. However, the tracks of three individual sensors are far from a straight line. We also calculate how far sensors move. The moving distances of the sensors and the constellation are is calculated by $\sqrt{x^2+y^2+z^2}$, and the outcomes are listed in Table 6.3.

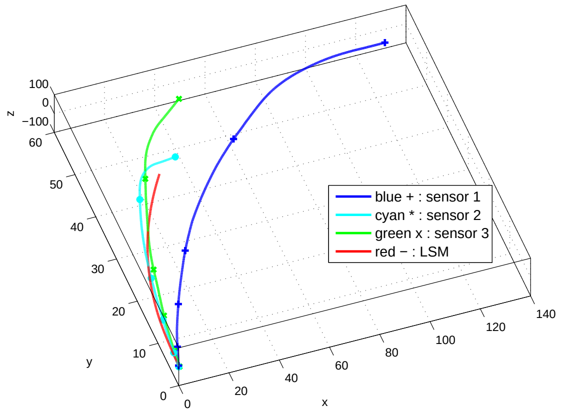


Figure 6.12: The track of a striaght line movement by average method.

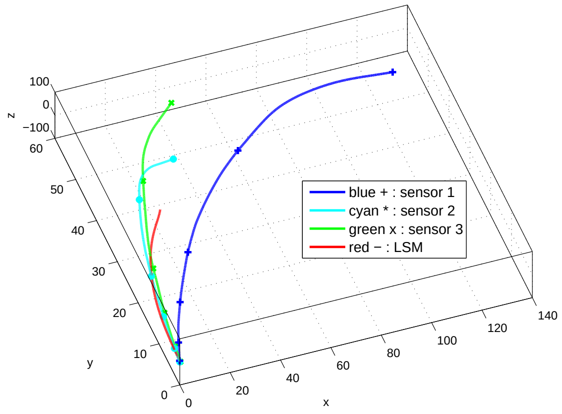


Figure 6.13: The track of a striaght line movement by threshold method.

Table 6.3: The distance measured after the constellation moves $45m$ in a straigt line.

	S1	S2	S3	LSM
Distance by average	150m	57m	74m	49.5m
Distance by threshold	148m	56m	72m	43.3m

6.4 Dynamic Experiment in a S-curve

Finally we do a dynamic moving and rotation test. For example, the constellation moves about a S-curve. The outcome of the S-curve test is depicted in Figure 6.14 and Figure 6.15. Figure 6.14 shows the result of S-curve with LSM by average method. Figure 6.15 shows the result of S-curve with LSM by threshold method. From the reconstructed tracks, the constellation marked by the red curve moves almost in a S-curve. Senor1, senor 2 and sensor 3 are far from a S-curve. The constellation is smooth better than three individual sensors.

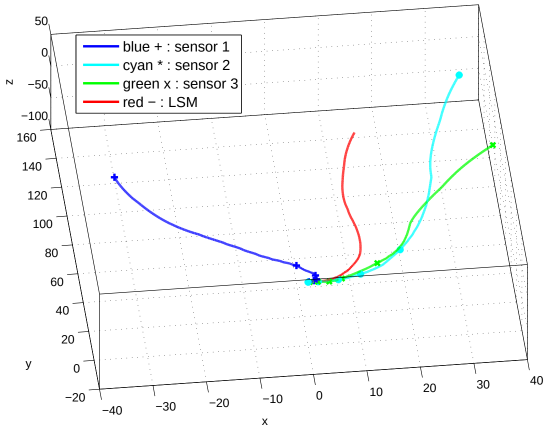


Figure 6.14: The track of a S-curve movement by average method.

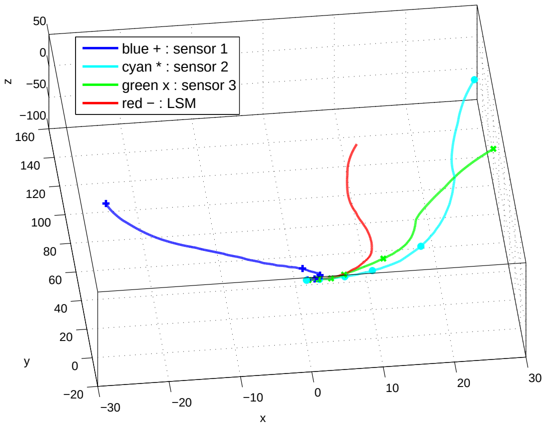


Figure 6.15: The track of a S-curve movement by threshold method.

Chapter 7

Conclusions

In this work, we proposed an inexpensive IMU solution called g-sensor constellations in which only accelerometers are used. It does not need gyroscopes and magnetometers to detect motion directions and the rotations of objects. The proposed technique can be applied to many applications, e.g. vehicle accident reconstruction, dead reckoning for PNSs, or human-computer interfaces. In the future, we will extend this work for constellations composed of more than three sensors and give analysis of accuracy. From our experiments, we also found that inaccurate calibration of sensors may cause significant accumulation error. Therefore, it is important to integrate calibration algorithm into the system.

Bibliography

[1] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, January 2006.

[2] A. Ofstad, E. Nicholas, R. Szcodronski, and R. R. Choudhury, "AAMPL: accelerometer augmented mobile phone localization," in *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments (MELT 2008)*, 19 September 2008, pp. 13–18.

[3] D. Alvarez, R. C. González, A. M. López, and J. C. Alvarez, "Comparison of step length estimators from weareable accelerometer devices," in *Proceedings of the 2006 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2006)*, 30 August - 3 September 2006, pp. 5964–5967.

[4] J. C. Alvarez, R. C. González, D. Alvarez, A. M. López, and J. Rodríguez-Uría, "Multisensor approach to walking distance estimation with foot inertial sensing," in *Proceedings of the 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2007)*, 22-26 August 2007, pp. 5719–5722.

[5] A. M. Sabatini, C. Martelloni, S. Scapellato, and F. Cavallo, "Assessment of walking features from foot inertial sensing," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 3, pp. 486–494, March 2005.

[6] K. Sagawa, H. Inooka, and Y. Satoh, "Non-restricted measurement of walking distance," in *Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics*, 8-11 October 2000, pp. 1847–1852 vol.3.

[7] X. Yun, E. R. Bachmann, H. Moore, and J. Calusdian, "Self-contained position tracking of human movement using small inertial/magnetic sensor modules," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, 10-14 April 2007, pp. 2526–2533.

[8] T. S. M. Hide, C. Moore, "Adaptive kalman filtering algorithms for integrating gps and low cost ins," in *Position Location and Navigation Symposium, 2004. PLANS 2004*, 26-29 April 2004, pp. 227 – 233.

[9] B. A. B. A. A. Rezaie, J. Moshiri, "Gps/ins integration using nonlinear blending filters," in *SICE, 2007. Annual Conference*, 17-20 Sept. 2007, pp. 1674 – 1680.

[10] C.-M. H. Dah-Jing Jwo, "An adaptive fuzzy strong tracking kalman filter for gps/ins navigation," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, 5-8 Nov. 2007, pp. 2266 – 2271.

[11] S. U. Byung-Uk Lee, "Unit quaternion representation of rotation," 1991.

[12] C. S. o. M. U. Murray, G., "Rotation about an arbitrary axis in 3 dimensions," 2005, pp. 1–5.

[13] D. Steven J. Leon, University of Massachusetts, "Linear algebra," 2003, p. chapter 4.

[14] E. G. S. W. Peter Michael Brodie, Oakland, "Simplified strapped down inertial navigation utilizing bnag-bang gyro torquing," in *US patent*, 26 July 1997.

[15] J. D. P. Demoz Gebre-Egziabher, Gabriel H. Elkaim and B. W. Parkinson, "A gyro-free quaternion-based attitude determination system suitable for implementation using low cost sensors," in *Position Location and Navigation Symposium, IEEE 2000*, 13-16 March 2000, pp. 185–192.

[16] E. R. B. Xiaoping Yun and R. B. McGhee, "A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements," in *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 57, NO. 3*, March 2008, pp. 638 – 650.

[17] B. Sch. of Eng., Sussex Univ., “Adaptive kalman filters for manoeuvring target tracking,” in *Target Tracking and Data Fusion (Digest No. 1998/282), IEE Colloquium on*, 9 June 1998, pp. 4/1 – 4/7.

[18] L. S. C. Chen J. H. and D. D. B., “Gyroscope free strapdown inertial measurement unit by six linear accelerometers,” in *AIAA Vol. 17, No.2*, March-April 1994, pp. 286–290.