Error Reduction Techniques for a MEMS Accelerometer-based Digital Input Device

TSANG, Chi Chiu

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

©The Chinese University of Hong Kong
March 2008

The Chinese University of Hong Kong holds the copyright of this thesis. Any person(s) intending to use a part or whole of the materials in the thesis in a proposed publication must seek copyright release from the Dean of the Graduate School.

To my parents and grandma, for their unreserved love.

Error Reduction Techniques for a MEMS Accelerometer-based Digital Input Device

submitted by

TSANG, Chi Chiu

for the degree of Master of Philosophy
at The Chinese University of Hong Kong

Abstract

In this office automation era, many handwritten documents are digitized for ease of backup and transmission. The demand for digital writing instruments is thus expected to grow rapidly in coming years. There are many different types of systems already available on the market. Recently, we have developed a prototype of a MEMS (Micro-Electro-Mechanical Systems) based 3D digital writing instrument which makes use of a micro inertial measurement unit (μIMU) constructed from MEMS accelerometers and gyroscopes for real-time capture of handwriting.

This idea has been proposed for over 10 years [43], however, to date there is still no successful product on the market. One reason is that the necessary double integration from acceleration to position propagates the noise associated with the sensors to the tracked position, and such errors increase without bound. In this thesis, a study of the sources of errors in μIMU, accelerometers was given. Different existing error reduction techniques are investigated and novel schemes proposed with an aim to provide a practical solution to this problem.

We have proposed two novel error reduction schemes for this digital writing instrument. They are developed from the Kalman filtering algorithm with the addition of absolute position information. One is from the fact that z-axis displacement in the navigation frame is always zero, and the other is the position measurement from an electromagnetic resonance (EMR) position detection board. With this extra information, the accuracy in the position estimate is improved. Particularly for the latter one, a successful handwriting can be produced with improved accuracy over previous methods.

基於微機電加速度傳感器的
數碼輸入設備的誤差減少技術研究

曾志超

香港中文大學
計算機科學與工程學課程
哲學碩士論文

二零零八年三月

摘要

在這個辦公室自動化的時代，為了方便保存和傳送，很多傳統手寫的文件都被數碼化。市場上亦出現了不同種類的產品以供用戶選擇，而未來幾年數碼輸入設備的需求將日益增加。最近，我們研發了一套基於微機電系統的三維數碼書寫設備。系統中包含了一個由三維微機電加速度傳感器和陀螺儀組成的微型慣性測量單元，藉以實時地記錄我們的書寫動作。

利用慣性測量單元記錄書寫動作的概念已經提出了有十年時間，但直至現在，市場上仍然沒有相關的產品。主要原因是在二次的積分過程中，傳感器內的噪音信息大大影響了筆跡的計算。有鑑於此，本論文會先針對加速度傳感器的噪音進行誤差分析，然後分析各種現有的誤差減少技術，並提出改進，藉此希望能夠解決上述誤差信息，從而為院校和商界提供一套可以在三維空間自由使用的數碼書寫系統。

利用卡爾曼濾波(Kalman filtering)算法和額外提供的絕對位置信息，我們提出了兩種新的針對於數碼書寫設備的誤差減少方案。一是基於二維的平面書寫z方向無位移的特點；而另一個則是通過電磁反饋裝置提供的位置信息，來提高位置預測的準確性。特別是使用後述的方法，我們可以成功地將書寫的筆跡重新在電腦上顯示出來，而準確度高於以前的方法。

Acknowledgement

This thesis would not be completed without the help of many people, I would like to spend this small space to express my heartfelt thanks.

I would like to express my first and foremost gratitude to my master degree and undergraduate final year project supervisor Professor Philip Heng Wai LEONG for his guidance in these three years. Without him, I think I would not gain this great chance to enter the Research Palace after my bachelor graduation. He has taught how to be a researcher. I cannot forget for this teaching that even though my performance always disappointed him, he still gave me a second chance to do better. Thanks for his endurance and patient teaching.

I also want to appreciate my master degree co-supervisor Professor Wen Jung Li for providing me the research assistant position and other financial supports. But the most important is, he led me to experience how powerful and marvelous of the micro-electro-mechanical system (MEMS) technology.

I would like to thank to my graduation committees, Professor Kin Hong LEE and Professor Kin Hong WONG from my department, and Professor Zheru CHI from the Department of Electronic and Information Engineering in The Hong Kong Polytechnic University, for spending their valuable time to give me suggestions and comments for improving this work.

I would like to thank to Dr Guanglie ZHANG, the project manager of the digital writing instrument. He is a talented and all-rounded engineer. We co-operated together since my undergraduate final year project. I have learnt a lot from him, especially in how to be a good engineer.

I am also grateful to my team members, Mr Guangyi SHI and Mr Yilun LUO for prototype development, Mr Zhuxin DONG for testing and experiment, Ms Sze Yin KWOK for prototype design and illustration, and Ms Heidi Yee Yan WONG for market analysis. Without their support, I think I cannot finish this project

in one-man power. And also thanks to Mr Octopus Cheung Shing CHAN, one of my CMNS colleagues, to help me to draw the diagrams in my dissertation in his spare time.

I would like to give thanks to my colleagues in Centre for Micro and Nano Systems (CMNS) for bringing me a comfortable working atmosphere. I can feel we are not just working in a research centre, but we are in a big family.

I would like to give my deepest gratitude to my parents and grandma for their love and support in these few years. I feel sorry that I have spent very little time at home and always made them worry. This thesis is dedicated to them.

Last but not least, I would like to thanks my God to lead me through this research road, especially when I am tired.

"He gives strength to the weary and increases the power of the weak."
– Isaiah 40:29

Statement of Originality

The work described in this thesis was carried out at the Centre for Micro and Nano System and the Department of Computer Science and Engineering, the Chinese University of Hong Kong, between 2005 and 2007, under the supervision of Prof Philip H.W. Leong and Prof Wen J. Li in the Department of Mechanical and Automation Engineering.

The work in this thesis is entirely original except where duly referenced. In particular

- The idea of using the μIMU to implement the ubiquitous digital writing instruments is the work of Prof Wen J. Li, Prof Philip H.W. Leong and Daka Development Ltd.
- The idea of using the Kalman filtering algorithm to reduce the error in tracking position was suggested by Prof Wen J. Li and Prof Philip H.W. Leong.
- The μIMUs used to test the algorithms were designed and implemented by Dr Guanglie Zhang, Mr Yilun Luo and Mr Guangyi Shi.
- The zero velocity compensation (ZVC) algorithm is the work of Ubiquitous Computing Laboratory at the Samsung Advanced Institute of Technology
- The stroke segmentation enhanced Kalman filtering work is that of the author.
- The zero z-axis position feedback enhanced Kalman filtering scheme was developed by Dr Guanglie Zhang, Mr Gary Chun Tak Chow and the author. It is implemented by the author.
- The idea of using electromagnetic resonance (EMR) position detection board for Kalman filter feedback is the work of Prof Wen J. Li and Dr Guanglie Zhang. The implementation of the EMR position detection board is the work of Dr Guanglie Zhang, Mr Zhuxin Dong and Mr Chor Fung Chung.

The EMR position detection algorithm is the work of the author. The implementation of the combined scheme of the EMR position feedback enhanced Kalman filtering is the work of the author.

- The idea of using optical tracking by high speed camera to calibrate the μIMU-based digital writing instruments and perform the error analysis of the error compensation algorithms for the instruments is the work of Prof Wen J. Li and Dr Guanglie Zhang. The implementation of the calibration system is the work of Mr Zhuxin Dong.

- All software used for simulation, modelling and testing the algorithms in this thesis was written by the author unless otherwise stated.

The material in this thesis has not been submitted for any other degree at this or any other institution.

Contents

Abstract **i**

Acknowledgement **iii**

Statement of Originality **v**

Table of Contents **vii**

List of Figures **x**

Nomenclature **xii**

1 Introduction **1**

1.1 Motivation 1

1.2 Objectives 3

1.3 Contributions 3

1.4 Thesis Organization 4

2 A Ubiquitous Digital Writing System **5**

2.1 Introduction 5

2.2 MEMS Motion Sensing Technology 6

2.2.1 Micro-Electro-Mechanical Systems (MEMS) 6

2.2.2 Principle of a MEMS Accelerometer 6

2.2.3 Principle of a MEMS Gyroscope 7

2.3 Architecture of Ubiquitous Digital Writing System 8

2.3.1 Micro Inertial Measurement Unit (μIMU) 8

2.3.2 Data Transmission Module 10

2.3.3 User Interface Software 10

2.4 Summary 12

3 Calibration of μ-Inertial Measurement Unit **13**

3.1 Introduction 13

3.2 Sources of Error 13

3.2.1 Deterministic Errors 13
3.2.2 Stochastic Error 14
3.3 Calibration of Accelerometers 14
3.4 Coordinate Transformation with Gravity Compensation 15
3.4.1 Coordinate Transformation 16
3.4.2 Attitude Determination 18
3.4.3 Gravity Compensation 19
3.5 Summary 20

4 Zero Velocity Compensation 21
4.1 Introduction 21
4.2 Algorithm Description 21
4.2.1 Stroke Segmentation 22
4.2.2 Zero Velocity Compensation (ZVC) 22
4.3 Experimental Results and Discussion 23
4.4 Summary 24

5 Kalman Filtering 28
5.1 Introduction 28
5.2 Summary of Kalman filtering algorithm 28
5.2.1 System Model 28
5.2.2 Initialization 29
5.2.3 Time Update 32
5.2.4 Measurement Update 33
5.2.5 Stroke Segmentation 34
5.3 Summary 34

6 Error Compensation from Position Feedback 35
6.1 Introduction 35
6.2 Global Positioning System (GPS) 35
6.3 Zero z-axis Kalman Filtering 36
6.3.1 Algorithm Implementation 36
6.3.2 Experimental Results and Discussion 40
6.4 Combined Electromagnetic Resonance (EMR) Position Detection Board and μIMU 43
6.4.1 EMR Position Detection System 43
6.4.2 A Combined Scheme 44
6.4.3 Algorithm Implementation 46
6.4.4 Synchronization 50
6.4.5 Experimental Results and Discussion 50

6.5 Summary 54

7 Conclusion **55**

7.1 Future Work 56

7.1.1 Improvement in the μIMU 56

7.1.2 Combined Camera Optical Tracking and μIMU 57

7.2 Concluding Remarks 58

A Derivation of Kalman Filtering Algorithm **59**

A.1 Introduction 59

A.2 Derivation of a Priori State Estimation Equation 60

A.3 Derivation of a Posteriori State Estimation Equation 60

A.4 Derivation of a Priori Error Covariance Matrix 61

A.5 Derivation of the Optimal Kalman Gain 62

A.6 Derivation of a Posteriori Error Covariance Matrix 63

B Derivation of Process Noise Covariance Matrix **64**

Bibliography **66**

Publications **69**

List of Figures

1.1 The Digital Writing Instrument 2
1.2 The Ubiquitous Digital Writing System 3

2.1 The Working Principle of a MEMS Accelerometer 6
2.2 The Working Principle of a MEMS Gyroscope 7
2.3 System Architecture of the Ubiquitous Digital Writing System . . 8
2.4 The Prototype of the μIMU 9
2.5 A User Interface Software for Ubiquitous Digital Writing System . 11
2.6 A Detailed Block Diagram of the Processing Algorithm Module . 11

3.1 Calibration of accelerometers 15
3.2 Coordinate system of navigation and body frames 16
3.3 Coordinate transformation from body frame to navigation frame . 18

4.1 Demonstration of the Stroke Segmentation Algorithm 22
4.2 Demonstration of the Zero Velocity Compensation (ZVC) algorithm 24
4.3 Comparison between using the ZVC algorithm and without any processing to the raw data in writing a letter "S" 25
4.4 Comparison between using the ZVC algorithm and without any processing to the raw data in writing a letter "U" 26

5.1 The Kalman Filtering Algorithm 29

6.1 Random Noise in the Three-axes Accelerometers 41
6.2 Noise level of Accelerometers after using Kalman filtering algorithm with Pen-tip Compensation 41
6.3 Noise level of Accelerometers after using Kalman filtering algorithm without Pen-tip Compensation 42
6.4 Performance comparison between the zero z-axis position feedback enhanced Kalman filtering and integration without any processing 42
6.5 Working Principle of Electromagnetic Resonance Motion Detection System 44
6.6 Electromagnetic Resonance Motion Detection System 45

6.7 Signal from Electromagnetic Resonance (EMR) Position Detection System 46
6.8 A Combined Scheme of μIMU and EMR Motion Detection Board 46
6.9 Optical Calibration System 51
6.10 The performance of the μIMU/EMR motion detection board coupled system 52

Nomenclature

ΔC	Change in capacitance during force is applied
Δd	Deflection of beam
Δt	Sampling time
ϵ	Velocity residue
$\hat{a}$	Acceleration estimate
$\hat{s}$	Position estimate
$\hat{v}$	Velocity estimate
$\hat{x}^{-}k$	A priori state estimate at time instant k
$\hat{x}_k$	A posteriori state estimate at time instant k
$\mathbf{A_b}$	Acceleration vector in the body frame
$\mathbf{A_n}$	Acceleration vector in the navigation frame
$\mathbf{G}$	Gravitational constant vector in the navigation frame, i.e. $\begin{bmatrix} 0 & 0 & -g \end{bmatrix}$
μIMU	Micro-inertial measurement unit
ω_{input}	Input angular velocity
ϕ	Roll, rotation about x-axis of the body frame
ψ	Yaw, rotation about z-axis of the body frame
σ_{th}	Threshold to determine whether the pen-tip is moving
θ	Pitch, rotation about y-axis of the body frame
A	Transition matrix
a	Acceleration
$a_{b,i}$	Acceleration of the axis i in the body frame, where $i = x, y, z$
$a_{coriolis}$	Coriolis acceleration
$a_{n,i}$	Acceleration of the axis i in the navigation frame, where $i = x, y, z$
C_0	Capacitance of the unit cell measured at stationary

d_0	Separation between the planes at stationary
$E[X]$	Expectation of random variable X
F	Applied force
g	Gravitational constant
H	Measurement matrix / Observation matrix
k	Spring constant of tether
K_k	Kalman gain at time instant k
m	Mass of the beam
P_k^-	A posteriori error covariance matrix at time instant k
P_k	A posteriori error covariance matrix at time instant k
Q	Process noise covariance matrix
q_c	Process noise covariance for continue-time
R	Measurement noise covariance matrix
s	Position
t	Time
T_b^n	Direction cosine matrix (DCM)
T_ϕ	Rotation matrix about roll (ϕ)
T_ψ	Rotation matrix about yaw (ψ)
T_θ	Rotation matrix about pitch (θ)
$v_{vibration}$	Linear velocity of the vibrator
X^T	Transpose of matrix X
y_k	Measurement vector at time instant k
EMR	Electromagnetic resonance
IMU	Inertial measurement unit
ZVC	Zero velocity compensation algorithm

Chapter 1

Introduction

1.1 Motivation

The "Electronic Whiteboard" and "Digital Pen" are new paradigms in the office automation industry that may someday completely replace the computer keyboard, which is still the preferred alphanumeric human-to-computer input device. These new devices aim to capture human handwriting or drawing motions in real-time and store motion strokes for character recognition or information retrieval at a later time.

In 1964, the first graphics tablet was launched, the RAND Tablet [12], also known as the Grafacon (Graphic Converter). It makes use of electromagnetic resonance to digitize pen motion. In the next 40 years of development, many different well-developed methodologies to digitize handwriting have been proposed. Targeting business and academic institutions, ultrasonic, infrared and optical sensing are currently the most popular technologies for detecting the position of a digital pen on a large area electronic whiteboard. These systems allow users to write on specific surfaces with restricted active areas by the usage of special dry-erase pens.

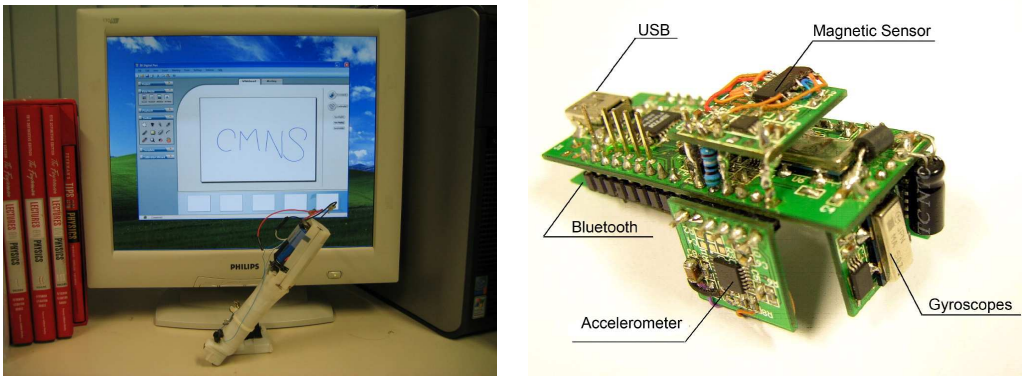
Luidia Inc. [24] and Sanford LP [33] have separately proposed systems, eBeam and mimio® respectively, that can modify a conventional whiteboard by placing a receiver in its corner. The receiver uses infrared and ultrasound technologies to translate pen movement into positions which are recorded on a computer. However, the price of the overall system is very expensive, over US$700, and the active area is limited, with maximum dimension of 2.4 $m \times 1.2$ m. Logitech [9] and Nokia [10] have released the Logitech© io^{TM}2 Digital Writing System and Nokia Digital Pen SU-1B respectively which is 3 times cheaper than the eBeam and mimio® solution. This technology comes from the Anoto Group AB which

(a) The Digital Pen (b) The Digital Eraser

Figure 1.1: The Digital Writing Instrument

uses optical detection techniques [1]. A specialized pen emits a lightwave that is deflected by patterns built onto specialized digital paper. By detecting the reflected light, the pen can be made to record its position on the paper. All existing products, including eBeam, mimio®, Logitech© io^{TM}2 and Nokia SU-1B, require special writing surfaces or attachments to function and the active area for position detection is limited.

Recently, our group developed a Ubiquitous Digital Writing Instrument to capture and record human handwriting or drawing motions in real-time based on Micro-Electro-Mechanical Systems (MEMS) motion sensing technology [51]. This system is developed from the Micro Input Devices System (MIDS) [31], which was also developed by our group. Using low-cost and small MEMS sensors, it is possible to build self-contained inertial sensors with overall system dimension of less than 1 cubic inch as shown in Figure 1.2(b). These are called micro-inertial measurement units (μIMU), which include accelerometers and gyroscopes to provide the acceleration and the angular rate of the motion. With the information provided, Lam *et al.* proposed several practical applications to use this system, such as robot controller [28, 29], wireless mouse, virtual keyboard [30, 31], game controller [47], sports science [16], etc. Using a bluetooth wireless module, the sensor units can track orientation and locomotion in real time. Recalling from high school physics that position is the double integral of acceleration, based on the acceleration and rotation in different axes of the given input device, it is possible to build a digital pen based on these inputs.



(a) The Hardware and Software Prototype (b) The μIMU with Bluetooth module

Figure 1.2: The Ubiquitous Digital Writing System

1.2 Objectives

MEMS motion technology can provide a novel solution for building a ubiquitous digital writing system. This handwriting position tracking idea was first proposed by C. Verplaetse [43] in 1996. However, to date, no successful digital pen using this idea exists on the market. This is because different sources of errors are introduced into the system during the manufacturing and assembly processes. Some are deterministic, such as sensor bias, and can be eliminated through calibration, but some are random, such as electronic and mechanical noise in the MEMS inertial sensors, and cannot be completely removed from the system. In inertial kinematics theory [42], the position is computed by the double integration of acceleration with respect to time. Hence, the main objective of this work is to propose practical techniques to minimize the error, so as to demonstrate the feasibility of MEMS accelerometer-based digital input device.

The detailed research aims were:

- Investigate existing error reduction algorithms for a MEMS accelerometer based digital input device.
- Explore a methodology to improve the position tracking accuracy of the MEMS accelerometer based digital input device.
- Demonstrate the feasibility of MEMS accelerometer-based digital input device by developing a prototype.

1.3 Contributions

This thesis presents methodologies for error reduction of accelerometer-based digital writing instruments to improve writing accuracy. The contributions of

this work are:

- A study of previous work on error reduction for a micro-inertial measurement unit (μIMU) navigation.
- An experimental study of zero velocity compensation (ZVC) algorithm.
- A novel Kalman filter based real-time error compensation methodology for a μIMU-based digital writing instrument that uses information from the z-axis of the pen to deduce absolute position.
- A novel Kalman filter based real-time error compensation algorithm that uses absolute position information from a 2-dimensional electromagnetic resonance (EMR) board.
- A systematic error analysis for the error compensation algorithms with an optical tracking based calibration system.

1.4 Thesis Organization

The thesis is structured as follows: Chapter 2 describes the system design of our ubiquitous digital writing instrument with the introduction of inertial navigation theory and MEMS motion sensing technology. Chapter 3 reviews sources of errors in using MEMS sensors and provides solutions to handle the deterministic errors in the system. Chapter 4 evaluates the Zero Velocity Compensation algorithm (ZVC), an offline error reduction algorithm proposed by other researchers. Chapter 5 gives an introduction to the Kalman Filtering algorithm which is used to handle error compensation and sensor fusion. Chapter 6 describes our error compensation algorithms which use pen-tip position feedback based on users' handwriting habits and a position feedback enhanced error compensation algorithm. Finally, a conclusion is made and future work is suggested in Chapter 7.

Chapter 2

A Ubiquitous Digital Writing System

2.1 Introduction

From elementary physics [15], position is the second integral of acceleration.

$$s = \int \int a \, dt \, dt \tag{2.1}$$

where s and a is the position and acceleration of the pen in the navigation frame respectively, and t is time. Hence, we can use accelerometers to measure the accelerations in the x, y and z-axes during writing, and then compute the position of the pen by the double integral according to Equation (2.1). After obtaining position as a function of time, handwriting can be reconstructed from the accelerometer output.

Owing to the advent of MEMS (Micro-Electro-Mechanical Systems) motion sensing technology, a micro-inertial measurement unit (μIMU), with dimensions of just $56\ mm \times 23\ mm \times 15\ mm$, can be constructed to sense the 3D motion of the pen as described before. Therefore, the overall digital writing system is very small, and it is very easy for user to take and use it at anywhere.

In this chapter, a brief introduction to the ubiquitous digital writing system will be given starting from an brief introduction to MEMS motion sensing technology. A description of the overall system architecture design is also given.

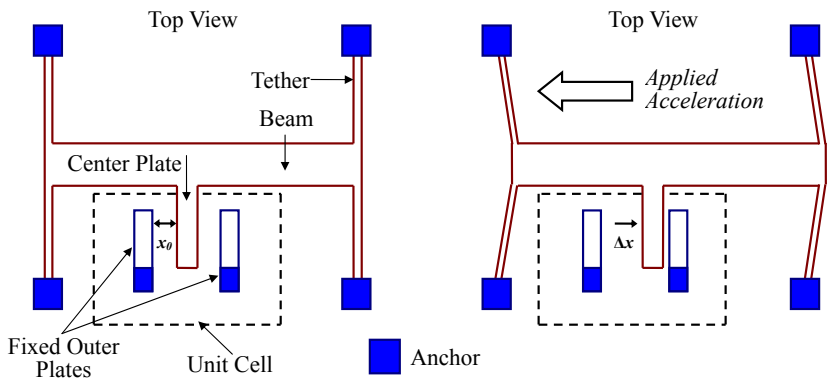


Figure 2.1: The Working Principle of a MEMS Accelerometer (modified from reference [21])

2.2 MEMS Motion Sensing Technology

2.2.1 Micro-Electro-Mechanical Systems (MEMS)

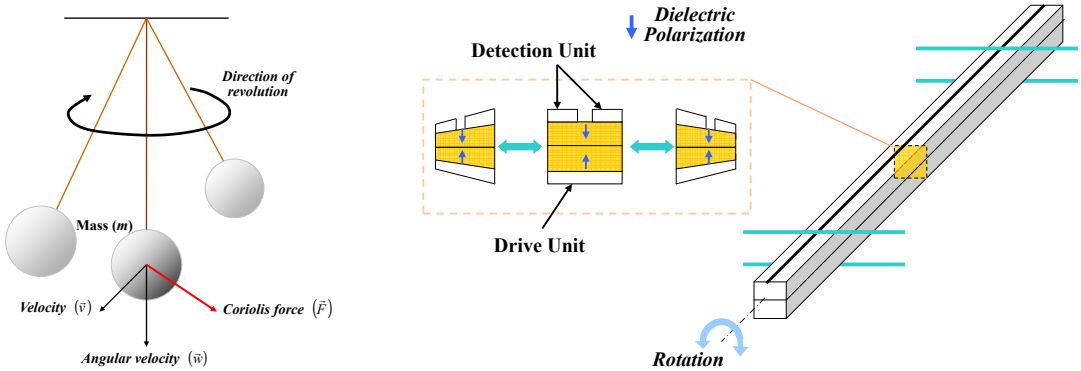
MEMS stands for Micro-Electro-Mechanical Systems which integrates mechanical units and electronic components together through micro-fabrication technology at the sub-millimeter scale. With this technology, we can build micro-structures through micro-machining and create sensors which are very small in size and suitable for the design of digital writing instruments.

There are several different methods for sensing motion. In the following subsections, we concentrate on discussing the working principles of accelerometers and gyroscopes used in our design of micro inertial measurement unit (μIMU) [32].

2.2.2 Principle of a MEMS Accelerometer

The accelerometer is an instrument which is used to measure acceleration of a target mounted object. Two Analog Device ADXL203 $\pm$1.7g dual-axis iMEMS® accelerometers [22] are used in our digital writing system. There are polysilicon springs inside the sensor which are used to suspend a beam over the surface of a silicon wafer and provide a resistance against applied force as shown in Figure 2.1. When acceleration is applied to the sensor, according to the Hooke's law, the beam deflects as described in Equation (2.2) and a differential capacitor is used to measure the distance of the beam deflected as shown in Equation (2.3). Finally, we can further measure the applied acceleration which is proportional to the deflection of the beam as described in Equation (2.4) [21, 22, 20, 32, 42, 49].

$$\Delta d = \frac{F}{k} \tag{2.2}$$



(a) Foucault's Pendulum (modified from reference [35])

(b) Structure of Gyroscope (modified from reference [2])

Figure 2.2: The Working Principle of a MEMS Gyroscope

$$\Delta C \approx C_0 \frac{\Delta d}{d_0} \tag{2.3}$$

$$a = \frac{F}{m} = \frac{k\Delta d}{m} \tag{2.4}$$

where F is the applied force, k is the spring constant of tether, Δd is deflection of the beam, a is the applied acceleration, m is mass of the beam, C_0 is the capacitance of the unit cell measured at stationary, ΔC is the change in capacitance during force is applied and d_0 is the separation between the planes at stationary.

2.2.3 Principle of a MEMS Gyroscope

The gyroscope is an instrument which is used to measure angular velocity of a target mounted object. Three Murata GYROSTAR® ENC-03M MEMS piezoelectric vibrating gyroscopes [34] are used in our digital writing system. There is a bimorph vibrator inside the instrument as shown in Figure 2.2(b) which resonates with linear velocity $v_{vibration} \cos(\Omega t)$. If the sensor is fixed to the target body rotating at rate ω_{input}, the vibrator inside experiences a time-varying Coriolis acceleration as shown in Equation (2.5) and Figure 2.2(a). The acceleration is at the same frequency as the driving acceleration, but at right angles to the vibrator velocity. Hence, the magnitude of the applied rotation about the axis orthogonal to vibrator can be determined by measuring the Coriolis acceleration generated [20, 34, 42, 49].

$$a_{coriolis}(t) = [-2\omega_{input} \otimes v_{vibration}] \cos(\Omega t) \tag{2.5}$$

where $a_{coriolis}(t)$ is the Coriolis acceleration at time instant t, ω_{input} is the input angular velocity, $v_{vibration}$ is the linear velocity of the vibrator.

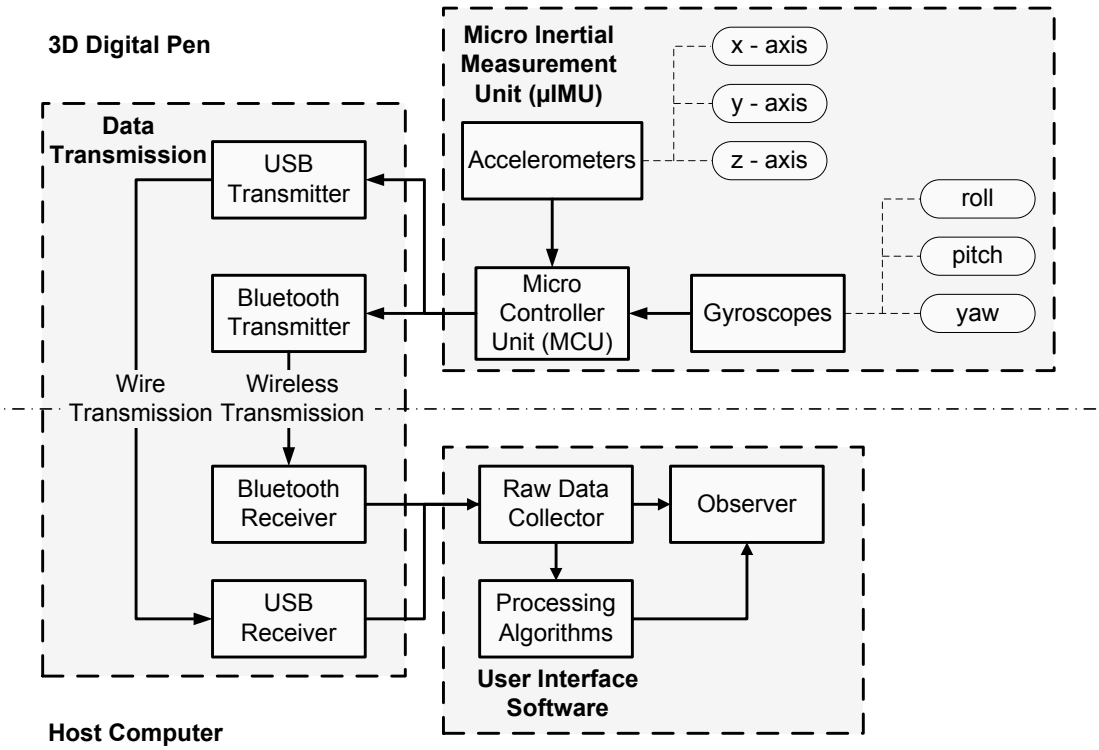


Figure 2.3: System Architecture of the Ubiquitous Digital Writing System

2.3 Architecture of Ubiquitous Digital Writing System

Figure 2.3 is a block diagram of the ubiquitous digital writing system which can be used to sense the 3D motion of a pen and reconstruct the script written by the pen on a host computer. There are three main modules which are the micro inertial measurement unit (μIMU), data transmission module and user interface software. The detailed description for each module is given in this section.

2.3.1 Micro Inertial Measurement Unit (μIMU)

Figure 2.4 shows a prototype of a μIMU. An inertial measurement unit (IMU) is used to measure acceleration and angular velocity of the attached object based on its motion. The word "micro" refers the size of overall measurement unit, the dimensions being 56 $mm \times 23$ $mm \times 15$ mm. This is very small compared to IMUs used in vehicles and aeroplanes which are typically in dimensions of 95 $mm \times 76$ $mm \times 81$ mm [23]. The μIMU can be further divided into three parts which are accelerometers, gyroscopes and microcontroller unit (MCU) as shown in Figure 2.3. The detailed functionalities of each part are described below.

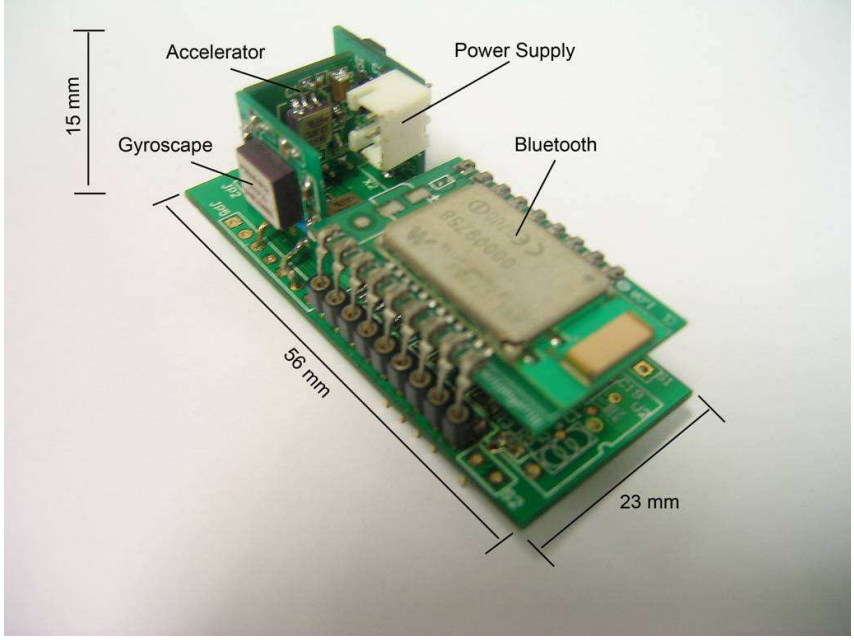


Figure 2.4: The Prototype of the μIMU

2.3.1.1 Accelerometers

Two Analog Device ADXL203 precision $\pm 1.7g$ dual-axis iMEMS® accelerometers [22] are used in our μIMU. These are aligned perpendicularly along the x, y and z-axes of the body frame of the pen. They are connected to a MCU and their output signals, which are in analog format, are digitized by an analog-to-digital converter (ADC) in the MCU which is also used to transmit the linear accelerations of the pen to the host computer through the data transmission module.

2.3.1.2 Gyroscopes

Three Murata piezoelectric vibrating single-axis gyroscopes (GYROSTAR® - ENC-03M) [34] are also used along the x, y and z-axes of the body frame of the pen to measure rotational angular velocities in three rotation angles, roll (ϕ), pitch (θ) and yaw (ψ) respectively. As with the accelerometers, they are connected to a MCU and their analog output signals digitized by the ADC to transmit rotational angular velocities to the host computer through the data transmission module.

2.3.1.3 Microcontroller Unit (MCU)

An Atmel ATmega32 MCU [8] is used to sample the 6 channels of accelerometers and gyroscopes at a rate of 200Hz and then digitize their analog outputs through its internal 10-bit ADC. The data are transmitted to the host computer through the data transmission module.

2.3.2 Data Transmission Module

A data transmission module is used to transmit sensor signal data from the μIMU to the host computer. We have two transmission modes, one is via USB and the other wireless. The host computer receives the data through a virtual serial communication port.

In the wired mode, the digital pen and host computer are connected through a USB cable. This is mainly used for system development. In actual usage, USB will be used only for recharging the pen.

The wireless mode uses two bluetooth modules, a transmitter and a receiver. Bluetooth is supported on many computer devices, especially mobile devices, such as laptop computers, personal digital assistants (PDAs), smart phones, etc. With an approximate range of 10 metres, users can use the pen anywhere in a room and transmit handwriting to a host computer.

2.3.3 User Interface Software

A user interface software as shown in Figure 2.5 is developed for the host computer. The software is divided into three main modules as shown in Figure 2.3, the raw data collector, processing algorithm and observer. The detailed functionalities of each module are described below.

2.3.3.1 Raw Data Collector

The raw data collector module is used for obtaining sensor signal data from the μIMU through its serial port. This can receive either USB or bluetooth data. This module is also used to segment acceleration and angular velocity in each axis and pass it to the processing algorithm module.

2.3.3.2 Processing Algorithm

The processing algorithm module is used for estimating the position of the pen from the raw data. It can be further separated into five parts as shown in Figure 2.6, zero bias compensation, attitude estimation, gravity compensation, coordinate transformation and position tracking. The detailed descriptions for these functional blocks are given below.

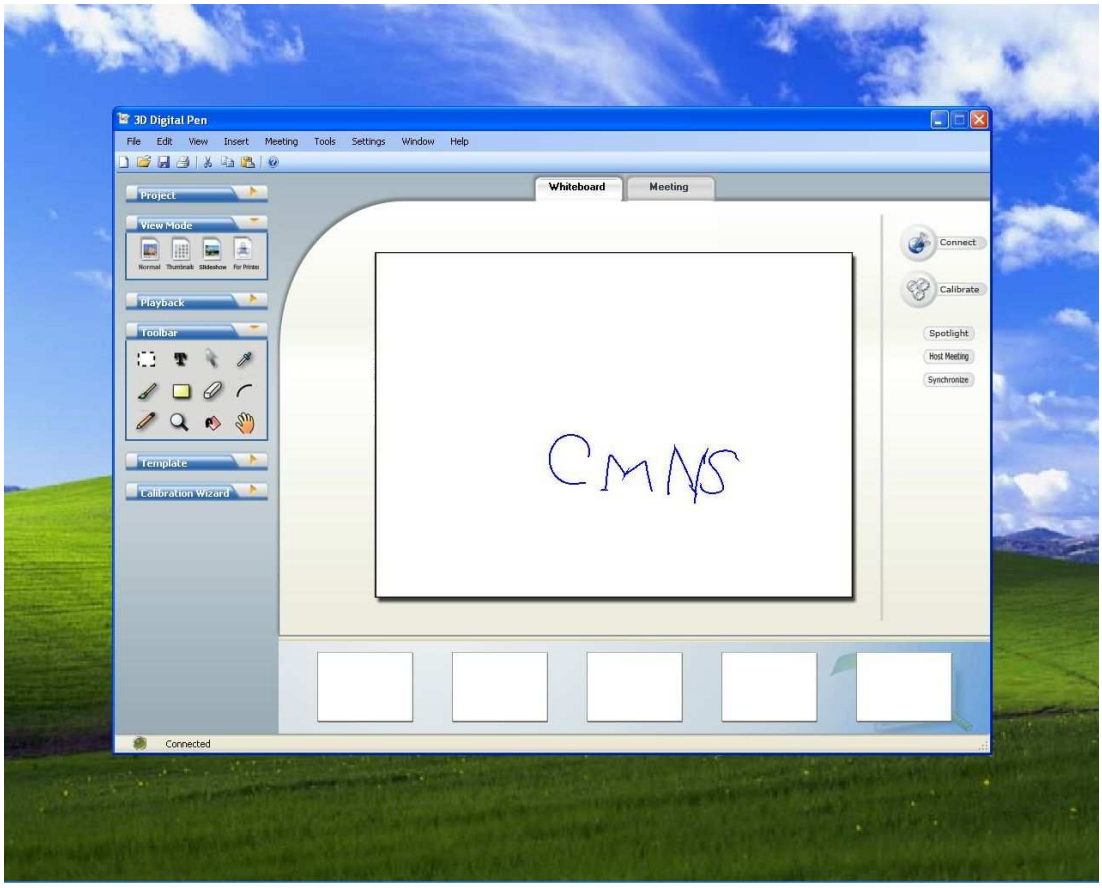


Figure 2.5: A User Interface Software for Ubiquitous Digital Writing System

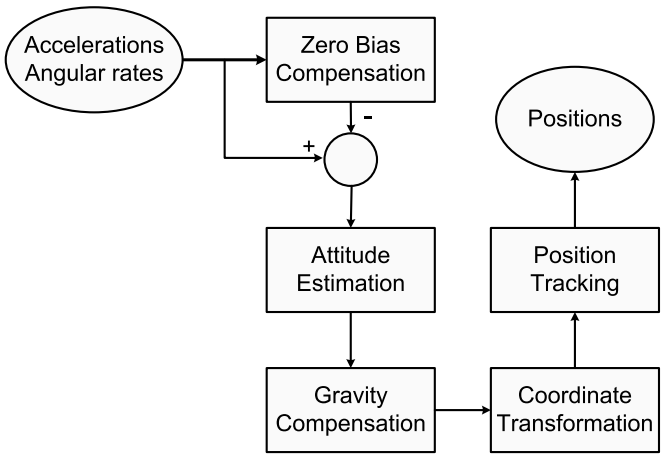


Figure 2.6: A Detailed Block Diagram of the Processing Algorithm Module

2.3.3.3 Observer

The observer module stores the position information and transfers it to the data display and data storage module. After collecting the position information, users can obtain the position information in two ways, the data display module can reconstruct the script and display it on the graphical user interface (GUI) of the software, or the data store module can save the position information with the raw data in a file for later reference.

2.4 Summary

In this chapter, we have described the overall design of a ubiquitous digital writing system which uses MEMS motion sensing technology for position tracking. We have also explained the fundamental working principle of MEMS accelerometers and gyroscopes which are key components in the system.

Chapter 3

Calibration of μ-Inertial Measurement Unit

3.1 Introduction

Owing to the double integration of Equation (2.1), a small noise in the accelerometer measurement will grow rapidly in the computed final position. Error reduction techniques are the key to a successful implementation of a digital pen. In this chapter, a calibration method and correction technique for deterministic μIMU errors is presented together with a study of accelerometer error sources.

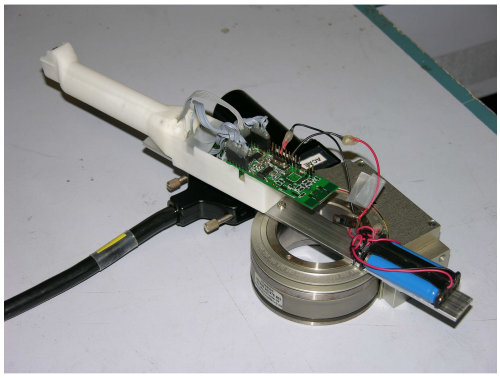
3.2 Sources of Error

In a μIMU, there are many sources of errors that can reduce its measurement accuracy. In this section, we present a study for the errors, and categorize them into two groups, deterministic and stochastic.

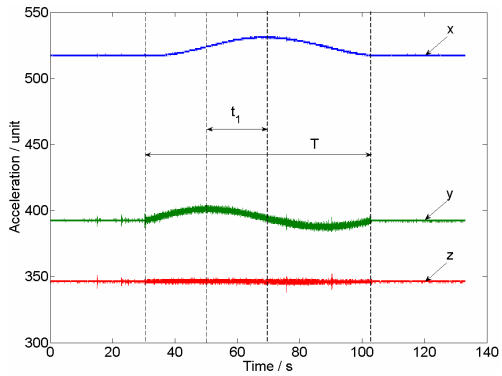
3.2.1 Deterministic Errors

Misalignment Configuration Misalignment problems can be categorized into two parts, external and internal. Since we use two 2D MEMS accelerometers to construct our μIMU, there is a risk that these two accelerometers are not perpendicular. This is known as external misalignment. Internal misalignment is due to misalignment during fabrication.

The external misalignment is easy to solve. We can use the experimental setup as shown in Figure 3.1(a), which is a constant speed rotation table, and then align the plane, which consists of two active sensing accelerometers, parallel to gravity. Owing to the orientation of the accelerometers, these two axes accelerometers



(a) The experimental setup for accelerometers alignment



(b) The experimental result of the accelerometers alignment

will pick up a sinusoid curve with 90° phase difference if they are perpendicular as shown in Figure 3.1(b). Hence, based on this phenomenon, we can test the accelerometers to determine whether they can provide this 90° phase difference, i.e. $t_1 = \frac{1}{4}T$. If not, we can supplement the remaining phase difference as a constant, i.e. $t_2 = \frac{1}{4}T - t_1$ when using the Direction Cosine Matrix (DCM) defined in Equation (3.9).

Quantization Since the analog MEMS accelerometers are digitized for the ease of processing and transmission, the signal received at the host computer will be discretized and quantization errors will result.

3.2.2 Stochastic Error

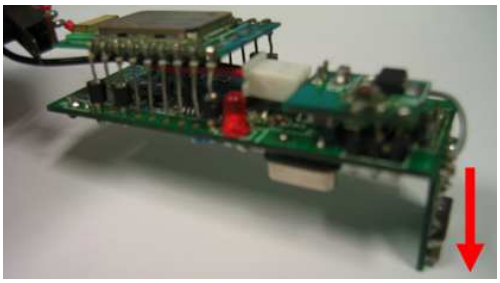
Thermal Noise Owing to Brownian motion, white noise is generated by thermal agitation due to the random charge carriers motion inside the sensors. This is also known as Johnson-Nyquist noise, and results in a noisy signal output with a root mean square value being the thermal noise equivalent acceleration (TNEA) as shown in Equation (3.1) [7, 13, 18, 26, 32, 37, 49].

$$TNEA = \sqrt{4Bk_BT} \tag{3.1}$$

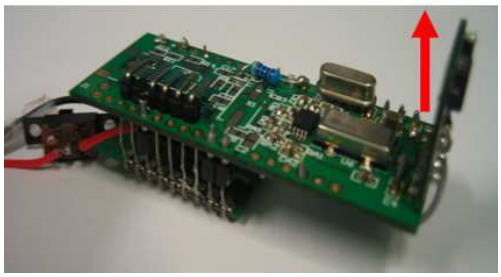
3.3 Calibration of Accelerometers

In order to provide a full range of $\pm 1.7g$ from an unsigned value, an offset voltage is added to the output [22]. In this section, we discuss how to calibrate the accelerometer to remove offset bias and acceleration gain errors.

As we know, if the μIMU is stationary, there is no external force on the μIMU except the gravity. Hence, if we obtain a maximal or a minimal output while



(c) Pointing to Earth



(d) Pointing away from Earth

Figure 3.1: Calibration of accelerometers

Axis	x	y	z
Maximum (max)	719	723	728
Minimum (min)	307	312	316
Direction ($sign$)	$-$	$+$	$+$

Table 3.1: Experimental values for accelerometer calibration

rotating the μIMU, the accelerometer must be parallel to gravity as shown in Figure 3.1. According to the coordinate system defined in Figure 3.2, we tabulate the axis tested, the maximal value (max), minimal value (min) and direction ($sign$) in Table 3.1. From max and min, we can compute the zero bias ($bias$) and the output range ($range$) with Equations (3.2) and (3.3). Hence the calibrated output is given in Equation (3.4). This assumes linearity in the accelerometer output and that zero acceleration is at the centre of the output range.

$$bias = \frac{(max + min)}{2} \tag{3.2}$$

$$range = \frac{(max - min)}{2g} \tag{3.3}$$

$$output_{calibrated} = \frac{sign\,(output_{raw} - bias)}{range} \tag{3.4}$$

In the above equations, $output_{calibrated}$ and $output_{raw}$ are the calibrated and raw accelerometer output respectively, g is the gravitational constant 9.81 ms^{-2}, and $sign$ is defined to follow the body frame coordinate system shown in Figure 3.2.

3.4 Coordinate Transformation with Gravity Compensation

Figure 3.2 shows the two coordinate systems of the digital writing instrument. One is the body frame, which represents the coordinate system aligned with the

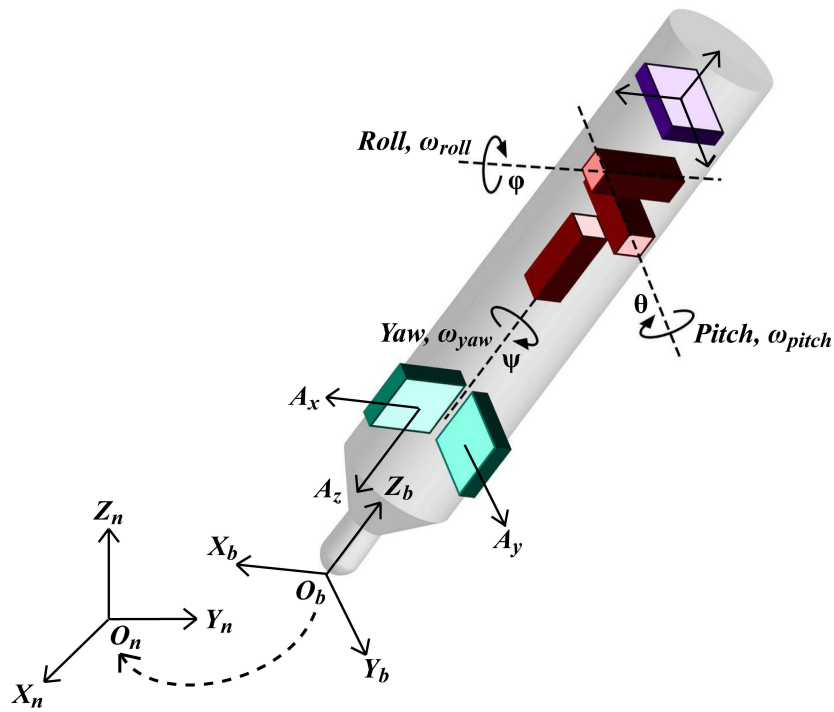


Figure 3.2: Coordinate system of navigation and body frames

configuration of the accelerometers attached on the digital writing instrument. The other is the navigation frame, which represents the coordinate system we used in the navigation tracking [5, 11, 20]. In order to reproduce the handwriting trajectory, we first need to transform the acceleration in the body frame to the navigation frame and remove the gravity for integration. In this section, we would like to have a detailed description for the coordinate transformation with gravity compensation and also with the attitude determination.

3.4.1 Coordinate Transformation

Given that the body frame b is aligned with the axes of the sensors. The attitude of the pen can be represented by the three Euler angles, yaw (ψ), pitch (θ) and roll (ϕ) and their corresponding rotation matrices $\mathbf{T}_\phi$ $\mathbf{T}_\theta$ and $\mathbf{T}_\psi$ are defined as follows:

Let T_j, where $j = \phi, \theta, \psi$, be the rotation matrix from the body frame to the navigation frame defined as follows:

$$\begin{bmatrix} a_{n,x} \\ a_{n,y} \\ a_{n,z} \end{bmatrix} = \mathbf{T_j} \begin{bmatrix} a_{b,x} \\ a_{b,y} \\ a_{b,z} \end{bmatrix} \tag{3.5}$$

where $a_{n,i}$ and $a_{b,i}$ are the acceleration with respect to the navigation frame and the body frame defined in Figure 3.2 respectively, and the axis $i = x, y, z$.

Roll (ϕ) is rotation about the x-axis of the body frame, X_b, from the body frame to the navigation frame as shown in Figure 3.3(a).

$$\mathbf{T}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{3.6}$$

Pitch (θ) is rotation about the y-axis of the body frame, Y_b, from the body frame to the navigation frame as shown in Figure 3.3(b).

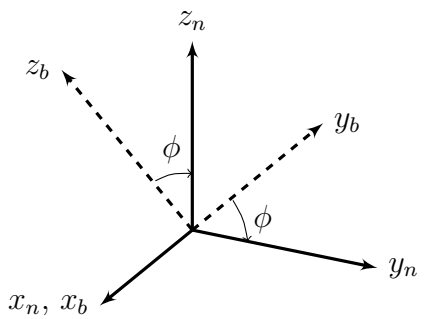
$$\mathbf{T}_\theta = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.7}$$

Yaw (ψ) is rotation about the z-axis of the body frame, Z_b, from the body frame to the navigation frame as shown in Figure 3.3(c).

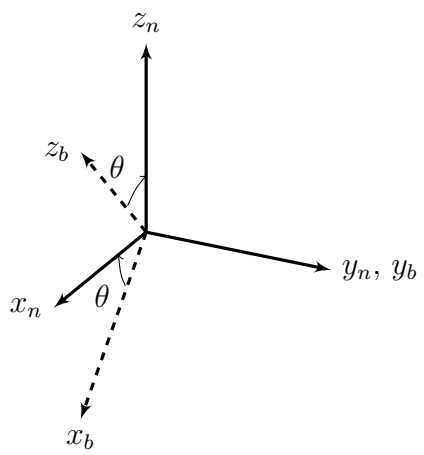
$$\mathbf{T}_\psi = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

$$\begin{aligned}
\mathbf{T}_\mathbf{b}^\mathbf{n} &= \mathbf{T}_\phi \mathbf{T}_\theta \mathbf{T}_\psi \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ \sin\phi\sin\theta & \cos\phi & -\sin\phi\cos\theta \\ -\cos\phi\sin\theta & \sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta\cos\psi & -\cos\theta\sin\psi & \sin\theta \\ \begin{pmatrix} \sin\phi\sin\theta\cos\psi \\ +\cos\phi\sin\psi \end{pmatrix} & \begin{pmatrix} -\sin\phi\sin\theta\sin\psi \\ +\cos\phi\cos\psi \end{pmatrix} & -\sin\phi\cos\theta \\ \begin{pmatrix} -\cos\phi\sin\theta\cos\psi \\ +\sin\phi\sin\psi \end{pmatrix} & \begin{pmatrix} \cos\phi\sin\theta\sin\psi \\ +\sin\phi\cos\psi \end{pmatrix} & \cos\phi\cos\theta \end{bmatrix}
\end{aligned} \tag{3.9}$$

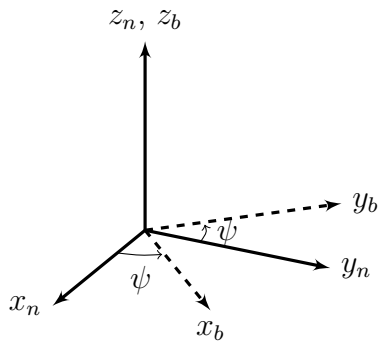
where $\mathbf{T}_\mathbf{b}^\mathbf{n}$ is also called Direction Cosine Matrix (DCM) which is used for vector transformation of the accelerations in the body frame to the navigation frame.



(a) Rotation about the x-axis of the body frame, X_b, from the body frame to the navigation frame



(b) Rotation about the y-axis of the body frame, Y_b, from the body frame to the navigation frame



(c) Rotation about the z-axis of the body frame, Z_b, from the body frame to the navigation frame

Figure 3.3: Coordinate transformation from body frame to navigation frame

$\mathbf{T}_\phi$, $\mathbf{T}_\theta$ and $\mathbf{T}_\psi$ are the rotation matrix about the roll (ϕ), pitch (θ) and yaw (ψ) respectively.

3.4.2 Attitude Determination

In Figure 3.2, we can determine the attitude of the pen when stationary because there is no external force applied to the system except the gravity. However, since the rotation plane of the angle ψ is perpendicular to gravity, we cannot calculate ψ from the accelerometer measurement alone. Thus we assume ψ to be zero through the experiment.

The acceleration in the navigation frame $\mathbf{A_n}$ is gravity $\mathbf{G}$ while the acceleration in the body frame $\mathbf{A_b}$ is the accelerometer measurement. The following equation

models the stationary situation.

$$\mathbf{G} = \mathbf{T_b^n A_b} \tag{3.10}$$

where $\mathbf{G} = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}^T$ is the gravitational constant vector in the navigation frame and $\mathbf{A_b} = \begin{bmatrix} a_{b,x} & a_{b,y} & a_{b,z} \end{bmatrix}^T$ is the acceleration in the body frame defined in Figure 3.2. This is the measured acceleration from the calibrated accelerometers without any transformation.

By the orthogonal property of the DCM and assuming $\psi = 0$, we have

$$\begin{aligned}
\mathbf{A_b} &= \mathbf{T_n^b G} = [\mathbf{T_b^n}]^T \mathbf{G} \\
\begin{bmatrix} a_{b,x} \\ a_{b,y} \\ a_{b,z} \end{bmatrix} &= \begin{bmatrix} \cos\theta & \sin\phi\sin\theta & -\cos\phi\sin\theta \\ 0 & \cos\phi & \sin\phi \\ \sin\theta & -\sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \\
&= \begin{bmatrix} g\cos\phi\sin\theta \\ -g\sin\phi \\ -g\cos\phi\cos\theta \end{bmatrix}
\end{aligned} \tag{3.11}$$

Based on Equation (3.11), we can derive the following three equations

$$\frac{a_{b,x}}{a_{b,z}} = \frac{g\cos\phi\sin\theta}{-g\cos\phi\cos\theta} = \tan\theta \tag{3.12}$$

$$a_{b,y} = -g\sin\phi \tag{3.13}$$

$$g = \sqrt{a_{b,x}^2 + a_{b,y}^2 + a_{b,z}^2} \tag{3.14}$$

and hence the equation for attitude from acceleration

$$\phi = \arcsin\left(\frac{-a_{b,y}}{g}\right) = \arctan\left(\frac{-a_{b,y}}{\sqrt{a_{b,x}^2 + a_{b,z}^2}}\right) \tag{3.15}$$

$$\theta = \arctan\left(\frac{a_{b,x}}{-a_{b,z}}\right) \tag{3.16}$$

3.4.3 Gravity Compensation

After obtaining the attitude of the pen, a coordinate transformation with gravity compensation is performed to remove the gravity bias and tilt [42].

$$\mathbf{A_n} = \mathbf{T_b^n A_b} + \mathbf{G} \tag{3.17}$$

where $\mathbf{A_n} = \begin{bmatrix} a_{n,x} & a_{n,y} & a_{n,z} \end{bmatrix}^T$ is the acceleration in the navigation frame, defined in Figure 3.2.

3.5 Summary

In this chapter, we described sources of errors in the μIMU, and also discussed how to calibrate the accelerometer, to determine the attitude of the pen in a stationary situation. With attitude information, we can perform a coordinate transformation with gravity compensation, and remove deterministic errors to obtain a more accurate acceleration measurement.

Chapter 4

Zero Velocity Compensation

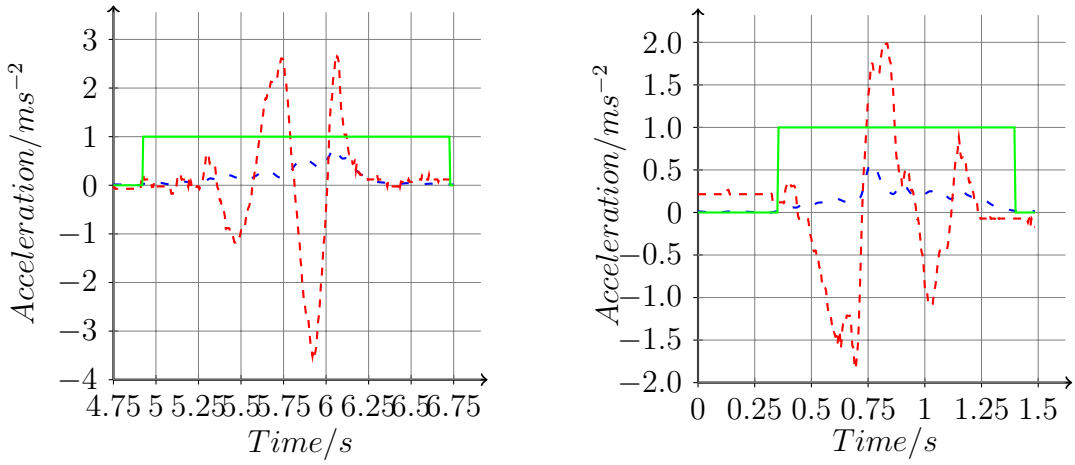
4.1 Introduction

As discussed in Chapter 3, the sources of error in the μIMU can be separated into two main parts, deterministic and stochastic. We have discussed techniques to tackle the deterministic error. In this chapter, we concentrate on methods to reduce stochastic error.

Recently, Samsung researchers proposed a method [3, 48], which is modified from the position refinement algorithm proposed by Frank [17]. The idea makes use of the fact that velocity and acceleration should be zero when the pen is stationary. With this assumption, we can consider the velocity residue during pauses as an error propagated from acceleration. It can then be compensated to reduce the effect of the positional drift. This algorithm is known as Zero Velocity Compensation (ZVC).

4.2 Algorithm Description

In this section, the zero velocity compensation (ZVC) algorithm is described in detail. There are two major parts, stroke segmentation and zero velocity compensation. First, we segment the raw accelerometer measurement of the handwriting into moving or stopped states. We record the start and stop times of the strokes. Then, with the assumption of zero velocity and acceleration when stopped, we update the velocity and acceleration, and estimate the position of the digital writing instrument. The detailed implementation will be described in the following sub-section.



(a) Stroke Segmentation of Letter "S"

(b) Stroke Segmentation of Letter "U"

Figure 4.1: Demonstration of Stroke Segmentation Algorithm. Dotted lines are the standard deviation of acceleration, dashed lines are input acceleration, and solid lines represent the stroke segmented region.

4.2.1 Stroke Segmentation

Stroke segmentation is used to separate pen-tip motion from it being stationary. So Bang *et al.* [3] proposed a very simple method to do the stroke segmentation, which uses the standard deviation of acceleration in the body frame $|\mathbf{A_b}|$, $\sigma^S_{|\mathbf{A_b}|}(k)$, within an S-sized sample window at time instant k to determine whether the pen-tip is moving by comparing to a threshold (σ_{th}) as described in Algorithm 4.1. If it is less than σ_{th}, we consider the pen to be at rest.

Algorithm 4.1 Stroke Segmentation

```
if σ^S_|A_b|(k) < σ_th then
    Stationary
else
    Moving
end if
```

4.2.2 Zero Velocity Compensation (ZVC)

Using stroke segmentation, we determine when the pen is stationary. If the pen is stationary, the velocity and acceleration should be zero. However, owing to the existence of errors in accelerometer measurements, the computed velocity may be non-zero. This velocity residue propagates to the position estimate through the integration process and causes the positional drift problem, which is a problem in the inertial navigation tracking that the computed position moves even if the target is stopped. This phenomenon occurs because the position is computed by double integration of the acceleration, and hence very sensitive to the noise in

the accelerometer measurement as the noise propagated to the position rapidly through the integration process. Bang *et al.* [3] proposed to reduce this velocity residue with a linear model as in Equation (4.1). The velocity and position estimates are updated as in Algorithm 4.2.

$$\hat{a}_{n,k} = a_{n,k} - \frac{v_{n,k_2} - v_{n,k_1}}{k_2 - k_1}\frac{1}{\Delta t}, k_1, k_2 \in k \tag{4.1}$$

where k_1 and k_2 are the time instants before and after writing respectively, (i.e. $k_1 < k < k_2$ is the writing interval), ϵ is the velocity residue, i.e. $v_{n,k_2} - v_{n,k_1}$, $\hat{s}_{n,k}$, $\hat{v}_{n,k}$ and $\hat{a}_{n,k}$ are the compensated position estimate, velocity estimate and acceleration, Δt is the sampling time of the accelerometer and the subscripts n and k mean the quantity is in navigation frame (n) at time instant k.

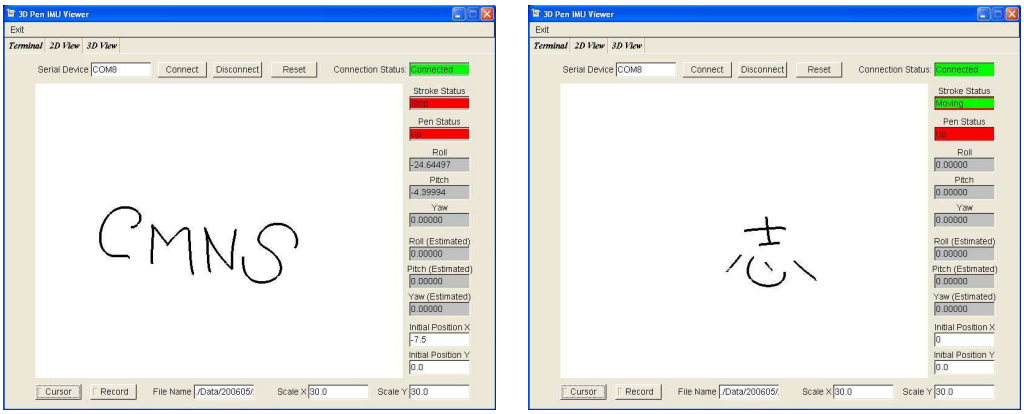
Algorithm 4.2 Zero Velocity Compensation (ZVC)

1: $v_{n,k} = v_{n,k-1} + a_{n,k}\Delta t$
2: **if** $k_1 < k < k_2$ **then**
3: $\quad \epsilon = v_{n,k_2} - v_{n,k_1}$
4: $\quad \hat{a}_{n,k} = a_{n,k} - \epsilon\frac{1}{k_2-k_1}\frac{1}{\Delta t}$
5: $\quad \hat{v}_{n,k} = \hat{v}_{n,k} - \epsilon\frac{k-k_1}{k_2-k_1}$
6: $\quad \hat{s}_{n,k} = \hat{s}_{n,k-1} + \hat{v}_{n,k}\Delta t + \hat{a}_{n,k}\Delta t^2$
7: **else**
8: $\quad \hat{v}_{n,k} = \hat{a}_{n,k} = 0$
9: **end if**

4.3 Experimental Results and Discussion

In order to evaluate the performance of the ZVC algorithm, several experiments were carried out. To simplify the experiment and concentrate the discussion on the position tracking performance of the algorithm, the attitude is assumed to be fixed throughout the experiments discussed in this chapter. Several letters were written at normal speed and can be successfully displayed on the user's computer as shown in Figure 4.2.

For the ZVC algorithm, no external device is required for error compensation. However, by comparing the performance between ZVC and integration without any processing in Figures 4.3(e) and 4.4(e), it is obvious that the scripts reproduced by ZVC are more similar to the actual one. It is because ZVC takes velocity to be the compensation reference input. If there is any error between two reference points, the velocity estimate is corrected during each pause in writing. Hence, it can reduce the amount of noise propagated to the position estimate.



(a) "CMNS" (b) a Chinese Character

Figure 4.2: Demonstration of the Zero Velocity Compensation (ZVC) algorithm

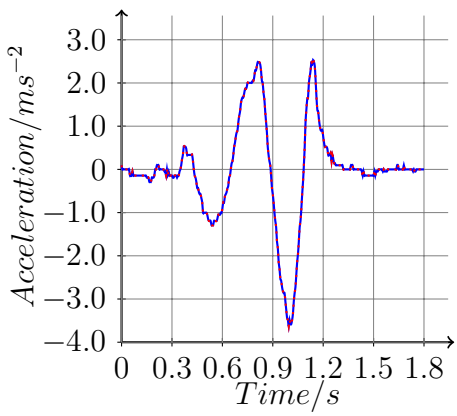
For integration without any processing, the error in the acceleration propagated to velocity and position. This results in positional drift and also velocity residue defined in this chapter. The existence of the velocity residue is evidence to justify the use of ZVC.

However, there are some disadvantages in using the ZVC algorithm. First, the time between pauses cannot be too long. The compensation process in ZVC assumes there is a constant error in acceleration, but it should be approximately zero mean Gaussian distributed in actuality. The velocity residue may not increase with time linearly. For a long time process, this model cannot determine the amount of error in acceleration at each time instant, and error in position still increases with time through the integration process. Hence, extra pauses are required to improve the performance, but this may not be acceptable to the user, especially for cursive writing.

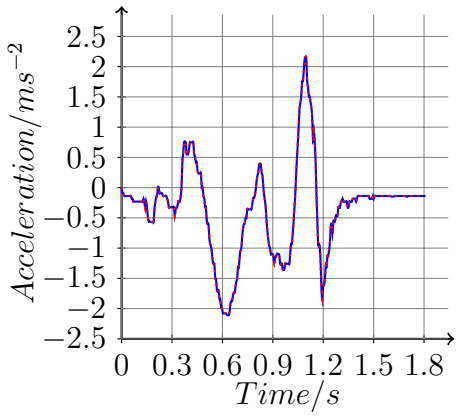
Finally, ZVC does not fulfill the real time display requirement. Bang *et al.* proposed this method for offline processing of handwriting recognition for mobile phones. Users are expected to wait until the stroke is finished. In our digital writing system, this approach may not be suitable. Since the latency time for each processing depends on the writing time for each stroke, the display latency varies considerably. Users expect to see the handwriting within a certain period, and hence this algorithm has some limitations for our real-time writing system.

4.4 Summary

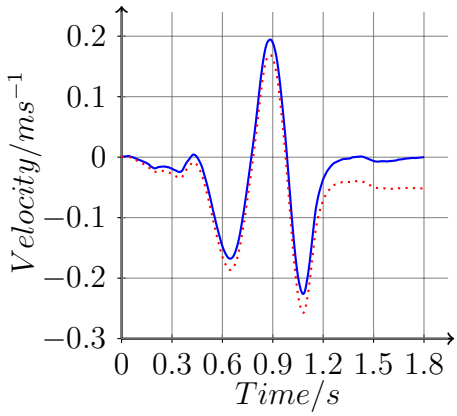
In this chapter, we have demonstrated and evaluated the ZVC algorithm. We can reduce the positional drift problem by compensating the velocity residue.



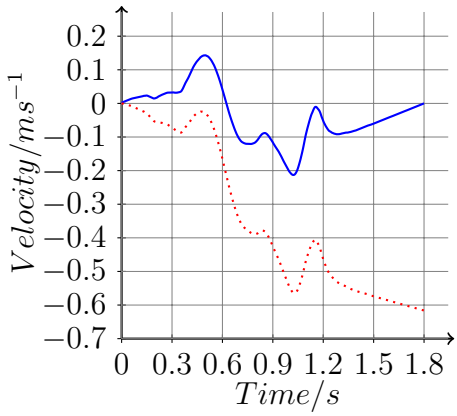
(a) Acceleration of Letter "S" in the x-axis



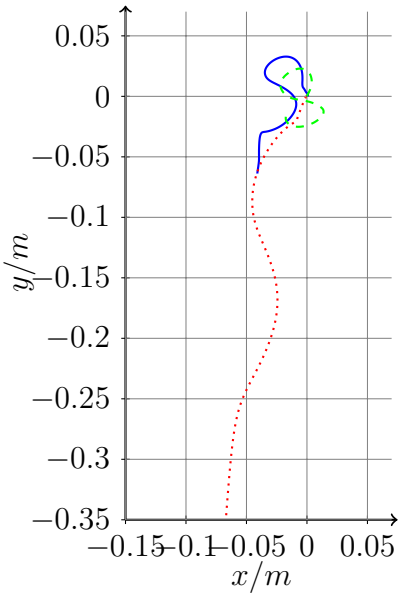
(b) Acceleration of Letter "S" in the y-axis



(c) Velocity of Letter "S" in the x-axis

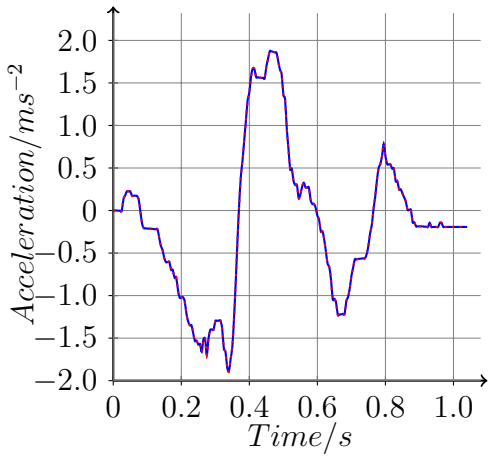


(d) Velocity of Letter "S" in the y-axis

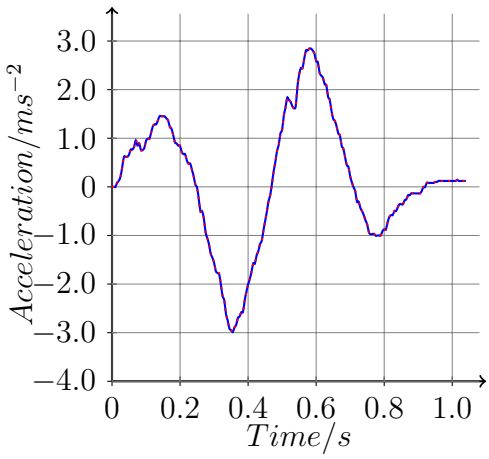


(e) Position of Letter "S"

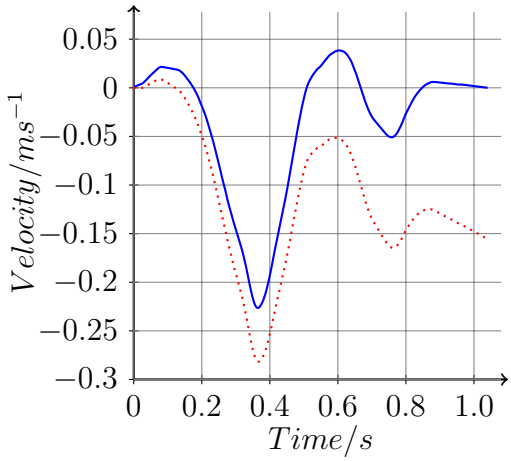
Figure 4.3: Comparison between using the ZVC algorithm and without any processing to the raw data in writing a letter "S". Solid lines are ZVC corrected whereas dotted lines are obtained by integration of the raw acceleration data. Dashed lines show the actual letter.



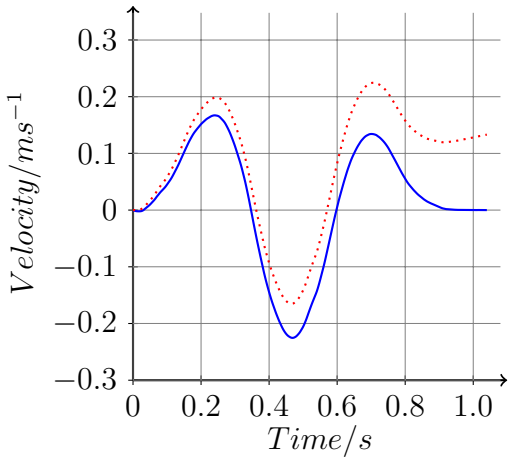
(a) Acceleration of Letter "U" in the x-axis



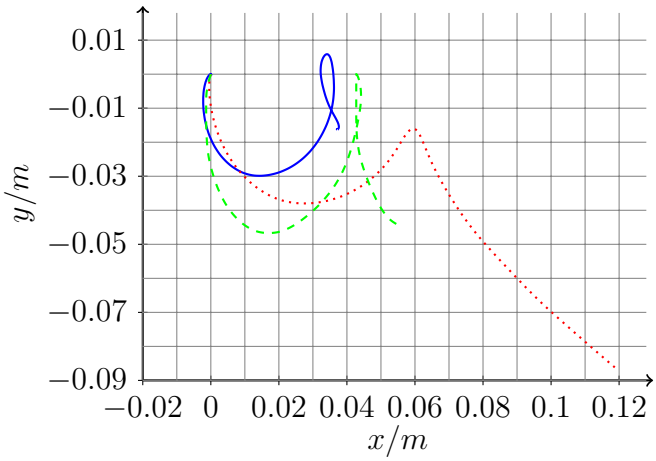
(b) Acceleration of Letter "U" in the y-axis



(c) Velocity of Letter "U" in the x-axis



(d) Velocity of Letter "U" in the y-axis



(e) Position of Letter "U"

Figure 4.4: Comparison between using the ZVC algorithm and without any processing to the raw data in writing a letter "U". Solid lines are ZVC corrected whereas dotted lines are obtained by integration of the raw acceleration data. Dashed lines show the actual letter.

The advantage of using this algorithm is that no external reference is required. However, there are several disadvantages. First, it is only useful for short durations. Frequent stopping is required for stroke segmentation, but this may not be acceptable to users. Last but not least, the output is not available until a stroke is completed. Since we do not know how long a stroke is, it cannot fulfill the real-time display requirement.

Chapter 5

Kalman Filtering

5.1 Introduction

In the field of navigation tracking, the Kalman filter is a well-known algorithm to handle the separation of probabilistic noise and state estimation of the system. In 1960, Rudolf E. Kalman published an adaptive discrete recursive estimation algorithm [27] which is based on the state-space system model to give an optimal solution to the Weiner filter, proposed by Norbert Wiener [45] in 1949. The Wiener filter is an optimal least mean square error filter to remove additional noise from a signal, given their auto-correlation and cross-correlation functions. The filter is linear and stationary. Comparing the Kalman filter to the Wiener filter, stationarity is no longer required, and an optimal solution can also be obtained for a time-varying system because the Kalman filter can adaptively update the system response to the measurement input. With the advent of digital computers, the discrete recursive state-space Kalman filtering algorithm [39] has become one of the key algorithms used in aerospace and vehicle navigation.

In this chapter, we provide a summary of Kalman filtering algorithm and give its implementation to handle the position estimation in our ubiquitous digital writing instrument system.

5.2 Summary of Kalman filtering algorithm

5.2.1 System Model

In modern control theory, transformation of process state is represented as a linear stochastic difference equation as shown in Equation (5.1). The measurement model that describes the relationship between the process state and the measurements is represented as a linear expression as shown in Equation (5.2).

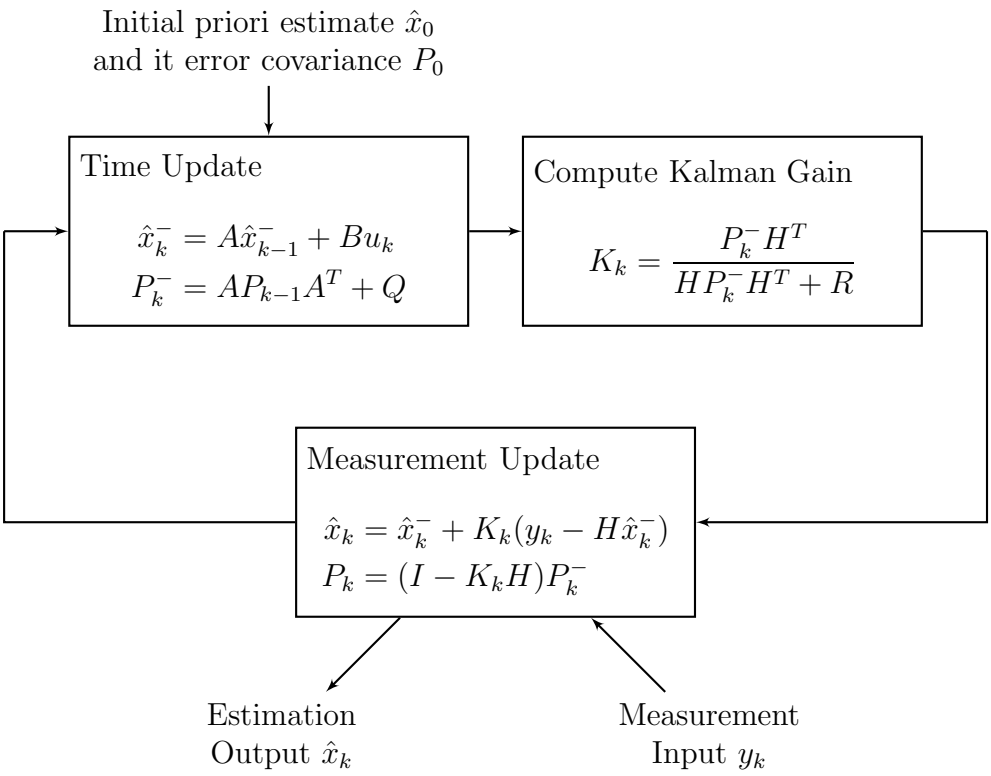


Figure 5.1: The Kalman Filtering Algorithm

Process Model:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{5.1}$$

Measurement Model:

$$y_k = Hx_k + v_k \tag{5.2}$$

where x_k is the state of the linear system, k is the time index, u is a control input to the system, y_k is the measurement input, and w and v are the random variables represent the process and measurement noise respectively, and H is the measurement matrix. For position estimation, the state propagation is the double integration of acceleration to position, hence the control input matrix, B, is not used and can be assigned to zero. Finally, A is the state transition matrix which is used to propagate the system state from the previous time instant, $k-1$ to the current time instant k. For a detailed derivation, refer to Appendix A.

5.2.2 Initialization

This stage is used to initialize the state estimate and error covariance.

System State Estimate Initialization In order to use the Kalman filter to estimate position, we need to initialize the state estimate vector ($\hat{x}$) and the error covariance matrix (P). For the inertial navigation applications, $\hat{x}$ includes the

estimate of the position ($\hat{s}_i$), velocity ($\hat{v}_i$) and acceleration ($\hat{a}_i$) of the pen in the 3-axes of the navigation frame as shown in Equation (5.3), where i represents the axis of the quantity.

$$\hat{x} = \begin{bmatrix} \hat{s}_{n,x} & \hat{v}_{n,x} & \hat{a}_{n,x} & \hat{s}_{n,y} & \hat{v}_{n,y} & \hat{a}_{n,y} & \hat{s}_{n,z} & \hat{v}_{n,z} & \hat{a}_{n,z} \end{bmatrix}^T \tag{5.3}$$

We first assume that the initial position of the pen is at the origin $(0, 0, 0)$, and initialize the system state as a 9×1 zero column vector as shown in Equation (5.4).

$$\hat{x}_0 = \mathbf{0}_{9\times 1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.4}$$

Error Covariance Initialization In order to obtain an optimal estimate from the noisy acceleration measurement, we define the covariance of the error residue of the actual system state to the estimated one, $E\left[(x - \hat{x}) \cdot (x - \hat{x})^T\right]$, to be an error covariance matrix, P, which is initialized as follows and updated later.

$$P_0^- = E\left[(x - \hat{x}) \cdot (x - \hat{x})^T\right] = \mathbf{I}_9 \tag{5.5}$$

$E[X]$ is the expectation of random variable X and $\mathbf{I}_9$ is a 9×9 identity matrix.

State Transition Matrix Recall the position (s) and velocity (v) actually in each axis is doubly integrated from acceleration (a). Hence the state transition matrix can be described as follows:

$$s_{n,i,k} = s_{n,i,k-1} + v_{n,i,k-1}\Delta t + \frac{1}{2}a_{n,i,k-1}\Delta t^2 \tag{5.6}$$

$$v_{n,i,k} = v_{n,i,k-1} + a_{n,i,k-1}\Delta t \tag{5.7}$$

where $s_{n,i,k}$, $v_{n,i,k}$ and $a_{n,i,k}$ are the position, velocity and acceleration in axis i of the navigation frame at time instant k respectively and Δt is the sampling time of the accelerometer.

By making use of the state-space model, we can model the above equation according to Equation (5.1)

$$x_{n,i,k} = A_i x_{n,i,k-1} \tag{5.8}$$

where $x_{n,i,k}$ is defined as $x_{n,i,k} = \begin{bmatrix} s_{n,i,k} & v_{n,i,k} & a_{n,i,k} \end{bmatrix}^T$, $i = x, y, z$, and A_i is the state transition matrix for axis i.

The state transition matrix for axis i, A_i is defined as follows:

$$A_i = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{5.9}$$

where Δt is the sampling period of the accelerometer.

The overall state transition matrix for the overall system state defined in Equation (5.3) is combined for the three axes.

$$A = \begin{bmatrix} A_x & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & A_y & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & A_z \end{bmatrix} \tag{5.10}$$

where $\mathbf{0}_3$ is the 3×3 zero matrix

$$\mathbf{0}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.11}$$

Measurement Matrix In order to perform the measurement update in the Kalman filter, a measurement input vector y_k at time instant k and a measurement matrix H should be defined in advance. The measurement matrix H, also called as observation matrix, is used to map the measurement input vector y_k with the corresponding state estimate variables to compute the error in state estimation during time update with the measurement input. The error vector is defined as $y_k - H\hat{x}_k^-$, which is called innovation, to multiply with the Kalman gain to update the state estimate in measurement update phase. Hence, the definition of the measurement matrix depends on the measurement input provided.

Process Noise Covariance Through the integration, the accelerometer noise will be propagated to its integrals. Process noise covariance matrix, Q_{k-1}, at time instant $k-1$ is defined and initialized as follows to allow the Kalman filter to estimate and reduce the effect of the noise propagated to the velocity and position and obtain an estimate. Since the 3-axes accelerometers are orthogonal,

zeros are used in the entries which are not defined in Q_i.

$$\begin{aligned} Q &= E[w \cdot w^T] \\ &= \int_0^{\Delta t} e^{A\tau} Q(t) e^{A^T \tau} d\tau \\ &= \begin{bmatrix} Q_x & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & Q_y & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & Q_z \end{bmatrix} \end{aligned} \tag{5.12}$$

where τ is time, $\mathbf{0}_3$ is the 3×3 zero matrix as defined in Equation (5.11). Q_i defined in Equation (5.13) is the process noise covariance matrix in axis i. For a detailed derivation, refer to Appendix B.

$$Q_i = \begin{bmatrix} \frac{1}{20} q_c \Delta t^5 & \frac{1}{8} q_c \Delta t^4 & \frac{1}{6} q_c \Delta t^3 \\ \frac{1}{8} q_c \Delta t^4 & \frac{1}{3} q_c \Delta t^3 & \frac{1}{2} q_c \Delta t^2 \\ \frac{1}{6} q_c \Delta t^3 & \frac{1}{2} q_c \Delta t^2 & q_c \Delta t \end{bmatrix} \tag{5.13}$$

Measurement Noise Covariance The measurement noise covariance matrix R is a description of the noise distribution in the measurement input provided. This can be defined from measuring the sensors noise through experiments.

5.2.3 Time Update

This stage is used to propagate the state estimate and error covariance based on the information at the previous time instant.

A Priori State Estimate Propagation The state estimate is propagated through the transition matrix.

$$\hat{x}_k^- = A\hat{x}_{k-1} \tag{5.14}$$

where $\hat{x}_k^-$ is the a priori state estimate at time instant k, $\hat{x}_{k-1}$ is the a posteriori state estimate at time instant $k-1$. A is the state transition matrix defined in Equation (5.10).

A Priori Error Covariance Propagation The error covariance is propagated through the transition matrix and the process noise covariance added.

$$P_k^- = AP_{k-1}A^T + Q_{k-1} \tag{5.15}$$

where P_k^- is the a priori error covariance matrix at time instant k, P_{k-1} is the a posteriori error covariance matrix at time instant $k-1$, and Q is the covariance of the process noise.

5.2.4 Measurement Update

This stage is used to update the state estimate and the error covariance based on the measurement input.

Optimal Kalman Gain Computation The Kalman gain, K_k, is used to update the state estimate vector, $\hat{x}$, from the innovation residue vector which is defined as the residue of the actual measurement to the estimated one, $y_k - H\hat{x}_k^-$. The computation of the optimal Kalman gain is defined as shown in Equation (5.16). It depends on the a priori process noise covariance, P_k^- and measurement noise covariance, R. The gain is an acceptability measure of the measurement input, y_k, and the a priori error covariance, P_k^-, in the update of the a posteriori state estimate, $\hat{x}_k$, and the a posteriori error covariance, P_k. This is the key to the Kalman filtering algorithm.

$$K_k = \frac{P_k^- H^T}{H P_k^- H^T + R} \tag{5.16}$$

where K_k is the optimal Kalman gain at time instant k, and P_k^- is the a priori error covariance matrix at time instant k.

A Posteriori State Estimate Propagation After computing the Kalman gain, K_k, the system state estimate, $\hat{x}$, is updated by the measurement input from the 3-axis accelerometers in the μIMU. Since the measurement is first acquired in the body frame of the digital pen under the effect of gravitational force, compensation with the coordinate transformation as described in section 3.4 should be first performed. The optimal Kalman gain also influences the update of the state estimate.

$$\hat{x}_k = \hat{x}_k^- + K_k \left(y_k - H\hat{x}_k^- \right) \tag{5.17}$$

where y_k is the acceleration measurement input at time instant k.

A Posteriori Error Covariance Propagation Besides the a posteriori state estimate propagation, the update of the error covariance matrix, P, also uses the optimal Kalman gain, K_k.

$$P_k = (I - K_k H)\, P_k^- \tag{5.18}$$

5.2.5 Stroke Segmentation

The Kalman filtering algorithm is a form of least mean square error algorithm. Its performance is highly affected by the measurement input. In the stationary state, it is obvious that there should be no motion and acceleration, however, noise in the accelerometer will cause the filter to incorrectly estimate and cannot result in zero acceleration during stationary region that induces the positional error. We use stroke segmentation, as described in Section 4.2.1 to identify stationary conditions. If there is writing, we update the Kalman filter by using the recursive equations as mentioned before without any modification. However, if stationary, we update the Kalman filter by directly setting the acceleration and velocity in the system state estimate to zero. Hence, there is no positional drift during pauses, and this helps to give a more accurate position estimate.

5.3 Summary

In this chapter, we described the Kalman filtering algorithm and demonstrated how it can be used for navigational tracking from accelerometer measurements. In future chapters, Kalman filtering will be used for error correction of accelerometer data.

Chapter 6

Error Compensation from Position Feedback

6.1 Introduction

In Chapter 4, we have covered noise reduction to improve the performance in position tracking, based on users' handwriting habits. However, the performance of position estimation is not robust and positional drift exists for the methods proposed. This is because the ZVC algorithm mentioned before do not use any position reference to correct the position estimate, hence the noise in accelerometer still propagates to position.

In vehicles and aeroplanes, the positional drift problem is also faced, but a global positioning system is used to assist the navigation. This provides absolute position information to improve the system performance through feedback. In this chapter, we will investigate the feasibility of building a similar position feedback system for a digital pen system to improve the accuracy of the tracked position.

For the sake of concentrating on the position estimation problem, all the experiments are carried out under an assumption that there is no rotation change in any axis during writing and in particular, there is no change in yaw, about the normal to the horizontal plane.

6.2 Global Positioning System (GPS)

It is now common to find a global positioning system (GPS) in vehicles and aeroplanes. GPS receives radio frequency signal from three or more GPS satellites in space. By measuring time delay between transmission and reception of the

signal in the receiver, the receiver can compute the altitude and position of the target object.

Besides that, the short-term positional errors from the inertial navigation systems (INS) are relatively small, but the accuracy in position calculation degrades without bound over time. In contrast, the GPS cannot provide high frequency updates of position, but the position from GPS will not drift away with time. The Kalman filtering algorithm is a recursive optimal least mean square error estimator, so that the increase in the measurement information will improve the accuracy of the position estimate. The Kalman filter can use statistical information about errors in both subsystems, fuse them together to take advantage of their respective advantages [20]. Therefore, many navigation systems use GPS position information as a second input for a Kalman filter.

Unfortunately, GPS is not practical for our digital writing instrument. Firstly, the writing instrument is mainly designed for indoor use, and the GPS signal will be blocked by the buildings. Even if we use the system outdoors, the resolution of GPS system is several metres [46], and cannot provide sufficient accuracy to improve pen position tracking, which requires accuracy to about 1 centimeter or less. An alternative system is needed to obtain position information to provide error feedback for the Kalman filter.

6.3 Zero z-axis Kalman Filtering

The accuracy of the Kalman filter can be increased by introducing more measurement information [20]. Most handwriting is done on a two dimensional surface; hence we can make use of the constraint that the position and velocity of the z-axis in the navigation frame in Figure 3.2 is always zero when the pen-tip touches the writing surface. This is used as a supplementary input to the Kalman filter and allows an error to be derived and feedback to be used. We also note that the pen is oblique to the navigation frame during writing so a 3-axes accelerometer is not orthogonal to the navigation frame z-axis and hence will project the errors onto the navigation frame z-axis.

6.3.1 Algorithm Implementation

Using these ideas, we modify the Kalman filtering algorithm discussed in Chapter 5. The variables used are the state vector (x_k), measurement vector (y_k), measurement matrix (H), transition matrix (A) and the process noise covariance

matrix (Q). The control-input matrix (B) is neglected since there is no control vector (u_k). Finally, the measurement noise covariance matrix (R) depends on the covariance of the noise in the measurement input (y).

System State Estimate Initialization As for the Kalman filtering in Chapter 5, we first need to define the state estimate vector ($\hat{\mathbf{x}}_\mathbf{k}$) as in Equation (6.1).

$$\hat{\mathbf{x}}_\mathbf{k} = \begin{bmatrix} \hat{\mathbf{x}}_{\mathbf{n,x,k}} & \hat{\mathbf{x}}_{\mathbf{n,x,k}} & \hat{\mathbf{x}}_{\mathbf{n,x,k}} & \hat{\mathbf{a}}_{\mathbf{b,k}} \end{bmatrix}^T \tag{6.1}$$

$$\hat{\mathbf{x}}_{\mathbf{n,i,k}} = \begin{bmatrix} \hat{s}_{n,i,k} & \hat{v}_{n,i,k} & \hat{a}_{n,i,k} \end{bmatrix}^T \tag{6.2}$$

$$\hat{\mathbf{a}}_{\mathbf{b,k}} = \begin{bmatrix} \hat{a}_{b,x,k} & \hat{a}_{b,y,k} & \hat{a}_{b,z,k} \end{bmatrix}^T \tag{6.3}$$

where k is the time index, $\hat{\mathbf{x}}_\mathbf{k}$ is the state estimate vector at time instant k, $\hat{s}_{n,i,k}$, $\hat{v}_{n,i,k}$ and $\hat{a}_{n,i,k}$ are the position, velocity and acceleration in the i-axis of the navigation frame at time instant k, and $\hat{a}_{b,i,k}$ is the acceleration in the i-axis of the body frame at time instant k, where $i = x, y, z$.

This is different to the state estimate vector defined in Equation (5.3), as it takes twelve inputs. An extra three accelerations in the x, y, z-axis of the body frame are included. The reason for keeping accelerations in both body and navigation frames is that we would like to distribute the error estimate from the zero z-axis position and velocity to the other two axes through a coordinate transformation.

Measurement Matrix Besides the 3-axes accelerometers used to estimate the system state, the zero z-axis velocity ($v_{n,z,k}$) and position ($s_{n,z,k}$) are also given as a measurement input in the Kalman filter. Hence, in Equation (6.4), the measurement input vector (y_k) is defined to consist of acceleration in the body frame, z-axis position and velocity in the navigation frame at time instant k. The measurement matrix H is defined in Equation (6.5), which is used to map the inputs $s_{n,z,k}$, $v_{n,z,k}$, $a_{b,x,k}$, $a_{b,y,k}$, $a_{b,z,k}$ to the corresponding elements of the system state $\hat{x}_k$ defined in Equation (6.1), i.e. the $7th$, $8th$, $10th$, $11th$ and $12th$ entries in the system state $\hat{x}_k$ mentioned.

$$\mathbf{y_k} = \begin{bmatrix} a_{b,x,k} & a_{b,y,k} & a_{b,z,k} & s_{n,z,k} & v_{n,z,k} \end{bmatrix}^T \tag{6.4}$$

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.5}$$

State Transition Matrix The state transition matrix for axis i, $A_{f,i}$, is defined as follows:

$$A_{f,i} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 0 \end{bmatrix} \tag{6.6}$$

where Δt is the sampling period of the accelerometer.

The state transition matrix for the overall system, which is defined in Equation (6.1) is combined for the three axes with the addition of three sub-DCM matrices and a 3×3 identity matrix as shown in Equation (6.7). In order to include the oblique effect in the noise propagation during estimation, the state transition matrix is changed. We use the accelerations in the body frame to be the measurement input instead of using the accelerations in the navigation frame. Hence, the $(3, 3)$-element in each sub-matrix $A_{f,i}$, defined in the Equation (6.6), is zero, so as to let three DCM sub-matrices transform the acceleration in body frame to the navigation frame. The sub-matrix I_3 is used to propagate the previous a posteriori acceleration estimates in the body frame to the current state.

$$A_f = \begin{bmatrix} A_{f,x} & 0 & 0 & DCM_1 \\ 0 & A_{f,y} & 0 & DCM_2 \\ 0 & 0 & A_{f,z} & DCM_3 \\ 0 & 0 & 0 & I_3 \end{bmatrix} \tag{6.7}$$

where DCM_1, DCM_2, and DCM_3 are defined below. As the 3-axes accelerations in navigation frame are separated in the state estimate vector $\hat{x}_k$, therefore the DCM is decomposed into three parts and padded with zero to have these three matrices to compute the coordinate rotation.

$$DCM_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \cos\theta\cos\psi & -\cos\theta\sin\psi & \sin\theta \end{bmatrix} \tag{6.8}$$

$$DCM_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \begin{pmatrix} \sin\phi\sin\theta\cos\psi \\ +\cos\phi\sin\psi \end{pmatrix} & \begin{pmatrix} -\sin\phi\sin\theta\sin\psi \\ +\cos\phi\cos\psi \end{pmatrix} & -\sin\phi\cos\theta \end{bmatrix} \tag{6.9}$$

$$DCM_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \begin{pmatrix} -\cos\phi\sin\theta\cos\psi \\ +\sin\phi\sin\psi \end{pmatrix} & \begin{pmatrix} \cos\phi\sin\theta\sin\psi \\ +\sin\phi\cos\psi \end{pmatrix} & \cos\phi\cos\theta \end{bmatrix} \tag{6.10}$$

and I_3 is the 3×3 identity matrix which is defined as follows:

$$\mathbf{I_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.11}$$

Process Noise Covariance The process noise covariance matrix is used to estimate and reduce the effect of the noise propagated to the velocity and position. The implementation depends on the state vector. By comparing the state vectors in Equations (5.3) and (6.1), since we include the 3-axes acceleration in the body frame in this enhanced Kalman filtering algorithm, the process noise covariance matrix is defined as follows. The first 9 state variables are the same as Equation (5.3), therefore the first 9×9 entries is the same as in Equation (5.12). For the last three diagonal element, i.e. $(10, 10)$, $(11, 11)$ and $(12, 12)$-elements of the process noise covariance matrix as shown in Equation (6.12), they are related to the acceleration in the body frame, so we put the discrete noise covariance ($q_c\Delta t$) in there. For the rest, since we have assumed that the 3-axes are orthogonal, zeros are used.

$$Q_f = \begin{bmatrix} Q_x & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & Q_y & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & Q_z & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & q_c\Delta t \cdot \mathbf{I_3} \end{bmatrix} \tag{6.12}$$

where Q_i is the process noise covariance matrix in axis $i = x, y, z$ as defined in Equation (5.13), $\mathbf{I_3}$ is a 3×3 identity matrix as defined in Equation (6.11) and $\mathbf{0_3}$ is the 3×3 zero matrix as defined in Equation (5.11).

Measurement Noise Covariance The first three diagonal elements in the measurement noise covariance matrix R depends on the noise distribution of the accelerometers used. Assuming that the accelerometers in each axis i are independent, they can be described as defined in Equation (6.13). For the other two measurement inputs, from our assumption, they are absolute inputs to the algo-

rithm, so there should be no noise for these two inputs, hence zero noise covariance is set for these two elements, and the whole measurement noise covariance matrix is defined as follows:

$$
\begin{aligned}
R &= E[v \cdot v^T] \\
&= \begin{bmatrix} R_x & 0 & 0 & 0 & 0 \\ 0 & R_y & 0 & 0 & 0 \\ 0 & 0 & R_z & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned} \tag{6.13}
$$

where R_x, R_y and R_z is the accelerometer noise covariance in x-axis, y-axis and z-axis respectively.

6.3.2 Experimental Results and Discussion

In order to test the accuracy of the algorithm proposed, we simulated stationary accelerometers and rotated the pen to produce an acceleration. We assume Gaussian distributed noise in the accelerometers with covariance (ω^2) of 10 $cm^2 s^{-4}$ as shown in Figure 6.1.

It can be seen that the overall noise level using our feedback error compensation technique shown in Figure 6.2 is much lower than for the accelerometer-only Kalman filtering algorithm with acceleration measurement only Figure 6.3, especially in the steady state.

Through simulation with realistic sensor noise levels as shown in Figure 6.1, comparing Figures 6.2 and 6.3, the proposed enhanced Kalman filtering technique can be seen to have a better acceleration error reduction than the accelerometer-only Kalman filter for a pen-based inertial measurement unit. However, as can be seen in Figure 6.4, an experimental result using real data, the position estimate given by the new algorithm is roughly the same as that given by the accelerometer-only Kalman filtering or the ZVC algorithm. This shows that although the algorithm can shorten the stabilization time for the estimation and maintain low fluctuation during the stable state, the position estimate is still very sensitive to accelerometer noise. Hence, zero z-axis velocity and position still cannot provide sufficient guidance to the estimation, and the position estimate does not improve significantly.

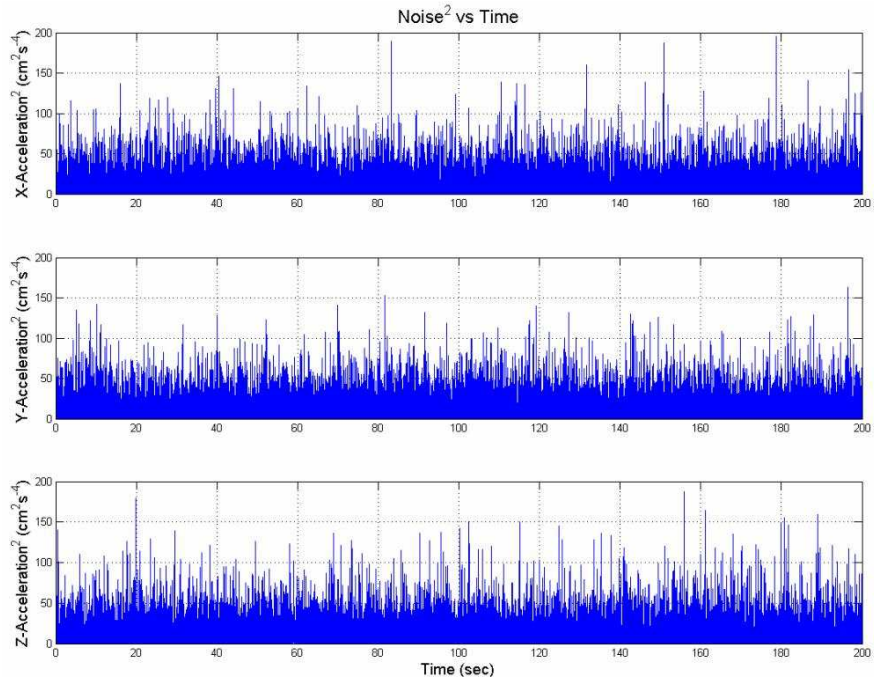


Figure 6.1: Random Noise in the Three-axes Accelerometers

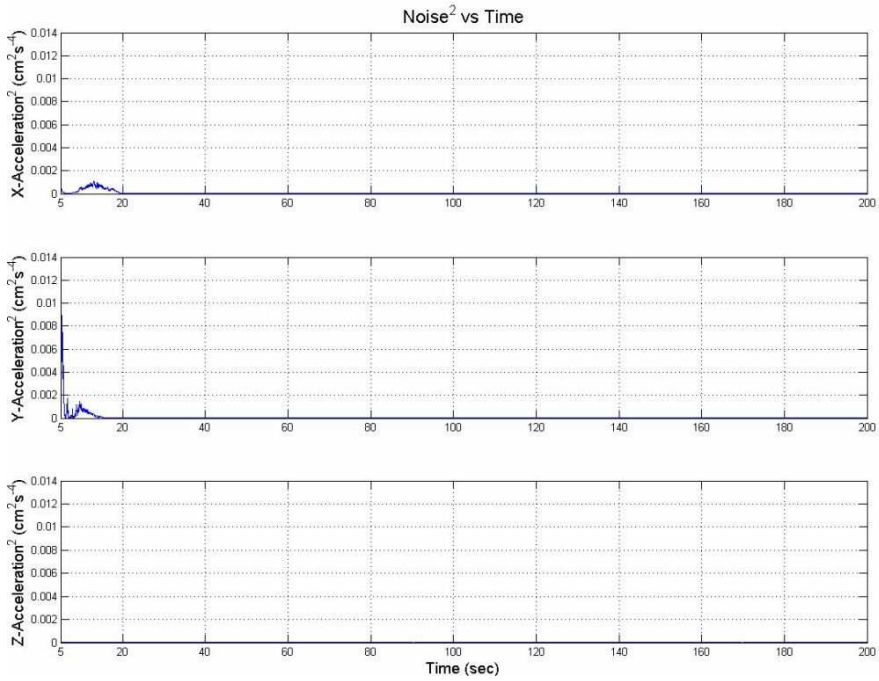


Figure 6.2: Noise level of Accelerometers after using Kalman filtering algorithm with Pen-tip Compensation

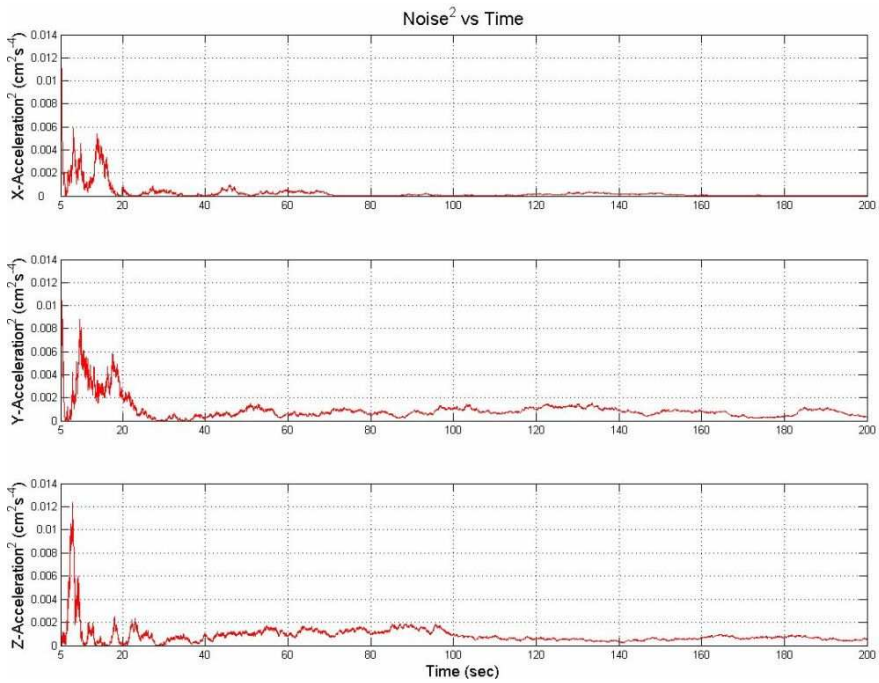


Figure 6.3: Noise level of Accelerometers after using Kalman filtering algorithm without Pen-tip Compensation

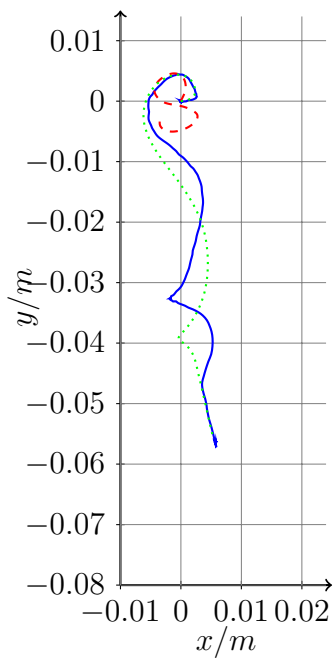


Figure 6.4: The Performance Comparison between the zero z-axis position feedback enhanced Kalman filtering and integration without any processing. The solid line is produced by the zero z-axis position feedback enhanced Kalman filtering, whereas the dotted line is produced by integration from raw data and the dashed line is the actual handwriting.

6.4 Combined Electromagnetic Resonance (EMR) Position Detection Board and μIMU

6.4.1 EMR Position Detection System

In the previous section, we have described a zero z-axis Kalman filtering algorithm. From the experimental results, it is not accurate enough to do the position estimation, but it shows that more information introduced to the system will help the estimation performance, especially in providing the position reference to the filter. Fortunately, a common human-computer interaction device, namely a graphics tablet, can provide position information. The first graphics tablet, the RAND Tablet, appeared in 1964 [12] and many modifications and improvements since then have been reported. The basic working principle, which was also used in the RAND Tablet, is to use electromagnetic resonance (EMR) to detect the position of a stylus.

In Figure 6.5, a system level diagram of an electromagnetic resonance motion detection system is given. In Figure 6.6(a), a grid of wires, which consists of two sets of coils arranged both horizontally and vertically, is used to determine the horizontal and vertical coordinates of the digital pen or eraser. The coil grid acts as antenna to receive the time-varying magnetic field generated by the resonant circuit (RLC circuit) shown in Figure 6.5 in the transmitter, which is installed on the pen and eraser as shown in Figure 6.6(b). Hence, if the transmitter is within a coil, the coil will generate a voltage with a magnitude that represents how close the pen is. This voltage is then digitized through an analog-to-digital converter (ADC) in the motion detection circuit as shown in Figure 6.6(c) to be transmitted to the host computer through a bluetooth module to locate the position of the pen.

Based on the basic working principle described above, when the pen moves across the board, the voltage is generated in each channel of the coil grid. This voltage will vary with the distance between the transmitter and the centre of the coil. Figure 6.7(a) shows the magnetic field strength of two signal channels from the EMR motion detection board, and Figure 6.7(b) shows their corresponding coordinates. These two signal channels are picked up from the coils with the largest two voltages, and are the closest two coils to the pen. In Figure 6.7, the solid line represents the channel having the largest signal magnitude; whereas the dashed line represents the channel having the second largest signal magnitude. If the transmitter is within a coil, defined as region I, and the solid line is higher

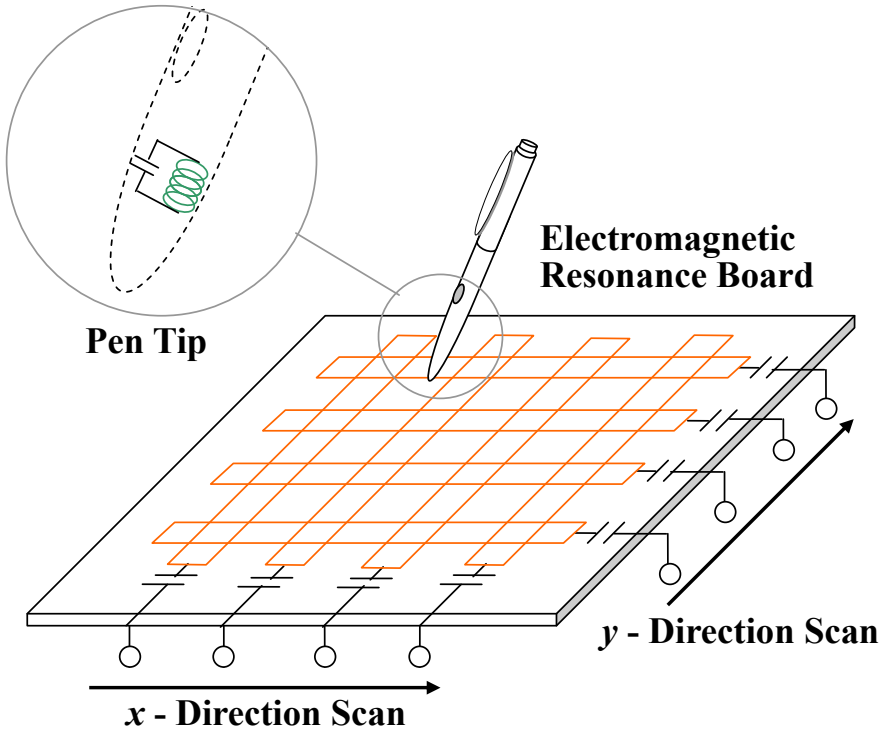


Figure 6.5: Working Principle of Electromagnetic Resonance Motion Detection System (modified from reference [36])

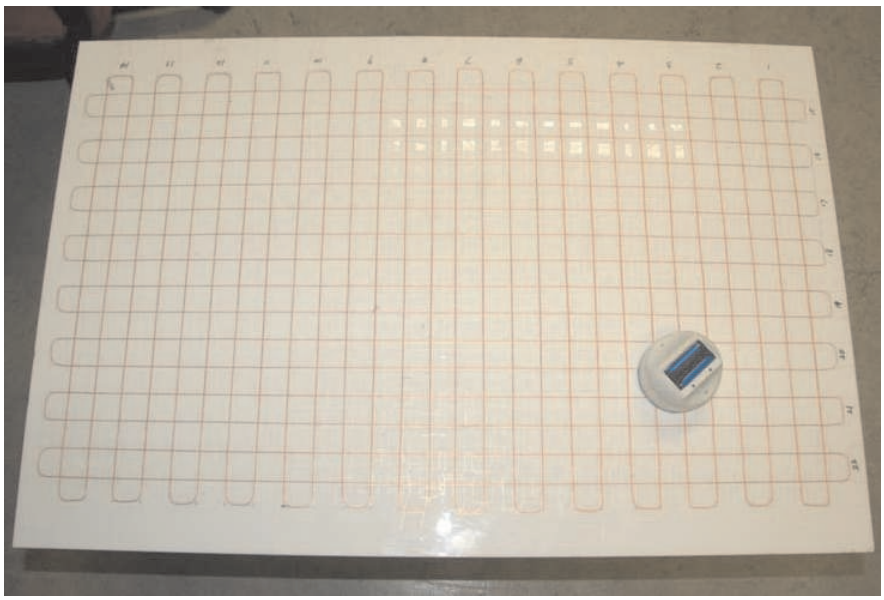
than a threshold as shown in Figure 6.7(a). We can obtain the corresponding coordinate as shown in Figure 6.7(b). If the transmitter is between two coils, defined as region II, we can take an average of the coordinates of the two signal channels to determine the position.

6.4.2 A Combined Scheme

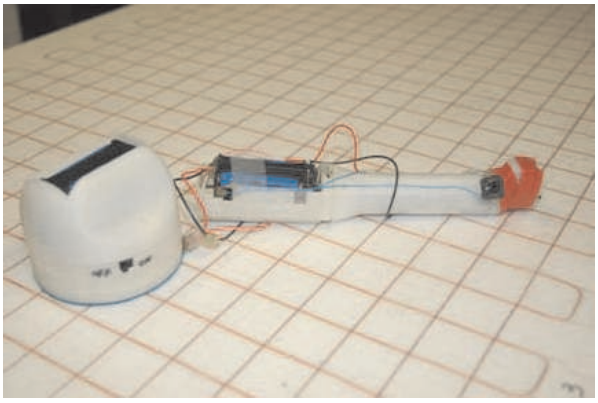
Owing to the existence of random noise in the μIMU, error propagates from accelerometer measurement to the position estimate and is unbounded and drifts with time. Many researchers have proposed different error compensation algorithm to solve it. However, up to date, there is no successful real-time solution to solve the problem for handwriting.

Although the EMR motion detection board can provide position information, its performance is poor for large writing areas, particularly if good resolution is needed. Since the accuracy of the motion detection board depends on the density of the coil, the searching time will increase with the number of channel used and results in the increase of the delay time in finding the position of the pen.

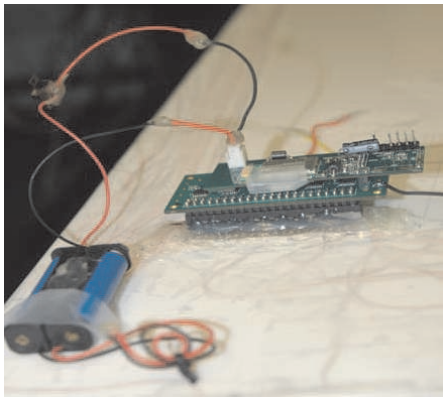
We propose a combined scheme using an μIMU and EMR motion detection board which comes from the idea of sensor fusion in an IMU/GPS coupled system



(a) Whiteboard with Coils

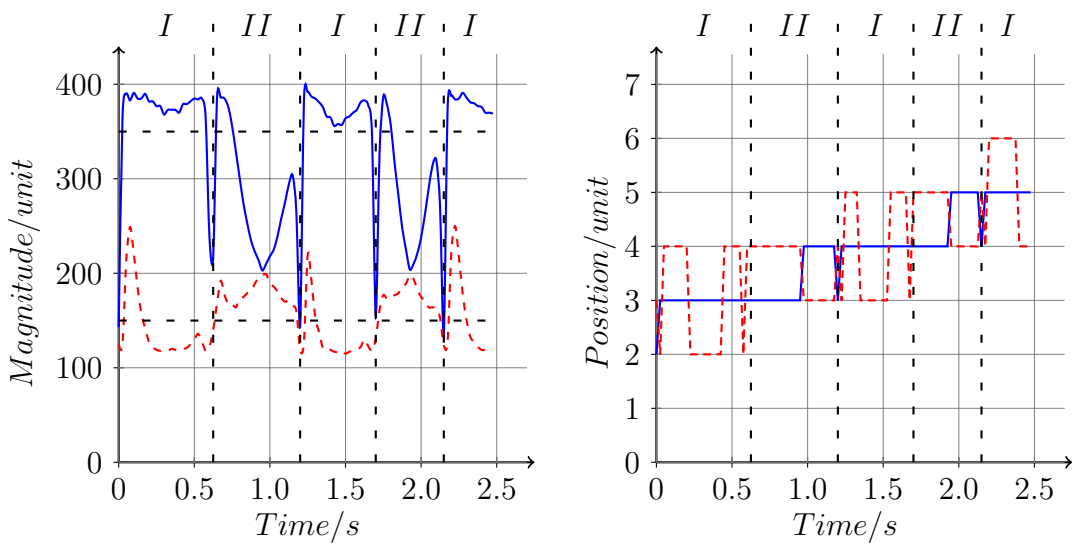


(b) Digital Pen and Eraser



(c) Electromagnetic Resonance Motion Detection Circuit

Figure 6.6: Electromagnetic Resonance Motion Detection System



(a) Magnitude vs Time

(b) Position vs Time

Figure 6.7: Signal from Electromagnetic Resonance (EMR) Position Detection System. The solid line represents the channel having the largest signal magnitude; whereas the dash line represents the channel having the second largest signal magnitude.

[20]. A low resolution EMR motion detection board can provide absolute position with bounded error, accuracy being limited by the grid size, and the μIMU can provide the detailed part of the motion. Hence, we combine the benefits of μIMU and EMR schemes to give more accurate position estimation.

6.4.3 Algorithm Implementation

In order to realize the idea mentioned above, we add the available position information collected by the EMR motion detection board as feedback to the Kalman filtering algorithm of Chapter 5.

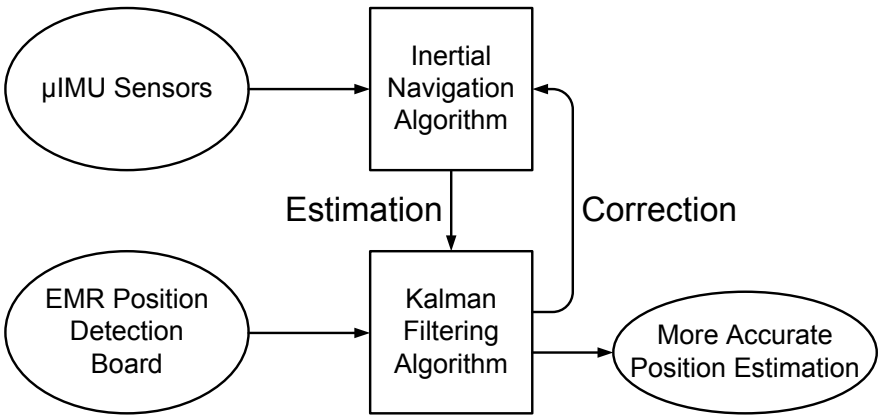


Figure 6.8: A Combined Scheme of μIMU and EMR Motion Detection Board

6.4.3.1 Initialization

Firstly, we need to define and initialize the system state and the other parameters used in the Kalman filter.

System State Estimate We first define the 3-axes navigation information, which is position, velocity and acceleration, as the system state defined in Equation (5.3) and initialize them as a zero vector with the assumption that the pen is at rest and at the origin $(0, 0, 0)$ at the beginning as described in Equation (5.4).

State Transition Matrix Since the system state defined in Equation (5.3) is the same as the system state defined for the Kalman filter in Equation (5.3), therefore we can reuse the state transition matrix described in Equation (5.10).

Error Covariance Matrix Similar to the state transition matrix, we use the same state transition matrix, so the error covariance matrix can be initialized as in the error covariance matrix of the Kalman filter described in Equation (5.5).

Measurement Matrix In the combined scheme proposed, we have two measurement sources to feedback the filter, the first one is from the μIMU, and the other is from the EMR motion detection board. Hence, we have two measurement matrix for these two measurement sources.

For the measurement matrix for the μIMU, H_{IMU}, since we make use of 3-axes accelerometers as the μIMU measurement input for the Kalman filter, therefore acceleration is observed during the measurement update. Before using the accelerometer measurement in the Kalman filter, a coordinate transformation must be performed as mentioned in Section 3.4 to transform the acceleration in body frame A_b to that in navigation frame A_n. Given the measurement vector $y_{IMU,k}$ as shown in Equation (6.14), from the measurement model defined in Equation (5.2), we need to select the 3-axes acceleration from the system state $\hat{x}$ for updating the Kalman filter from the measurement. As defined in Equation (5.3), the accelerations in the navigation frame $a_{n,x}$, $a_{n,y}$ and $a_{n,z}$ are the $3rd$, $6th$ and $9th$ elements of the system state x. Hence we construct the measurement matrix H_{IMU} as described in Equation (6.15) to map the measurement input $y_{IMU,k}$ to the corresponding accelerations in the system state estimate $\hat{x}_k$.

$$y_{IMU,k} = \begin{bmatrix} a_{n,x} & a_{n,y} & a_{n,z} \end{bmatrix}^T \tag{6.14}$$

$$H_{IMU} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.15}$$

For the measurement matrix for the EMR motion detection board described in Equation (6.17), since the board gives out the position of x-axis and y-axis, s_x and s_y, as shown in Equation (6.16) to feedback the system, hence the measurement matrix H_{EMR} is described as Equation (6.17) which updates the position of x-axis and y-axis in the system state $\hat{x}$ to the measurement input from the EMR system.

$$y_{EMR,k} = \begin{bmatrix} s_x & s_y \end{bmatrix}^T \tag{6.16}$$

$$H_{EMR} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.17}$$

Process Noise Covariance As for the error covariance matrix, the process noise covariance matrix is the same as that of the Kalman filter described in Equation (5.12).

Measurement Noise Covariance For the μIMU measurement noise covariance matrix R_{IMU}, since the measurement input just has 3-axis accelerations, therefore, under the assumption of independent and perpendicular accelerometers used in each axis, the measurement noise covariance matrix can be described as a diagonal matrix as follows:

$$\begin{aligned} R_{IMU} &= E[v \cdot v^T] \\ &= \begin{bmatrix} R_x & 0 & 0 \\ 0 & R_y & 0 \\ 0 & 0 & R_z \end{bmatrix} \end{aligned} \tag{6.18}$$

where R_x, R_y and R_z is the accelerometer noise covariance in x-axis, y-axis and z-axis respectively.

However, for the EMR motion detection board measurement noise covariance, R_{EMR}, since the position, given by the EMR motion detection system, represents the centre of the grid and the pen location is evenly distributed over the entire board, we assume a uniform distribution, and the measurement noise covariance in each axis, $R_{EMR,i}$ can be computed as follows:

$$R_{EMR,i} = \int_{-b}^{b} s^2 ds = \frac{2b^3}{3} \tag{6.19}$$

where s is position and b is half the width of the grid, i.e. 1.5 cm.

Since the motion detection of the axes are independent, the overall EMR motion detection board measurement noise covariance matrix is a diagonal matrix as follows:

$$R_{EMR} = \begin{bmatrix} R_{EMR,x} & 0 \\ 0 & R_{EMR,y} \end{bmatrix} \tag{6.20}$$

6.4.3.2 Time Update

The state estimate and error covariance are propagated based on the information at previous time instant $k-1$ to the current time instant k by using the Equations (5.14) and (5.15).

6.4.3.3 Accelerometer Measurement Update

When the accelerometer information is available, we can update the Kalman filter based on the acceleration as discussed in Chapter 5 with the Equations (5.16), (5.17) and (5.18).

6.4.3.4 EMR Position Detection Board Measurement Update

This stage updates the state estimation based on the measurement from the EMR motion detection system. We use the following equations.

Kalman Gain for EMR Position Detection Board Measurement Update

$$K_{EMR,k} = \frac{P_k^- H_{EMR}^T}{H_{EMR} P_k^- H_{EMR}^T + R_{EMR}} \tag{6.21}$$

where K_{EMR}^k is the Kalman gain for the EMR motion detection board measurement update at time instant k, P_k^- is the a priori error covariance matrix at time instant k, H_{EMR} is the measurement matrix for the EMR motion detection system as defined in Equation (6.17), and R_{EMR} is the EMR motion detection board measurement noise covariance matrix described in Equation (6.19).

A Posteriori State Estimate Propagation

$$\hat{x}_k = \hat{x}_k^- + K_{EMR}^k \left(y_{EMR,k} - H_{EMR} \hat{x}_k^- \right) \tag{6.22}$$

where $y_{EMR,k}$ is the position measurement from the EMR motion detection board.

A Posteriori Error Covariance Propagation

$$P_k = \left(I - K_{EMR}^k H_{EMR}\right) P_k^- \tag{6.23}$$

where P_k is the posterior error covariance matrix at time instant k.

6.4.4 Synchronization

Since the measurements from μIMU and EMR motion detection board are separated in the system, a synchronization process should be performed before navigation tracking. We assume that the transmission interval between two sets of data from sensors boards are equal, meaning that the transmission intervals in μIMU and the motion detection board are fixed to 200 Hz and 40 Hz receptively. Since we receive the data from the sensor board in a block of multiple samples, we use the machine clock of the host computer to record the initial data and final data reception time, and then label data from both μIMU and EMR motion detection board with a timestamp. After that, we sort the two data sequences according to the timestamp and compute the position estimate with ordered data.

6.4.5 Experimental Results and Discussion

In order to investigate the feasibility of the combined scheme, we make use of an optical calibration system, proposed by Dong *et al.* [14]. The setup is shown in Figure 6.9. In this system, a high speed (200 Hz) camera is used to record the handwriting script. A motion estimation algorithm, Parallel Full Search (PFS), together with a Correlation Coefficient (CC) matching criteria is used to trace the locus of the script at 200 Hz. With this position information, we can measure the error, which is the distance between the actual position and the estimated position, for every sample point.

Direct integration is used to determine position from raw data using the following equation

$$s_{n,i,k} = s_{n,i,k-1} + v_{n,i,k-1}\Delta t + \frac{1}{2}a_{n,i,k-1}\Delta t^2 \tag{6.24}$$

$$v_{n,i,k} = v_{n,i,k-1} + a_{n,i,k-1}\Delta t \tag{6.25}$$

where $s_{n,i,k}$, $v_{n,i,k}$ and $a_{n,i,k}$ are the position, velocity and acceleration in axis i of the navigation frame at time instant k respectively and Δt is the sampling time of the accelerometer.
This is compared with the ZVC algorithm and the coupled scheme.

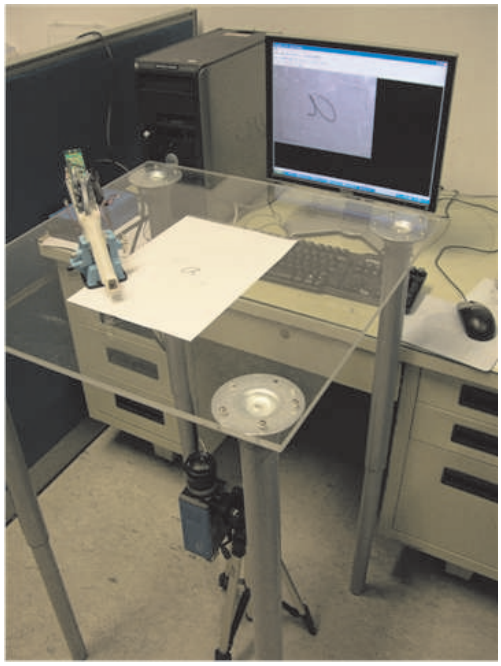
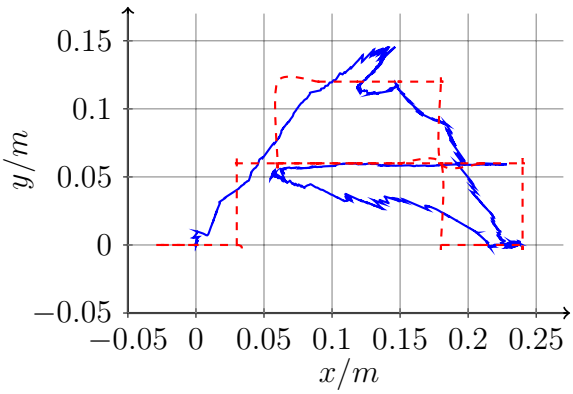


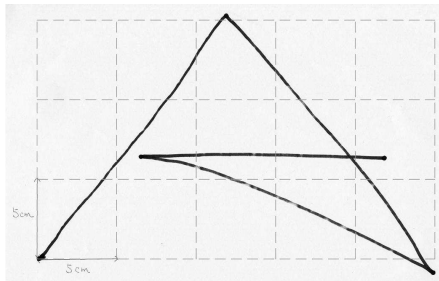
Figure 6.9: Optical Calibration System

Table 6.1 shows a comparison of the average distance estimation error per sample point in using the three algorithms. From 26 letters, in terms of the average distance estimation error, the ZVC algorithm is better 84.6% (22/26) of the time compared with direct integration from raw data. And also, the coupled scheme is always better than direct integration from the raw data, and 76.9% (20/26) of the time better than the ZVC algorithm.

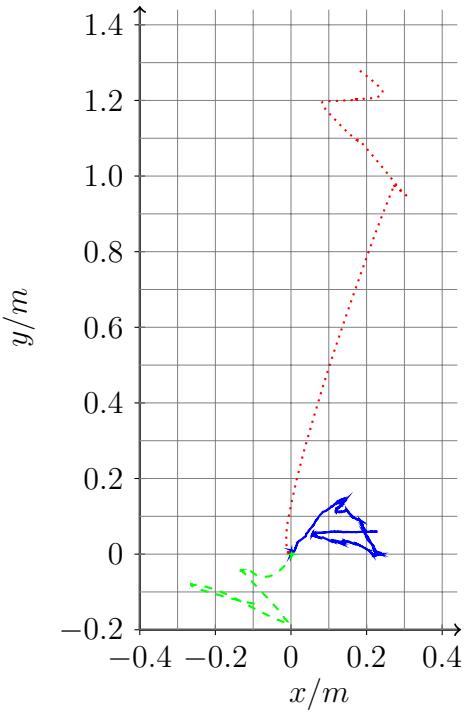
We further test the complete system by writing a letter "A" as shown in Figure 6.10(b) with our new algorithm. In Figure 6.10(c), the dotted line represents the position estimated by the integration from the raw data; the dashed line represents the position estimated by the ZVC algorithm, and the solid line represents the position estimated by the combined EMR/μIMU scheme. Compared to the result obtained by direct integration from the raw data and the ZVC algorithm, the position drift in the script reproduced by the μIMU/EMR motion detection board coupled system has been removed and the position estimate error is bounded, that shows why the performance of the coupled scheme is better than direct integration. However, the EMR motion detection board cannot provide high resolution pen position information, because there is a tradeoff between the grid separation distance and the board sampling frequency. With the help of the μIMU, improved results can be obtained, but there is a limitation is minimum letter size since the resolution is limited to 3 cm. That is why some of the cases in the test show that the coupled scheme cannot provide a more accurate result in terms of the average distance estimation error than the ZVC algorithm.



(a) "A" reproduced by μIMU/EMR motion detection board coupled system



(b) The actual "A" written in the experiment



(c) The comparison of the position tracking performance between integration without any processing, ZVC and the new algorithm. The solid line is produced by the coupled scheme; whereas the dash line and the dotted line is produced by ZVC algorithm and integration from the raw data.

Figure 6.10: The performance of the μIMU/EMR motion detection board coupled system

Letter	Integration from raw data	ZVC algorithm	Coupled Scheme
a	0.129212	0.032566	**0.011514**
b	0.026502	0.030031	**0.014817**
c	0.046248	0.018910	**0.008414**
d	0.231922	0.013393	**0.012635**
e	0.057270	0.014407	**0.010532**
f	0.048683	0.032737	**0.017284**
g	0.122060	0.051408	**0.015710**
h	0.035847	0.017685	**0.010579**
i	0.073466	0.018630	**0.009849**
j	0.033743	0.036292	**0.011147**
k	0.020570	0.042459	**0.016998**
l	0.027476	**0.006819**	0.009191
m	0.117594	**0.015396**	0.026178
n	0.026800	**0.005529**	0.022134
o	0.051911	**0.007474**	0.020675
p	0.167984	**0.015567**	0.016323
q	0.142942	0.042050	**0.012922**
r	0.026205	0.032892	**0.015568**
s	0.062084	0.016142	**0.014145**
t	0.044068	0.014701	**0.014258**
u	0.062084	0.016142	**0.014145**
v	0.023015	0.013845	**0.011506**
w	0.085311	0.019105	**0.014937**
x	0.216970	**0.012452**	0.018213
y	0.106834	0.024192	**0.013328**
z	0.087265	0.050143	**0.013718**

Table 6.1: A comparison of the average distance errors per sample point between position estimated by the integration from the raw data, the ZVC algorithm and the coupled scheme (unit: metre).

6.5 Summary

In this chapter, we have demonstrated two novel position estimation techniques. One is the zero z-axis Kalman filtering and the other is the μIMU/EMR motion detection board coupled system. The first approach cannot provide sufficient improvement in the accuracy of the position estimation, but it tells us the importance of the amount of the measurement information provided to the system. Based on this idea, the second approach tries to improve the position estimation from the μIMU by combining a low resolution absolute position reading from the EMR board with detailed but noisy information from the μIMU. Experimental results were presented which show the feasibility of the idea and its superiority over direct integration and the ZVC schemes.

Chapter 7

Conclusion

The main objective of this work was to develop practical techniques to minimize the error in position tracking in MEMS accelerometer-based digital input device. Several subproblems were addressed and original contributions made.

MEMS Accelerometer-based Digital Writing Instrument

We developed a MEMS accelerometer-based digital writing instrument. The accelerometer picks up the acceleration generated during handwriting which is transmitted to a host computer to compute the position of the pen tip. However, as random noise degrades the acceleration readings, a positional drift results when double integration of acceleration is applied. Several error reduction schemes were investigated including zero velocity compensation (ZVC) and Kalman filtering.

Zero Velocity Compensation

Zero velocity compensation (ZVC), proposed by the Samsung Advanced Institute of Technology [3, 5], demonstrated an ability to remove positional drift. However, a one stroke delay is introduced and it is not able to determine position in real-time.

Error Compensation from Position Feedback

Kalman filtering algorithm is a well-known sensor fusion algorithm. Based on the idea of coupled IMU/GPS systems, we developed two algorithms to improve the position tracking performance for the μIMU. First, a zero z-axis Kalman filtering algorithm (Algorithm 7.1), which assumes writing is in a two dimensional plane with the z-axis being zero in the navigation frame was tried. However, from the experimental results, the position tracking accuracy still was not sufficient for

a practical digital writing instrument system. This is because this idea can correct the z-axis, but not the x and y-axes.

Algorithm 7.1 Zero z-axis Kalman Filter

1: **loop**
2: Time Update
$\hat{x}_k^- = A\hat{x}_{k-1}^-$
$P_k^- = AP_{k-1}A^T + Q$
3: Compute Kalman Gain
$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R}$
4: Measurement Update
$x_k = \hat{x}_k^- + K_k(y_k - H\hat{x}_k^-)$
$P_k = (I - K_k H)P_k^-$
5: **end loop**

We further developed a practical coupled system for digital writing instrument (Algorithm 7.2). It uses the μIMU with an electromagnetic resonance (EMR) motion detection board, which can provide the position information of the pen in x-axis and y-axis of the navigation frame. Using an optical tracking scheme to record absolute position, it was shown that it has improved performance over direct integration and the ZVC algorithm. From experimental results, the EMR board can track position with limited position error, this depending on the grid size. Although the EMR board cannot provide position information at a high frequency (limited to only 40 Hz), the μIMU, which can output the acceleration information in 200 Hz, can track the detailed part of the handwriting motion. The handwriting trajectory can be successful reproduced through this coupled scheme.

7.1 Future Work

7.1.1 Improvement in the μIMU

Throughout the whole dissertation, we use MEMS accelerometers to perform navigation tracking under the assumptions of zero rotation during writing and no rotation in yaw. This assumption is not very practical, especially in cursive writing, so we should also include the effect of rotational change in the position computation. In the future, a MEMS gyroscope, already installed in our μIMU, can be used to acquire angular rate, and this information used to track the position in a more accurate and realistic manner.

Algorithm 7.2 Coupled EMR and μIMU Kalman Filter

1: **loop**
2: Time Update
$\hat{x}_k^- = A\hat{x}_{k-1}^-$
$P_k^- = AP_{k-1}A^T + Q$
3: Compute Kalman Gain for μIMU
$K_{IMU,k} = \frac{P_k^- H_{IMU}^T}{H_{IMU} P_k^- H_{IMU}^T + R_{IMU}}$
4: Measurement Update for μIMU
$\hat{x}_k = \hat{x}_k^- + K_{IMU,k}(y_{IMU,k} - H_{IMU}\hat{x}_k^-)$
$P_k = (I - K_{IMU,k}H_{IMU})P_k^-$
5: Compute Kalman Gain for EMR Position Detection Board
$K_{EMR,k} = \frac{P_k^- H_{EMR}^T}{H_{EMR} P_k^- H_{EMR}^T + R_{EMR}}$
6: Measurement Update for EMR Position Detection Board
$\hat{x}_k = \hat{x}_k^- + K_k(y_{EMR,k} - H_{EMR}\hat{x}_k^-)$
$P_k = (I - K_{EMR,k}H_{EMR})P_k^-$
7: **end loop**

Besides that, the misalignment of the sensors also contributes to error in the position computation. It would be better to use a 6 degree-of-freedom MEMS inertial sensor, which consists of 3-axis accelerometers and 3-axis gyroscopes in a single chip, instead of using two 2-axis accelerometers and three 1-axis gyroscopes. This can reduce misalignment errors during μIMU installation and reduce the size of the μIMU.

At this stage, the MEMS sensor technology is still under development phrase, so that it is unavoidable that the cost of the sensors and the overall system is much higher than other existing solutions. However, with the improvement of the fabrication technology, we can forecast that the cost will be further reduced when the mass production is available and the competition between the MEMS sensor companies increased.

7.1.2 Combined Camera Optical Tracking and μIMU

As discussed in the Chapter 6, we have combined EMR motion detection and the μIMU to compute position. However, it is not practical to use a large dedicated writing board for a ubiquitous digital writing system. Fortunately, in the computer vision technology can provide an accurate position tracking solution. We can install a small-sized digital camera, roughly 1 $cm \times 1$ cm in size, on the digital pen, and then transmit the image data to a host computer to track the movement of the pen. However, as for the EMR motion detection board, the digital camera cannot provide position information at high frequency, and a Kalman filter that combines the absolute camera information with μIMU data should be

used.

7.2 Concluding Remarks

In this thesis, we have demonstrated the feasibility of using MEMS accelerometers to implement a ubiquitous digital writing instruments. Owing to the random noise associated with accelerometers, error reduction techniques are needed to improve the accuracy of position tracking. We have illustrated and discussed several error reduction schemes, and found that the most promising scheme is to use position information from an EMR motion detection board. When this is used with accelerometer information, tracking position error can be bound to a reasonable value, and a satisfactory digital writing instrument based on this scheme can be developed.

Appendix A

Derivation of Kalman Filtering Algorithm

A.1 Introduction

In this appendix, we would like to derive the Kalman filter algorithm to let us study it from a theoretical perspective [25, 4, 6, 19, 20, 27, 38, 40, 41, 42, 44, 50]. The idea of the Kalman filtering algorithm is to obtain a optimal state estimate based on the past state estimate and the measurement input. The algorithm is based on the state-space approach to model the system, so we first define the process and measurement models for state estimation.

Process Model:

$$x_k = Ax_{k-1} + w_k \tag{A.1}$$

Measurement Model:

$$y_k = Hx_k + v_k \tag{A.2}$$

where x_k is the state vector, y_k is the measurement vector, A is the transition matrix; whereas H is the observation matrix, w_k is the process noise covariance, and v_k is the measurement noise covariance at the time instant k.

Before the derivation, we would like to first given the assumption that we will make in the Kalman filtering

- The system state estimate is unbiased.

$$E\left[\tilde{x}_k\right] \equiv E\left[\hat{x}_k - x_k\right] = 0, \forall k \tag{A.3}$$

- The process noise covariance and measurement noise covariance is zero-mean Gaussian distributed.

$$E\left[w_k\right] = 0, \forall k \tag{A.4}$$

$$E\left[v_k\right] = 0, \forall k \tag{A.5}$$

- The process noise covariance is uncorrelated to the estimation error.

$$E\left[w_{k+1}\left(\hat{x}_k - x_k\right)^T\right] = E\left[\left(\hat{x}_k - x_k\right) w_{k+1}^T\right] = 0 \tag{A.6}$$

- The measurement noise covariance is uncorrelated to the estimation error.

$$E\left[v_{k+1}\left(\hat{x}_{k+1} - x_{k+1}\right)^T\right] = E\left[\left(\hat{x}_{k+1} - x_{k+1}\right) v_{k+1}^T\right] = 0 \tag{A.7}$$

Based on the arrival of the measurement input, the recursive process can be separated into two stages, a priori state and a posteriori state.

A.2 Derivation of a Priori State Estimation Equation

In the Kalman filtering, we take the expectation of the state to be the state estimate $\hat{x}_k$, hence, we can deduce the a priori state estimation equation as follows:

$$\begin{aligned} E\left[x_k\right] &= AE\left[x_{k-1}\right] + E\left[w_k\right] \\ \hat{x}_k &= A\hat{x}_{k-1} \end{aligned} \tag{A.8}$$

A.3 Derivation of a Posteriori State Estimation Equation

In the Kalman filtering, after the arrival of the measurement input, the state estimate will be updated based on the previous state estimate and the measurement input, so we define the a posteriori state estimation as follow:

$$\hat{x}_{k+1} \equiv K_{k+1}^A \hat{x}_k + K_{k+1}^B z_{k+1} \tag{A.9}$$

We define a posteriori error residue $\tilde{x}_k$ is the difference between the a posteriori state estimate $\hat{x}_k$ and the actual state x_k at time instant k.

$$\begin{aligned} \tilde{x}_{k+1} &\equiv \hat{x}_{k+1} - x_{k+1} \\ &= \left(K_{k+1}^A \hat{x}_k + K_{k+1}^B z_{k+1}\right) - x_{k+1} \\ &= K_{k+1}^A \hat{x}_k + K_{k+1}^B \left(Hx_{k+1} + v_{k+1}\right) - x_{k+1} + \left(K_{k+1}^A x_k - K_{k+1}^A x_k\right) \\ &= K_{k+1}^A \left(\hat{x}_k - x_k\right) + K_{k+1}^B \left[H\left(Ax_k + w_{k+1}\right) + v_{k+1}\right] - \left(Ax_k + w_{k+1}\right) + K_{k+1}^A x_k \\ &= K_{k+1}^A \tilde{x}_k + \left(K_{k+1}^B HA - A + K_{k+1}^A\right) x_k + \left(K_{k+1}^B H - I\right) w_{k+1} + K_{k+1}^B v_k \end{aligned} \tag{A.10}$$

Taking the expectation of Equation (A.10), we have

$$\begin{aligned} E\left[\tilde{x}_{k+1}\right] = K_{k+1}' E\left[\tilde{x}_k\right] + \left(K_{k+1}HA - A + K_{k+1}'\right) E\left[x_k\right] \\ + \left(K_{k+1}H - I\right) E\left[w_{k+1}\right] + K_{k+1} E\left[v_k\right] \end{aligned}$$

Based on the assumptions described in the Equations (A.3), (A.4) and (A.5), we can obtain

$$\left(K_{k+1}HA - A + K'_{k+1}\right) E\left[x_k\right] = 0 \tag{A.11}$$

Since $E\left[x_k\right]$ may not be equal to zero, which implies

$$K_{k+1}HA - A + K'_{k+1} = 0 \tag{A.12}$$

$$K'_{k+1} = \left(I - K_{k+1}H\right) A \tag{A.13}$$

Therefore, the a posteriori state estimation equation is derived as follows:

$$\begin{aligned}\hat{x}_{k+1} &= \left(I - K_{k+1}H\right) A\hat{x}_k + K_{k+1}z_{k+1} \\ &= A\hat{x}_k + K_{k+1}\left(z_{k+1} - HA\hat{x}_k\right) \\ &= \hat{x}^-_{k+1} + K_{k+1}\left(z_{k+1} - H\hat{x}^-_{k+1}\right)\end{aligned} \tag{A.14}$$

A.4 Derivation of a Priori Error Covariance Matrix

We define a priori error residue $\tilde{x}^-_k$ is the difference between the a priori state estimate $\hat{x}^-_k$ and the actual state x_k at time instant k.

$$\begin{aligned}\tilde{x}^-_{k+1} &\equiv \hat{x}^-_{k+1} - x_{k+1} \\ &= A\hat{x}_k - Ax_k - w_{k+1} \\ &= A\left(\hat{x}_k - x_k\right) - w_{k+1}\end{aligned}$$

Based on the assumption described in Equation (A.6), we have

$$P^-_{k+1} \equiv E\left[\tilde{x}^-_{k+1}\tilde{x}^{-T}_{k+1}\right] \tag{A.15}$$

$$\begin{aligned}&= E\left[\left(A\tilde{x}_k - w_{k+1}\right)\left(A\tilde{x}_k - w_{k+1}\right)^T\right] \\ &= AE\left[\tilde{x}_k\tilde{x}^T_k\right]A^T + E\left[w_{k+1}w^T_{k+1}\right] \\ &- E\left[w_{k+1}\tilde{x}^T_k\right]A^T - AE\left[\tilde{x}_k w^T_{k+1}\right] \\ &= AP_kA^T + Q_{k+1}\end{aligned} \tag{A.16}$$

where the process error covariance matrix is defined as follows:

$$Q \equiv E\left[w_{k+1}w^T_{k+1}\right] \tag{A.17}$$

A.5 Derivation of the Optimal Kalman Gain

Considering the a posteriori error reside $\tilde{x}_{k+1}$ again,

$$\begin{aligned}
\tilde{x}_{k+1} &\equiv \hat{x}_{k+1} - x_{k+1} \\
&= \left([I - K_{k+1}H]\,\hat{x}^{-}_{k+1} + K_{k+1}z_{k+1}\right) - x_{k+1} \\
&= [I - K_{k+1}H]\,\hat{x}^{-}_{k+1} + K_{k+1}\left[Hx_{k+1} + v_{k+1}\right] - x_{k+1} \\
&= \hat{x}^{-}_{k+1} - K_{k+1}H\hat{x}^{-}_{k+1} + K_{k+1}Hx_{k+1} + K_{k+1}v_{k+1} - x_{k+1} \\
&= \left(\hat{x}^{-}_{k+1} - x_{k+1}\right) - K_{k+1}H\left(\hat{x}^{-}_{k+1} - x_{k+1}\right) + K_{k+1}v_{k+1} \\
&= [I - K_{k+1}H]\left(\hat{x}^{-}_{k+1} - x_{k+1}\right) + K_{k+1}v_{k+1} \\
&= [I - K_{k+1}H]\,\tilde{x}^{-}_{k+1} + K_{k+1}v_{k+1} \qquad \text{(A.18)}
\end{aligned}$$

Based on the assumption described in Equation (A.7), we have

$$\begin{aligned}
P_{k+1} &\equiv E\left[\tilde{x}_{k+1}\tilde{x}^{T}_{k+1}\right] \\
&= [I - K_{k+1}H]\,E\left[\tilde{x}^{-}_{k+1}\tilde{x}^{-T}_{k+1}\right][I - K_{k+1}H]^{T} + K_{k+1}E\left[v_{k+1}v^{T}_{k+1}\right]K^{T}_{k+1} \\
&= [I - K_{k+1}H]\,P^{-}_{k+1}\,[I - K_{k+1}H]^{T} + K_{k+1}R_{k+1}K^{T}_{k+1} \\
&= [I - K_{k+1}H]\,P^{-}_{k+1}\left[I - H^{T}K^{T}_{k+1}\right] + K_{k+1}R_{k+1}K^{T}_{k+1} \qquad \text{(A.19)} \\
&= P^{-}_{k+1} - K_{k+1}HP^{-}_{k+1} - P^{-}_{k+1}H^{T}K^{T}_{k+1} \\
&+ K_{k+1}HP^{-}_{k+1}H^{T}K^{T}_{k+1} + K_{k+1}R_{k+1}K^{T}_{k+1} \qquad \text{(A.20)}
\end{aligned}$$

where the measurement noise covariance matrix is defined as follows:

$$R_{k+1} \equiv E\left[v_{k+1}v^{T}_{k+1}\right] \qquad \text{(A.21)}$$

The *trace* of a matrix is the sum of the diagonal elements of the matrix. Therefore, the trace of the error covariance matrix (P_{k+1}) is the sum of the mean squared errors. Hence, the mean squared errors is minimized by minimizing $Trace\,(P_{k+1})$

$$\begin{aligned}
Trace\,(P_{k+1}) &= Trace\left(P^{-}_{k+1}\right) - 2Trace\left(K_{k+1}HP^{-}_{k+1}\right) \\
&+ Trace\left(K_{k+1}\left[HP^{-}_{k+1}H^{T}\right]K^{T}_{k+1}\right) \\
&+ Trace\left(K_{k+1}R_{k+1}K^{T}_{k+1}\right) \qquad \text{(A.22)}
\end{aligned}$$

Taking the partial derivative with respect to K_{k+1}, we have

$$\frac{\delta Trace\,(P_{k+1})}{\delta K_{k+1}} = -2P^{-}_{k+1}H^{T} + 2K_{k+1}HP^{-}_{k+1}H^{T} + 2K_{k+1}R_{k+1} = 0 \qquad \text{(A.23)}$$

Therefore, the optimal Kalman gain is derived as follows:

$$K_{k+1} = \frac{P^{-}_{k+1}H^{T}}{HP^{-}_{k+1}H^{T} + R_{k+1}} \qquad \text{(A.24)}$$

A.6 Derivation of a Posteriori Error Covariance Matrix

From the Equation (A.20), we have

$$\begin{aligned} P_{k+1} &= [I - K_{k+1}H]\, P_{k+1}^{-}\, [I - K_{k+1}H]^T + K_{k+1}R_{k+1}K_{k+1}^T \\ &= [I - K_{k+1}H]\, P_{k+1}^{-} - P_{k+1}^{-}H^T K_{k+1}^T + K_{k+1}HP_{k+1}^{-}H^T K_{k+1}^T + K_{k+1}R_{k+1}K_{k+1}^T \\ &= [I - K_{k+1}H]\, P_{k+1}^{-} - P_{k+1}^{-}H^T K_{k+1}^T + K_{k+1}\left[HP_{k+1}^{-}H^T + R_{k+1}\right] K_{k+1}^T \end{aligned} \tag{A.25}$$

Based on the optimal Kalman gain defined in the Equation (A.24), we have

$$P_{k+1}^{-}H^T = K_{k+1}\left[HP_{k+1}^{-}H^T + R_{k+1}\right] \tag{A.26}$$

and hence, the a posteriori error covariance matrix is

$$\begin{aligned} P_{k+1} &= [I - K_{k+1}H]\, P_{k+1}^{-} - K_{k+1}\left[HP_{k+1}^{-}H^T + R_{k+1}\right] K_{k+1}^T \\ &\quad + K_{k+1}\left[HP_{k+1}^{-}H^T + R_{k+1}\right] K_{k+1}^T \\ &= [I - K_{k+1}H]\, P_{k+1}^{-} \end{aligned} \tag{A.27}$$

Appendix B

Derivation of Process Noise Covariance Matrix

In this appendix, we would like to derive the process noise covariance matrix Q. First, we suppose that $\dot{X} = AX$ where $X = \begin{bmatrix} x & \dot{x} & \ddot{x} \end{bmatrix}^T$, so that the transition matrix A is defined as follows:

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \ddot{x} \\ 0 \end{bmatrix} &= A \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \\ \Rightarrow A &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{B.1}$$

Given that the continue-time process noise covariance matrix, Q_t is defined as follows:

$$Q_t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_c \end{bmatrix} \tag{B.2}$$

where q_c is the process noise covariance for continue-time.

From the Equation (B.1), we have

$$A^2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{B.3}$$

$$\begin{aligned} A^3 &= A^2 \cdot A \\ &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{B.4}$$

$$
\begin{aligned}
e^{A\tau} &= I + A\tau + \frac{1}{2!}A^2\tau^2 + \frac{1}{2!}A^2\tau^2 + \dots \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & \tau & 0 \\ 0 & 0 & \tau \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \frac{1}{2}\tau^2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 & \tau & \frac{1}{2}\tau^2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{B.5}
$$

Similarly,

$$
e^{A^T\tau} = \begin{bmatrix} 1 & 0 & 0 \\ \tau & 1 & 0 \\ \frac{1}{2}\tau^2 & \tau & 1 \end{bmatrix} \tag{B.6}
$$

Therefore, the discrete-time process noise covariance matrix is derived as follows:

$$
\begin{aligned}
Q &= \int_0^{dt} e^{A\tau} Q_t e^{A^T\tau} d\tau \\
&= \int_0^{dt} \begin{bmatrix} 1 & \tau & \frac{1}{2}\tau^2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \tau & 1 & 0 \\ \frac{1}{2}\tau^2 & \tau & 1 \end{bmatrix} d\tau \\
&= \int_0^{dt} \begin{bmatrix} 0 & 0 & \frac{1}{2}q_c\tau^2 \\ 0 & 0 & q_c\tau \\ 0 & 0 & q_c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \tau & 1 & 0 \\ \frac{1}{2}\tau^2 & \tau & 1 \end{bmatrix} d\tau \\
&= \int_0^{dt} \begin{bmatrix} \frac{1}{4}q_c\tau^4 & \frac{1}{2}q_c\tau^3 & \frac{1}{2}q_c\tau^2 \\ \frac{1}{2}q_c\tau^3 & q_c\tau^2 & q_c\tau \\ \frac{1}{2}q_c\tau^2 & q_c\tau & q_c \end{bmatrix} d\tau \\
&= \begin{bmatrix} \frac{1}{20}q_c dt^5 & \frac{1}{8}q_c dt^4 & \frac{1}{6}q_c dt^3 \\ \frac{1}{8}q_c dt^4 & \frac{1}{3}q_c dt^3 & \frac{1}{2}q_c dt^2 \\ \frac{1}{6}q_c dt^3 & \frac{1}{2}q_c dt^2 & q_c dt \end{bmatrix}
\end{aligned} \tag{B.7}
$$

Bibliography

[1] Anoto Group AB. `http://www.anoto.com`.

[2] Hidetaka Abe. Applications expand for downsized piezoelectric vibrating gyroscopes. *Dataweek*, November 2005.

[3] Won-Chul Bang, Wook Chang, Kyeong-Ho Kang, Eun-Seok Choi, Alexey Potanin, and Dong-Yoon Kim. Self-contained Spatial Input Device for Wearable Computers. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC'03)*, pages 26–34, 2003.

[4] Robert Grover Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB exercises and solutions*. John Wiley & Sons, New York, USA, third edition, 1997.

[5] Eun-Seok Choi, Wook Chang, Won-Chul Bang, Jing Yang, Sung-Jung Cho, Jong-Koo Oh, Joon-Kee Cho, and Dong-Yoon Kim. Development of the Gyro-free Handwriting Input Device based on Inertial Navigation System (INS) Theory. In *Proceedings of SICE Annual Conference 2004 in Sapporo*, pages 1176–1181, 2004.

[6] Charles K. Chui and Guanrong Chen. *Kalman Filtering with Real-Time Applications*. Springer-Verlag Berlin Heidelberg, Germany, 1987.

[7] A. N. Cleland and M. L. Roukes. Noise processes in nanomechanical resonators. *Journal of Applied Physics*, 92(5):2758–2769, 2002.

[8] Atmel Corporation. *Atmel® ATmega32(L) 8-bit AVR® Microcontroller with 32K Bytes In-System Programmable Flash Summary*, 2006.

[9] Logitech International Inc. Corporation. `http://www.logitect.com`.

[10] Nokia Corporation. `http://www.nokia.com`.

[11] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, Reading, Massachusetts, USA, second edition, 1989.

[12] Malcolm Davis and T. O. Ellis. The RAND Tablet: A Man-Machine Graphical Communications Device. Technical Report RM-4122-ARPA, The RAND Corporation, August 1964.

[13] Zoran Djuric. Noise Sources in Microelectromechanical Systems. In *Proceedings of 22nd International Conference on Microelectronics, 2000*, volume 1, pages 85–96, 2000.

[14] Zhuxin Dong, Guanglie Zhang, Yilun Luo, Chi Chiu Tsang, Guangyi Shi, Sze Yin Kwok, Wen J. Li, Philip H. W. Leong, and Ming Yiu Wong. A Calibration Method for MEMS Inertial Sensors Based on Optical Tracking. In *Proceedings of 2007 IEEE International Conference on Nano/Micro Engineered and Molecular Systems (IEEE-NEMS 2007)*, pages 542–547, 2007.

[15] Richard P. Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics: The Definitive Edition.* Addison-Wesley, 2006.

[16] Davey T. W. Fong, Joe C. Y. Wong, Alan H. F. Lam, Raymond H. W. Lam, and Wen J. Li. A Wireless Motion Sensing System Using ADXL MEMS Accelerometers for Sports Science Applications. In *Proceedings of the 5th World Congress on Intelligent Control and Automation*, pages 5635–5640, June 2004.

[17] Mark Frank. Positioning refinement algorithm, September 2001. U.S. Patent 6292751.

[18] Thomas B. Gabrielson. Mechanical-Thermal NOise in Micromachined Acoustic and Vibration Sensors. *IEEE Transactions on Electron Devices*, 40(5):903–909, May 1993.

[19] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB.* John Wiley & Sons, second edition, 2001.

[20] Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration.* John Wiley & Sons, second edition, 2007.

[21] Analog Devices Inc. *ADXL50 Monolithic Accelerometer With Signal Conditioning Datasheet*, 1996.

[22] Analog Devices Inc. *ADXL103/ADXL203 Precision ±1.7g Single-/Dual-Axis iMEMS® Accelerometer Datasheet*, 2006.

[23] Crossbow Technology Inc. *Crossbow IMU400 6DOF Inertial Measurement Unit Datasheet.*

[24] Luidia Inc. eBeam - Interactive Whiteboard Technology. `http://www.e-beam.com`.

[25] Applied Technology Institute. Kalman filter derivation. `http://www.aticourses.com/kalman_filter.pdf`.

[26] J. B. Johnson. Thermal Agitation of Electricity in Conductors. *Physical Review*, 32(1):97–109, July 1928.

[27] Rudolf E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[28] Alan H. F. Lam, Raymond H. W. Lam, Wen J. Li, Martin Y. Y. Leung, and Yunhui Liu. Motion sensing for robot hands using mids. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, volume 3, pages 3181–3186, September 2003.

[29] Alan H. F. Lam, Martin Y. Y. Leung, Raymond H. W. Lam, Wen J. Li, and Yunhui Liu. Grasping Motion Control of Robotic Hands Using MIDS. In *Proceedings of the 2003 Chinese Intelligent Automation Conference*, pages 557–562, December 2003.

[30] Alan H. F. Lam and Wen J. Li. MIDS: GUI and TUI in Mid-Air Using MEMS Sensors. In *Proceedings of the 2002 International Conference on Control and Automation, ICCA 2002*, pages 1218–1222, June 2002.

[31] Alan H. F. Lam, Wen J. Li, Yunhui Liu, and Ning Xi. MIDS: Micro Input Devices System Using MEMS Sensors. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1184–1189, October 2002.

[32] Wen J. Li. Lectures on Topics in MEMS and Nano Robotics, 2006. `http://www.mae.cuhk.edu.hk/~ace3200`.

[33] Sanford LP. mimio - Interactive whiteboard. `http://www.mimio.com`.

[34] Murata Manufacturing Co. Ltd. *Piezoelectric Vibrating Gyroscopes (GYROSTAR®) ENC-03M Datasheet*, 2003.

[35] Murata Manufacturing Co. Ltd. Sensors from the motor car. *Murata Mail*, Spring, 2003.

[36] Wacom Co. Ltd. EMR® (Electro-Magnetic Resonance) Technology, 2006. `http://www.wacom-components.com/english/technology/emr.html`.

[37] H. Nyquist. Thermal Agitation of Electric Charge in Conductors. *Physical Review*, 32(1):110–113, July 1928.

[38] Robert M. Rogers. *Applied Mathematics in Integrated Navigation Systems*. American Institute of Aeronautics and Astronautics, Inc., Reston, Virginia, USA, second edition, 2003.

[39] Gerald L. Smith, Stanley F. Schmidt, and Leonard A. McGee. Application of statistical filter theory to the optimal estimation of position and velocity on-board a circumlunar vehicle. Technical Report NASA-TR-R-135, Ames Research Center, National Aeronautics and Space Administration, January 1962.

[40] Harold W. Sorenson, editor. *Kalman Filtering: Theory and Application*. The Institute of Electrical and Electronics Engineers, 1985.

[41] N.A. Thacker and A.J. Lacey. Tutorial: The likelihood interpretation of the kalman filter. Technical Report Tina Memo No. 1996-002, Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, May 2006.

[42] D. H. Titterton and J. L. Weston. *Strapdown Inertial Navigation Technology.* The Institution of Electrical Engineers, Stevenage, U.K., second edition, 2004.

[43] C. Verplaetse. Inertial proprioceptive devices: Self-motion-sensing toys and tools. *IBM Systems Journal*, 35(3&4), 1996.

[44] Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. In *Proceedings of SIGGRAPH 2001*, 2001.

[45] Norbert Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series.* Wiley, New York, USA, 1949.

[46] Wikipedia. Global Positioning System — Wikipedia, The Free Encyclopedia, 2007. `http://en.wikipedia.org/w/index.php?title=Global_Positioning_System&oldid=145105873`.

[47] Ruiguang Yan, Joe C. Y. Wong, Alan H. F. Lam, and Wen J. Li. E-MIDS: A Novel Game Controller for Mobile Devices. In *Proceedings of 3rd International Conference on Application and Development of Computer Games (ADCOG 2004)*, pages 54–59, February 2004.

[48] Jing Yang, Wook Chang, Won-Chul Bang, Eun-Seok Choi, Kyoung-Ho Kang, Sung-Jung Cho, and Dong-Yoon Kim. Analysis and Compensation of Errors in the Input Device Based on Inertial Sensors. In *Proceedings of the International Conference on Infromation Tehcnology: Coding and Computing (ITCC'04)*, volume 2, pages 790–796, 2004.

[49] Navid Yazdi, Farrokh Ayazi, and Kalil Najafi. Micromachined Inertial Sensors. *Proceedings of the IEEE*, 86(8), August 1998.

[50] Paul Zarchan and Howard Musoff. *Fundamentals of Kalman filtering: a practical approach.* American Institute of Aeronautics and Astronautics, Inc., Reston, Virginia, USA, second edition, 2005.

[51] Guanglie Zhang, Guangyi Shi, Yilun Luo, Heidi Wong, Wen J. Li, Philip H. W. Leong, and Ming Yiu Wong. Towards an Ubiquitous Wireless Digital Writing Instrument Using MEMS Motion Sensing Technology. In *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 795–800, 2005.

Publications

Full Length Conference Papers

- Y. Luo, C.C. Tsang, G. Zhang, Z. Dong, G. Shi, S.Y. Kwok, W.J. Li, P.H.W. Leong, M.Y. Wong, "An Attitude Compensation Technique for a MEMS Motion Sensor Based Digital Writing Instrument", *Proceedings of 2006 IEEE International Conference on Nano/Micro Engineered and Molecular Systems (IEEE-NEMS 2006)*, pp. 909-914, Zhuhai, China, 18-21 January 2006.

- C.C. Tsang, C.T. Chow, G. Zhang, Y. Luo, Z. Dong, G. Shi, S. Y. Kwok, H.Y.Y. Wong, W.J. Li, P.H.W. Leong and M.Y. Wong, "A Novel Real-Time Error Compensation Methodology for μIMU-based Digital Writing Instrument", *Proceedings of 2006 IEEE International Conference on Robotics and Biomimetics (IEEE-ROBIO 2006)*, pp. 678-681, Kunming, China, 17-20 December 2006.

- Z. Dong, G. Zhang, Y. Luo, C.C. Tsang, G. Shi, S.Y. Kwok, W.J. Li, P.H.W. Leong and M.Y. Wong, "A Calibration Method for MEMS Inertial Sensors Based on Optical Tracking", *Proceedings of 2007 IEEE International Conference on Nano/Micro Engineered and Molecular Systems (IEEE-NEMS 2007)*, pp. 542-547, Bangkok, Thailand, 16-19 January 2007.

- C.C. Tsang, G. Zhang, C.F. Chung, Z. Dong, G. Shi, W.J. Li, and P.H.W. Leong, "Handwriting Tracking based on Coupled μIMU/Electromagnetic Resonance Motion Detection", *Proceedings of 2007 IEEE International Conference on Robotics and Biomimetics (IEEE-ROBIO 2007)*, pp. 377-381, Sanya, China, 15-18 December 2007.

- Z. Dong, G. Zhang, C.C. Tsang, G. Shi, W.J. Li, P.H.W. Leong and M.Y. Wong, "μIMU-Based Handwriting Recognition Calibration by Optical Tracking", *Proceedings of 2007 IEEE International Conference on Robotics and Biomimetics (IEEE-ROBIO 2007)*, pp. 382-387, Sanya, China, 15-18 December 2007.