



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학박사 학위논문

Fusion of the Integration and Parametric
Approaches for PDR in Multiple Poses of
Smartphone

적분 및 매개변수 기법 융합을 이용한
스마트폰 다중 동작에서 보행 항법

2020 년 8 월

서울대학교 대학원
기계항공공학부

박 소 영

Fusion of the Integration and Parametric Approaches for PDR in Multiple Poses of Smartphone

적분 및 매개변수 기법 융합을 이용한
스마트폰 다중 동작에서 보행 항법

지도교수 박 찬 국

이 논문을 공학박사 학위논문으로 제출함

2020 년 5 월

서울대학교 대학원

기계항공공학부

박 소 영

박소영의 공학박사 학위논문을 인준함

2020 년 6 월

위 원 장

김 유 단

(인)

부위원장

백 찬 국

(인)

위 원

기 창 린

(인)

위 원

송 진 우

(인)

위 원

조 성 윤

(인)



Fusion of the Integration and Parametric Approaches for PDR in Multiple Poses of Smartphone

by

Soyoung Park

Submitted to the Department of Mechanical and Aerospace Engineering
in partial fulfillment of the requirements for the degree of

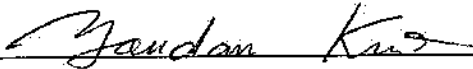
DOCTOR OF PHILOSOPHY

In Aerospace Engineering at the

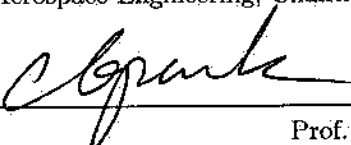
SEOUL NATIONAL UNIVERSITY

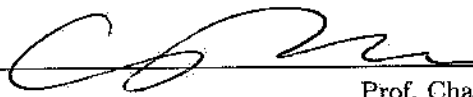
June 2020


Accepted by

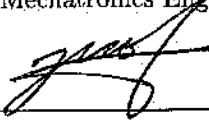

Prof. Youdan Kim
Dept. of Aerospace Engineering, Chairman of Committee

Certified by


Prof. Chan Gook Park
Dept. of Aerospace Engineering, Principal Advisor


Prof. Changdon Kee
Dept. of Aerospace Engineering


Prof. Jin Woo Song
Dept. of Intelligent Mechatronics Engineering, Sejong University


Prof. Seong Yun Cho
School of Mechanical and Automotive Engineering, Kyungil University

Abstract

Fusion of the Integration and Parametric Approaches for PDR in Multiple Poses of Smartphone

Soyoung Park

Department of Mechanical and Aerospace Engineering

The Graduate School

Seoul National University

In this dissertation, an IA-PA fusion-based PDR (Pedestrian Dead Reckoning) using low-cost inertial sensors is proposed to improve the indoor position estimation. Specifically, an IA (Integration Approach)-based PDR algorithm combined with measurements from PA (Parametric Approach) is constructed so that the algorithm is operated even in various poses that occur when a pedestrian moves with a smartphone indoors. In addition, I propose an algorithm that estimates the device attitude robustly in a disturbing situation by an ellipsoidal method. In addition, by using the machine learning-based pose recognition, it is possible to improve the position estimation performance by varying the measurement update according to the poses.

First, I propose an adaptive attitude estimation based on ellipsoid technique to accurately estimate the direction of movement of a smartphone device. The AHRS (Attitude and Heading Reference System) uses an accelerometer and a magnetometer as measurements to calculate the attitude based on the gyro and

to compensate for drift caused by gyro sensor errors. In general, the attitude estimation performance is poor in acceleration and geomagnetic disturbance situations, but in order to effectively improve the estimation performance, this dissertation proposes an ellipsoid-based adaptive attitude estimation technique. When a measurement disturbance comes in, it is possible to update the measurement more accurately than the adaptive estimation technique without considering the direction by adjusting the measurement covariance with the ellipsoid method considering the direction of the disturbance. In particular, when the disturbance only comes in one axis, the proposed algorithm can use the measurement partly by updating the other two axes considering the direction. The proposed algorithm shows its effectiveness in attitude estimation under disturbances through the rate table and motion capture equipment.

Next, I propose a PDR algorithm that integrates IA and PA that can be operated in various poses. When moving indoors using a smartphone, there are many degrees of freedom, so various poses such as making a phone call, texting, and putting a pants pocket are possible. In the existing smartphone-based positioning algorithms, the position is estimated based on the PA, which can be used only when the pedestrian's walking direction and the device's direction coincide, and if it does not, the position error due to the mismatch in angle is large. In order to solve this problem, this dissertation proposes an algorithm that constructs state variables based on the IA and uses the position vector from the PA as a measurement. If the walking direction and the device heading do not match based on the pose recognized through machine learning technique, the position is updated in consideration of the direction calculated using PCA (Principal Component Analysis) and the step length obtained through the PA. It can be operated robustly even in various poses that occur.

Through experiments considering various operating conditions and paths,

it is confirmed that the proposed method stably estimates the position and improves performance even in various indoor environments.

Keywords: Indoor navigation, Pedestrian Dead Reckoning (PDR), Extended Kalman Filter (EKF), Ellipsoidal method-based adaptive attitude estimation, IA (Integration Approach)-PA (Parametric Approach) fusion method

Student Number: 2013-20674

Contents

Abstract	i
Contents	vii
List of Tables	ix
List of Figures	xiii
Chapter 1 Introduction	1
1.1 Motivation and Background	1
1.2 Objectives and Contribution	5
1.3 Organization of the Dissertation	6
Chapter 2 Pedestrian Dead Reckoning System	8
2.1 Overview of Pedestrian Dead Reckoning	8
2.2 Parametric Approach	9
2.2.1 Step detection algorithm	11
2.2.2 Step length estimation algorithm	13
2.2.3 Heading estimation	14
2.3 Integration Approach	15
2.3.1 Extended Kalman filter	16
2.3.2 INS-EKF-ZUPT	19
2.4 Activity Recognition using Machine Learning	21

2.4.1	Challenges in HAR	21
2.4.2	Activity recognition chain	22
Chapter 3	Attitude Estimation in Smartphone	26
3.1	Adaptive Attitude Estimation in Smartphone	26
3.1.1	Indirect Kalman filter-based attitude estimation	26
3.1.2	Conventional attitude estimation algorithms	29
3.1.3	Adaptive attitude estimation using ellipsoidal methods . .	30
3.2	Experimental Results	36
3.2.1	Simulation	36
3.2.2	Rate table experiment	44
3.2.3	Handheld rotation experiment	46
3.2.4	Magnetic disturbance experiment	49
3.3	Summary	53
Chapter 4	Pedestrian Dead Reckoning in Multiple Poses of a Smartphone	54
4.1	System Overview	55
4.2	Machine Learning-based Pose Classification	56
4.2.1	Training dataset	57
4.2.2	Feature extraction and selection	58
4.2.3	Pose classification result using supervised learning in PDR	62
4.3	Fusion of the Integration and Parametric Approaches in PDR . .	65
4.3.1	System model	67
4.3.2	Measurement model	67
4.3.3	Mode selection	74
4.3.4	Observability analysis	76
4.4	Experimental Results	82

4.4.1	AHRS results	82
4.4.2	PCA results	84
4.4.3	IA-PA results	88
4.5	Summary	100
Chapter 5 Conclusions		103
5.1	Summary of the Contributions	103
5.2	Future Works	105
국문초록		125
Acknowledgements		127

List of Tables

Table 3.1	Comparison of adaptation rules	35
Table 3.2	Sensor specifications	36
Table 3.3	Simulation cases for trajectory #1 and #2	37
Table 3.4	Attitude error in simulation #1	38
Table 3.5	Attitude error in simulation #2	41
Table 3.6	Attitude error during movement in rate table	44
Table 3.7	Attitude error during hand movement	48
Table 3.8	Attitude error during magnetic disturbance	49
Table 4.1	Sensor specifications for Xsens MTw	58
Table 4.2	Selected features from mRMR (descending order)	61
Table 4.3	Accuracy results-SVM	63
Table 4.4	Accuracy results-kNN	65
Table 4.5	Attitude error compared with Xsens MTw	83
Table 4.6	Gyro bias error compared with stationary data	83
Table 4.7	Initial gyro bias set	83
Table 4.8	Walking direction by PCA results	85
Table 4.9	Position results for trajectory #1	93
Table 4.10	Position results for trajectory #2	98

List of Figures

Figure 2.1	IA and PA-based PDR difference in 3D [1]	9
Figure 2.2	Definition of step and stride	10
Figure 2.3	PA-based PDR components	10
Figure 2.4	Step detection algorithms	11
Figure 2.5	Step phase classification	19
Figure 2.6	Stance phase detection block diagram	20
Figure 2.7	Typical ARC from wearable sensors [2]	22
Figure 2.8	Classifiers	25
Figure 3.1	Proposed ellipsoidal method-based adaptive AHRS diagram	31
Figure 3.2	Adaptive measurement covariance process	32
Figure 3.3	Simulation trajectory #1	37
Figure 3.4	Simulation results for trajectory #1	39
Figure 3.5	Simulation trajectory #2	40
Figure 3.6	Simulation results for trajectory #2	43
Figure 3.7	The sensor on the rate table	44
Figure 3.8	Rate table experiment results	45
Figure 3.9	The sensor with Vicon markers	46
Figure 3.10	Handheld experiment results	48

Figure 3.11	Magnetic disturbance experiment results for constant disturbance	51
Figure 3.12	Magnetic disturbance experiment results for changing disturbance	52
Figure 4.1	Overall IA-PA fusion PDR algorithm	56
Figure 4.2	Target poses: chest, text, swing, trouser	57
Figure 4.3	Overall pose recognition algorithm	57
Figure 4.4	Selected features according to poses	62
Figure 4.5	Classification accuracy by the number of feature ordered by mRMR	64
Figure 4.6	Measurement update	69
Figure 4.7	Measurement updating mode	74
Figure 4.8	Walking scenarios and corresponding modes	75
Figure 4.9	Mode decision flow	75
Figure 4.10	Estimated gyro bias error in different initial values	84
Figure 4.11	PCA and acceleration (n-frame) for 1 stride	86
Figure 4.12	PCA results in box whisker plot	87
Figure 4.13	Trajectory #1	88
Figure 4.14	Step detection results	89
Figure 4.15	Walking direction difference of PCA and AHRS	91
Figure 4.16	Estimated position	92
Figure 4.17	Trajectory #2	93
Figure 4.18	Position and attitude results for shirt pocket pose	94
Figure 4.19	Position and attitude results for trouser pocket pose	96
Figure 4.20	Position and attitude results for swing pose	97
Figure 4.21	Position and attitude results for text only pose	99

Figure 4.22 Position result for trajectory #3 102

Chapter 1

Introduction

1.1 Motivation and Background

Indoor navigation has been actively studied as the growing number of people desires to locate themselves through smart devices. Unlike outdoor navigation, GPS is not available in an indoor environment due to inadequate coverage of the satellite signal and multipath errors, various alternatives are presented with additional sensors or methods [3]. A direct sensing-based localization tracks the position of a pedestrian by the sensing of identifiers or tags installed in the environment before experiments [4]. Those include RFID (Radio-Frequency IDentification), IR (InfraRed), Ultrasound, and Bluetooth. RFID stores and retrieves data by means of electromagnetic transmission to an RF compatible integrated circuit [5]. The light and small tags of a RFID positioning system are beneficial when used in conjunction with many applications, but numerous types of infrastructure are required for accurate positioning [6]. IR localization uses IR transmitters installed in a known location, with a unique ID broadcast by each transmitter [7]. Although IR-based positioning is accurate and uses small, and light-weight components, the system is expensive and sensitive to interference from natural and artificial light [8]. Ultrasound uses emitters which are installed in the infrastructure to broadcast ultrasound waves, and a receiver carried by a user receives these signals from the two closest emitters to

determine his location [9]. The system is inexpensive and has a wide coverage area; however, ultrasound-based positioning systems often experience blocked or reflected signals or are affected by noise sources and low measurement accuracy [10]. A wireless standard for WPANs (Wireless Personal Area Networks), Bluetooth, is light and ubiquitous given its use in numerous devices, such as laptops, desktops, and mobile phones [11]. Despite its advantages of reusability and the fact that Bluetooth is low-cost and powerful technology, its accuracy is low, and it is sensitive to environmental changes [12].

With regard to pattern-recognition-based localization, well-known examples are WLAN fingerprinting and vision [4]. WLAN (Wireless Local Area Network) can also be used to estimate the location of a user with a device, and the infrastructure is already available for WLAN technology, which leads to low installation costs [13]. However, various obstacles, including walls, furniture, and doors, degrade the performance accuracy of WLAN, and reducing the power used by WLAN devices is another important issue [3]. Captured images of environments are matched with a database to determine the position and orientation of the user when using vision techniques [14]. The requirement of high computing power for image matching is one of the main challenges affecting this method [15]. The methods that fuse the information from cameras and IMUs (Inertial Measurement Unit) for navigation purpose are often called VINS (Visual-Inertial Navigation System) [16–18], and it has been extensively studied and applied in many industrial fields in recent years [19]. Indoor navigation with the VINS degrades in rapid motion and lens occlusion on the camera measurements.

Pedestrian positioning using inertial sensors has been used in a wide range of fields, including ambulatory human motion analysis. Micromachined gyroscopes and accelerometers are used in many applications, such as monitoring daily

living activities [20–22], evaluating internal mechanical workload in ergonomic studies [23–26], measuring nervous system disorders [27–30], and mixed and augmented reality [31–33].

DR (Dead Reckoning)-based localization uses sensors attached to the users to estimate relative positions based on the previous or known position [34]. A benefit of the DR approach is low installation cost and does not require any additional sensors, but the accumulation errors constitute a significant problem of this technique [35]. Localization using DR is primarily categorized as IA and PA. For the tracking with shoe-mounted inertial sensors, IA-based system that estimates the current position by integrating acceleration is applicable. It is because one of the strong pseudo-measurements, ZUPT (Zero velocity UPdaTe), uses the velocity of a foot being zero during the stance phase [36,37]. When the sensor is held on hand such as smartphones, on the other hand, the PA system that estimates position by step detection, step length estimation, and heading estimation is applied instead of the IA. To estimate the heading of a device is an essential part of the PA, and the attitude estimation using the angular rate from the gyroscope, the specific force from the accelerometer, magnetic field from the magnetometer are called AHRS. Using the characteristics that gyro measure has a low-frequency component, and accelerometer and magnetometer have high-frequency one, the AHRS combines those sensors with filtering methods such as complementary filter and Kalman filter. However, its performance degrades, especially when the sensor is moving fast or exposed to a magnetic disturbance.

In addition, handheld smartphones are usually unrestricted and often have device heading changes that do not match the direction of walking. In order to remove the heading offset between them, some researchers have attempted to solve this problem [38–43]. In [38], the various PCA-based walking direction estimation methods are compared with broad experimental study in case of

pocket. [39] uses a rotating axis for heading estimation dealing with swing, call, trouser pocket poses in the smartphone. Tian proposes adaptive offset compensation using the heading in straight holding mode under the swing, holding, and trouser pocket [40]. PCA-GA (Global Acceleration) method is proposed with the combination of TRIAD heading estimation in [41]. In addition, Deng proposes heading estimation using PCA [42, 43]. The above methods use PCA or fixed offset angle to solve the heading mismatch after the pose from the machine learning such as DT (Decision Tree), SVM (Support Vector Machine), or FSM (Finite State Machine). In the PA-based PDR system, the small heading difference leads to large position error, whereas the PCA-based methods vary heading values significantly depending on the data distribution between two steps. In addition, PDR components of step detection, step length, and heading estimation are different in terms of the classified poses, which also means that the misclassification leads to position errors in PDR.

In this dissertation, there are two contributions. The purpose of those algorithms is to implement an accurate and consistent PDR algorithm including pose changes while walking. To start with, accurate device attitude estimation using the ellipsoidal method is proposed. While walking indoors with a smartphone, acceleration and magnetic disturbance frequently occur, which degrades the attitude estimation performance. The proposed ellipsoidal method inflates the measurement covariance by considering the direction of measurements. This allows estimating attitude more accurately than the adaptive algorithms not considering the measurement direction. Next, in order to overcome the limitations of conventional PA-based PDR in smartphone, the integration of IA and PA PDR is proposed. Specifically, pedestrian navigation algorithm in IA is implemented using the attitude and step length from the PA. With the help of the estimated device attitude and classified poses by machine learning method, it

is possible to recognize the heading and walking direction mismatch situations. Then, the walking direction acquired from the PCA of tangential acceleration data is used to update the IA states. The proposed algorithm is operated even in various poses that occur when a pedestrian moves with a smartphone indoors.

The proposed algorithm is for real-time indoor pedestrian positioning systems. It can also be used in various fields such as health care systems, mixed and augmented reality and motion capture systems.

1.2 Objectives and Contribution

The goal of this dissertation research is an attempt to improve the performance of smartphone-based PDR systems using low-cost IMUs. The proposed algorithm uses the ellipsoid method for adaptive attitude estimation and fusion of the IA and PA to estimate the accurate position of a pedestrian under various poses. The original contributions of this dissertation are:

1. Adaptive attitude estimation using ellipsoidal method
 - For a consistent and accurate estimation, the adaptive attitude algorithm using an ellipsoidal method is proposed.
 - Residual vectors are first acquired to deal with measurement errors from the acceleration and magnetic disturbance in AHRS.
 - Comparing the residual vectors and measurement noises, the covering ellipsoid, measurement covariance, is estimated.
 - The inflated measurement covariance is used in EKF to estimate attitude errors and gyro bias errors.
 - The proposed attitude algorithm works accurately with the rate table, hand rotation with visual markers, and magnetic disturbing experiments.

2. Fusion of integration and parametric approach in PDR for multiple poses

- IA and PA-based PDR algorithm for position estimation is designed.
- PDR states of IA are estimated using measurements from PA and PCA.
- PCA of tangential acceleration is performed when the walking direction and device attitude do not match to find the walking direction of poses.
- The four poses (text, shirt pocket, trouser pocket, and swing) are classified using machine learning techniques to help find the measurement updating mode.
- EKF uses ZUPT, AHRS, PA, and PCA to estimate and correct the error states of IA.
- The proposed algorithm, as a result, ensures observability of error states and position accuracy.
- The proposed algorithms can be applied in various fields such as smartphone users in the buildings, first responders, virtual and augmented reality.

1.3 Organization of the Dissertation

The remainder of the dissertation is organized as follows.

Chapter 2 gives an overview of the personal navigation system. IMU-based pedestrian positioning methods are summarized. Conventional IA and PA-based PDR algorithms and components are provided. In addition, machine learning techniques for pose classification are described.

In chapter 3, a new approach to the adaptive attitude estimation using

the ellipsoidal method is proposed. This algorithm inflates the measurement covariance according to the residual vector, taking into account the residuals of the accelerometer and magnetometer measurements. The proposed algorithm is tested using rate table and visual marker references and shows its effectiveness under various conditions.

Chapter 4 presents the fusion of IA and PA-based PDR system. Parametric step length and estimated device heading correct the error states in the IA-based PDR. If the walking direction and device heading do not match, the walking direction derived from the principal components of acceleration in navigation frame is used for the measurement update. Pose classification by machine learning and heading difference are used to find the correct measurement update mode. The estimated position is tested with the trajectory including four different poses.

Finally, in chapter 5, the major contributions of this dissertation is summarized and an overview of the future research directions are presented.

Chapter 2

Pedestrian Dead Reckoning System

2.1 Overview of Pedestrian Dead Reckoning

PDR is one of the inertial sensor-based pedestrian navigation systems. A MEMS (Micro-Electro-Mechanical Systems) IMU with a 3-axis accelerometer, gyroscope and magnetometer is used to estimate the location of a pedestrian in the PDR. Recently, MEMS technology has made it possible to produce inexpensive lightweight and compact inertial sensors with low power consumption. These are desirable characteristics for portable navigation systems, but the accuracy of MEMS IMUs is relatively low. To overcome this disadvantage, various types of PDR algorithms have been developed.

PDR is a dead reckoning system that makes the assumption that the position of a pedestrian is changed by steps. Based on this, the PDR estimates the location of the pedestrian by observing the movement of steps. Depending on the position of the installed sensor, the PDR can be classified into the IA and PA as in Fig. 2.1. The IA-based PDR system integrates an inertial sensor to calculate the position and uses measurements such as zero velocity update or contact phase velocity update to prevent an increase in exponential error from the integral [35–37, 44, 45]. The system needs to find the correct phase for zero velocity or contact phase measurements. Therefore, it is applied to the sensor attached to the foot. However, the PDR with PA uses a parametric method

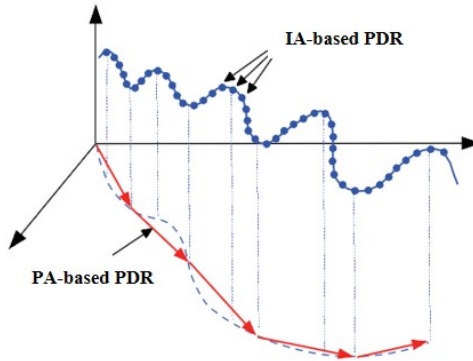


Figure 2.1: IA and PA-based PDR difference in 3D [1]

such as walking frequency, acceleration variance, etc. to estimate the current position by estimating the distance from the previous step [46–48]. When a step is detected, it is only required to calculate the current direction and the length between them, so the mounting position of the sensor is irrelevant as long as the direction of the device and the gait is matched. A representative application of the PA scheme is waist-mounted [49, 50] and a handheld smartphone [51–55], but the prerequisite is matching in device heading and walking direction.

This chapter describes the well-known approaches of PDR. The PA-based PDR algorithm that can be used for mobile devices are first described in Section 2.2. The IA-based PDR with EKF (Extended Kalman Filter) and its application to PDR are described in Section 2.3. The following sections provide more information.

2.2 Parametric Approach

In this paper, the PDR methods are largely classified as IA and PA according to the sensor location. The PA-based PDR is pedestrian-dependent method and estimates its position by acquiring heading and distance between steps or a

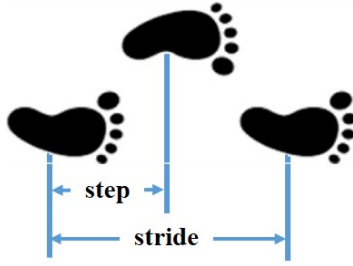


Figure 2.2: Definition of step and stride

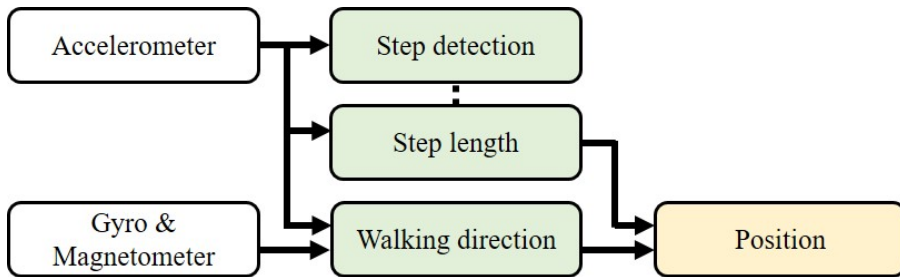


Figure 2.3: PA-based PDR components

stride. As depicted in Fig. 2.2, a step is the distance between the foot moving forward in front of the other one, and a stride is the distance between two successive steps of the same foot [56].

In general, the PA-based PDR algorithm is composed of a step detection, a step length estimation, and a heading estimation. The PA-PDR is generally applied to mobile devices that can be located in various locations on the human body. When the initial location is known, the PDR system estimates the relative position from the previous step. Fig. 2.3 shows a simple block diagram of the PA-based PDR algorithm, with each component described in a subsection to follow.

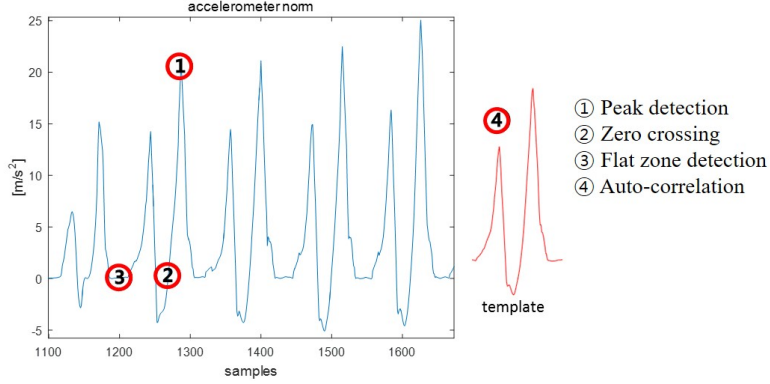


Figure 2.4: Step detection algorithms

2.2.1 Step detection algorithm

The first step in PA-based PDR is to identify steps. Step detection is usually an easy problem, but if you have false or miss detections, or if your smartphone’s action varies, you may encounter significant errors when estimating overall walking distance. Even if the step length algorithm is accurate, errors when estimating the position can be considerable due to inaccuracies in the step detection process. Therefore, accurate step detection is the basis on which an accurate estimation of the position can be made in the PDR method.

As shown in the Fig. 2.4, there are several existing step detection techniques, including peak detection method, zero-crossing detection method, stance phase detection method, and auto-correlation method. These methods use the outputs of accelerometers and gyros [34, 45, 57]. The peak detection method has the advantage of being able to detect a step accurately at the moment of a heel strike, but there is a high possibility of misdetection in case local minimum for the various speed of a pedestrian [41, 58, 59]. The zero-crossing method is easy to implement the algorithm to the system but sensitive to the jitter around the threshold [60, 61]. The stance phase detection method which is also called flat

zone detection works well when the sensor is attached to the foot. However, detecting a zero velocity zone is hardly made under the condition that user is carrying a smartphone on hand or backpack. Lastly, correlating using step templates is highly dependent on step and subject speed [62]. There are other various step detection methods such as FSM [63], fast Fourier transform [64,65], continuous wavelet transform [66], and dynamic time warping [67].

When it comes to navigate a pedestrian with a smartphone or tablet, placements of the device should be considered for step detection. According to the [68], the possible placements for unconstrained smartphone include hand-held, texting, calling, trouser back and front pocket, handbag, backpack, and shirt pocket [69], and those are usually classified through the machine learning techniques.

The accelerometer attached to the body is affected not only by the acceleration of the body, but also by other factors such as noise and accelerometer bias, gravity, etc. In this dissertation, a three-axis acceleration norm is used for the step detection algorithm as (2.1).

$$f_{norm} = \sqrt{f_x^2 + f_y^2 + f_z^2} \quad (2.1)$$

where f_{norm} is the acceleration norm. f_x, f_y, f_z denote the output of 3-axis accelerometer in x-axis, y-axis and z-axis, respectively.

In addition, sliding window sum is used to reduce noise as follows.

$$SWS(k) = \sum_{t=k-N+1}^k f_{norm}(t) \quad (2.2)$$

where SWS represents the sliding window sum and N represents the window size. The window size is usually set smaller than the duration of the detected phase.

In case of the foot-mounted PDR, the stance phase detection is alternatively processed while updating with the zero velocity measurements. It is detected using norm of gyro [70, 71], the local variance of acceleration [72], and the variances of simply modified signal features [73].

2.2.2 Step length estimation algorithm

Step length estimation can be divided into two main classes: direct methods and indirect methods [74]. The direct methods estimate the stride length directly through integration. It has been found to apply this method to sensors mounted on shank and pelvis [75–77]. The disadvantages of this method are obvious: the sensor error must be accurately compensated and the attitude must be calculated accurately. Otherwise, errors accumulate quickly due to accumulation of errors by integration.

On the other hand, the indirect methods estimate step length using a model. This method is also divided into geometric model and statistical model. The first uses the biomechanical characteristics of the human body, so it is easy to understand the proposed step length model. This includes inverted pendulum model and empirical model. The pendulum model uses the relationship between step length (SL), vertical displacement (H) and leg length (L) as (2.3).

$$SL = 2\sqrt{2LH - H^2} \quad (2.3)$$

The empirical step length models are also widely used as (2.4).

$$SL = K \sqrt[4]{f_{vert,max} - f_{vert,min}} \quad (2.4)$$

where K is design parameter and f_{vert} is a vertical acceleration. The above step length estimation using biomechanical model is usually applied in waist-mounted sensors. This also indicates that the model above is accurate when sensor is in the center of mass and has incorrect limits in other placements.

In case of statistical regression methods commonly used for a smartphone, variables such as walking frequency (WF) and acceleration variance (AV) are usually used, and the relationships among variables are estimated through parametric and non-parametric techniques. Linear regression model is one of the representative methods in the parametric approach (2.5) [48, 69, 78–80].

$$SL = \alpha \cdot WF + \beta \cdot AV + \gamma \quad (2.5)$$

where α, β , and γ are the coefficients to be determined.

This method is accurate when modeled for a specific person, but it has the disadvantage that large errors may occur when tested for various people. Another limitation is that modeling requires a wide range of variables and steps.

Non-parametric technique uses learning techniques to estimate step length. In general, it performs better than the parametric method, but is more likely to be overfitting and requires a larger data and less information about features. The methods found in the literature are Gaussian process regression [81], artificial neural networks [82], and so on.

2.2.3 Heading estimation

To estimate the heading of a device is an essential part of PA, and the attitude estimation using the angular rate from the gyroscope, the specific force from the accelerometer, magnetic field from magnetometer are called AHRS. As long as there are no acceleration and magnetic disturbance, roll and pitch for accelerometers and yaw for magnetometers are calculated as (2.6), (2.7) and (2.8), respectively.

$$\phi_{acc} = \tan^{-1}(f_y/f_z) \quad (2.6)$$

$$\theta_{acc} = \tan^{-1}(f_x / \sqrt{f_y^2 + f_z^2}) \quad (2.7)$$

$$\psi_{mag} = \tan^{-1} \frac{-m_y c\phi + m_z s\phi}{m_x c\theta + m_y s\theta s\phi + m_z s\theta c\phi} \quad (2.8)$$

where $s\phi, s\theta, c\phi, c\theta$ are $\sin\phi, \sin\theta, \cos\phi, \cos\theta$, respectively.

Using the characteristics that gyro measure has a low-frequency component, and accelerometer and magnetometer have high-frequency one, AHRS combines those sensors with filtering methods. However, its performance degrades, especially when the sensor is moving fast or exposed to a magnetic disturbance in hand. More details are covered in chapter 3.

2.3 Integration Approach

The IA-based PDR is a system that tracks the position by estimating the entire 3D trajectory of the sensor at a given moment. 3-axis accelerometers and gyroscopes are used to track orientation and position changes [83]. In the strap-down configuration used by pedestrians, the sensors are joined in a rigid package and securely attached to the body. I describe the navigation reference frame (with axes in horizontal and vertical planes), the sensor reference frame (3 mutually perpendicular measuring axes pointing in any world direction) and the body reference frame (3 mutually perpendicular measuring axes with sensors attached).

In robotics, the attitude of the sensor can often be constrained, for example, so that the sensor z-axis coincides with the vertical world axis. The tracking position subtracts the gravitational signal from the vertical accelerometer signal and performs a double integral for the rest of the 3D acceleration (i.e., integrating once in speed and twice in displacement). However, in a PDR situation,

the sensor is not likely to be axially aligned, but will continue to rotate about the world frame during the gait cycle. Therefore, it is necessary to track the rotation of the sensor using the angular velocity provided by the gyroscope. This introduces a third integration for each location update.

Inevitably, measurement errors are present in the sensor data, and their triple integration potentially leads to an increase in cubic time (drift). INS (Inertial Navigation System) in aviation, marine and military uses highly accurate sensors that keep the source of the error very small and allow tracking for many hours. These are too bulky and expensive for PDR, so MEMS technology should be used instead. MEMS sensors are small and portable, but they can cause more serious errors. Open loop integration of MEMS inertial sensors is only possible for 1-2 minutes before drift is dominated [34].

The strap-down inertial navigation algorithm has been well studied, and the standard approach to limit drift uses the EKF in complementary or indirect form, so the filter directly tracks errors in system state, not system state itself. A 15-state model is commonly used: 3 states for position, velocity and attitude errors, 6 states for modeling accelerometer and gyroscope bias, respectively [84, 85].

2.3.1 Extended Kalman filter

Theoretically, the Kalman Filter is an estimator for a linear-quadratic problem, which is a problem of estimating the instantaneous state of a linear dynamic system disturbed by white noise. The resulting estimator is statistically optimal with respect to the quadratic function of the estimation error.

However, the typical Kalman filter considers the linear filter of the linear system. Unfortunately, there is no linear system. Nonlinear filtering can be a difficult and complex subject, but some nonlinear estimation methods have be-

come widespread. These technologies include nonlinear extension of Kalman filters, unscented filtering, and particle filtering. In this paper, the signal shows a nonlinear system. However, PDR systems usually take into account real-time operation, which requires a fast operation time. Also, the system model we thought of x can be easily linearized, which will be presented later. Therefore, EKF is suitable for sensor error correction. In this chapter, discrete time EKF is derived taking into account discrete time dynamics and discrete time measurements. Assume we have a system model as follows.

$$\begin{aligned}
\mathbf{x}_k &= \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\
\mathbf{z}_k &= \mathbf{h}_k(\mathbf{x}_k, \nu_k) \\
\mathbf{w}_k &\sim (\mathbf{0}, \mathbf{Q}_k) \\
\nu_k &\sim (\mathbf{0}, \mathbf{R}_k)
\end{aligned} \tag{2.9}$$

We perform a Taylor series expansion of the state equation around $\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1}^+$ and $\mathbf{w}_{k-1} = \mathbf{0}$ to obtain the following:

$$\begin{aligned}
\mathbf{x}_k &= \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \mathbf{0}) + \left. \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1}^+} (\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^+) + \left. \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{w}} \right|_{\hat{\mathbf{x}}_{k-1}^+} \mathbf{w}_{k-1} \\
&= \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \mathbf{0}) + \mathbf{F}_{k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^+) + \mathbf{L}_{k-1} \mathbf{w}_{k-1} \\
&= \mathbf{F}_{k-1} \mathbf{x}_{k-1} + [\mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \mathbf{0}) - \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1}^+] + \mathbf{L}_{k-1} \mathbf{w}_{k-1} \\
&= \mathbf{F}_{k-1} \mathbf{x}_{k-1} + \tilde{\mathbf{u}}_{k-1} + \tilde{\mathbf{w}}_{k-1}
\end{aligned} \tag{2.10}$$

\mathbf{F}_{k-1} and \mathbf{L}_{k-1} are defined by the above equation. The known signal $\tilde{\mathbf{u}}_k$ and the noise signal $\tilde{\mathbf{w}}_k$ are defined as follows:

$$\begin{aligned}
\tilde{\mathbf{u}}_k &= \mathbf{f}_k(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0}) - \mathbf{F}_k \hat{\mathbf{x}}_k^+ \\
\tilde{\mathbf{w}}_k &\sim (\mathbf{0}, \mathbf{L}_k \mathbf{Q}_k \mathbf{L}_k^T)
\end{aligned} \tag{2.11}$$

We linearize the measurement equation around $\mathbf{x}_k = \hat{\mathbf{x}}_k^-$ and $\boldsymbol{\nu}_k = \mathbf{0}$ to obtain the following.

$$\begin{aligned}
\mathbf{z}_k &= \mathbf{h}_k(\hat{\mathbf{x}}_k^-, \mathbf{0}) + \left. \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^-} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \left. \frac{\partial \mathbf{f}_k}{\partial \boldsymbol{\nu}} \right|_{\hat{\mathbf{x}}_k^-} \boldsymbol{\nu}_k \\
&= \mathbf{h}_k(\hat{\mathbf{x}}_k^-, \mathbf{0}) + \mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \mathbf{M}_k \boldsymbol{\nu}_k \\
&= \mathbf{H}_k \mathbf{x}_k + [\mathbf{h}_k(\hat{\mathbf{x}}_k^-, \mathbf{0}) - \mathbf{H}_k \hat{\mathbf{x}}_k^-] + \mathbf{M}_k \boldsymbol{\nu}_k \\
&= \mathbf{H}_k \mathbf{x}_k + \tilde{\mathbf{z}}_k + \tilde{\boldsymbol{\nu}}_k
\end{aligned} \tag{2.12}$$

\mathbf{H}_k and \mathbf{M}_k are defined by the above equation. The known signal $\tilde{\mathbf{z}}_k$ and the noise signal $\tilde{\boldsymbol{\nu}}_k$ are defined as

$$\begin{aligned}
\tilde{\mathbf{z}}_k &= \mathbf{h}_k(\hat{\mathbf{x}}_k^-, \mathbf{0}) - \mathbf{H}_k \hat{\mathbf{x}}_k^- \\
\tilde{\boldsymbol{\nu}}_k &\sim (\mathbf{0}, \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^T)
\end{aligned} \tag{2.13}$$

There are a linear state space system of (2.10) and a linear measurement of (2.12). In other words, we can estimate the state using the standard Kalman filter equation. The following equations are for discrete time EKF.

$$\begin{aligned}
\mathbf{P}_k^- &= \mathbf{F}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{F}_{k-1}^T + \mathbf{L}_{k-1} \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^T \\
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^T)^{-1} \\
\hat{\mathbf{x}}_k^- &= \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \mathbf{0}) \\
\mathbf{z}_k &= \mathbf{h}_k(\hat{\mathbf{x}}_k^-, \mathbf{0}) - \mathbf{H}_k \hat{\mathbf{x}}_k^- \\
\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- - \tilde{\mathbf{z}}_k) \\
&= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-, \mathbf{0})) \\
\mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-
\end{aligned} \tag{2.14}$$

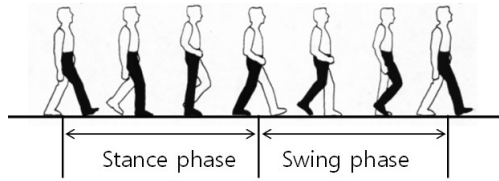


Figure 2.5: Step phase classification

2.3.2 INS-EKF-ZUPT

IA-based PDR, which typically used for a foot-mounted IMU, is based on an INS with EKF. The ZUPT-aided EKF mounted on the foot can be used to estimate the position of the INS, as it is possible to estimate the error of the INS using the assumption that the shoes cling to the floor during the stance phase. The key idea of INS-EKF-ZUPT is to estimate the accumulated INS error by IMU sensor bias and white noise using ZUPT-aided EKF. The EKF is updated with a speed measurement using ZUPT whenever the foot is on the ground.

Stance phase detection

Gait motion can be largely divided into stance and swing phases as shown in Fig. 2.5. The stance phase detection algorithm detects the attachment between the shoe and the ground and uses the signal features to find the stance and swing phase while walking. In general, the attitude phase algorithm determines the phase based on the gyroscope and accelerometer signals [70,71]. At [70], the attitude phase is determined using the gyroscope output standard. In [71], positional phase is detected using local dispersion of acceleration. One of the stance phase detection algorithms is based on the variance of the simply modified signal characteristics [72], as shown in Fig. 2.6. Fig. 2.6 shows a block diagram

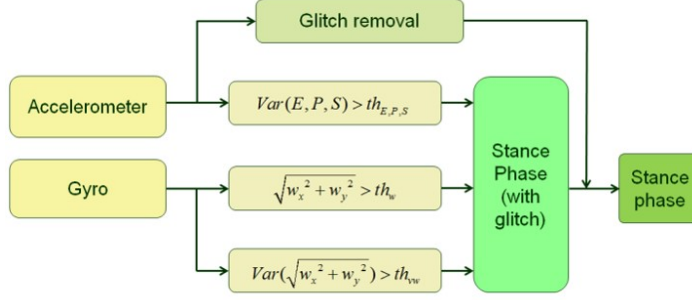


Figure 2.6: Stance phase detection block diagram

of the stance phase detection procedures. The stance phase is simply detected using a modified signaling function and glitch removal. When the sensor is attached to the shoe by defining the x-axis in the forward direction, the y-axis in the upper direction, and the z-axis in the right direction, the x-axis and z-axis acceleration output values are used, which change significantly as the pedestrian moves. The procedure is as follows. First, three modified accelerometer signals (energy, product, and sum) are used to detect stance phase. As given by (2.15), energy, product, and sum of the signals are calculated. Next, the local variance of energy, product, and sum is calculated, respectively. If each local variance is below the threshold, that particular condition is indicated by 1, which means the situation meets the condition, as in equation (2.16). However, because the three conditions only use the x and z axis accelerometer outputs, the attitude phase algorithm cannot be used for sidewalking, crawling, descending stairs, ascending stairs, and other types of movements.

$$\begin{aligned}
 Energy &= \text{sqrt}(f_x^2 + f_z^2) \\
 Product &= f_x \cdot f_z \\
 Sum &= f_x + f_z
 \end{aligned}
 \tag{2.15}$$

$$cond_E = \begin{cases} 1 & var(E_{k-14} : E_k) < th_E \\ 0 & otherwise \end{cases} \quad (2.16)$$

2.4 Activity Recognition using Machine Learning

Mobile devices such as tablets and smartphones are everywhere around people, so navigating themselves regardless of inside and outside of building has been interesting research topic over the last few years. In addition, numerous poses caused by the placement of the device such as calling, user's pocket, and texting degrade accuracy as the device does not attached to the part of pedestrian's body [58].

In terms of HAR (Human Activity Recognition), machine learning techniques using MEMS inertial sensors for HCI (Human Computer Interaction) are commonly used, and numerous methods are evaluated for accurate classification using body-worn inertial sensors [86].

2.4.1 Challenges in HAR

There are several challenges in HCI systems mentioned in [2]. Those are, in short, intra-class variability, inter-class similarity, definition of physical activities, class imbalance, ground truth annotation, and data collection experiment design.

Intra-class variability is a matter of diversity within classes. This can be caused by several people performing the same action, and there are also differences in whether the same action occurs in the morning or night. Next, inter-class similarity is another class, but sensor characteristics are similar. And there is a general definition issues of motion because of difficulties that occur in HAR.

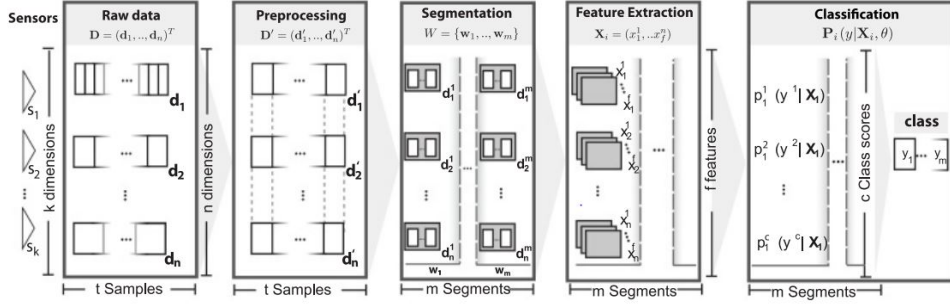


Figure 2.7: Typical ARC from wearable sensors [2]

Human activities are very complex and diverse, and depending on the situation, the activities can be carried out in different ways. Class imbalance means that there is an imbalance between classes because some actions do not occur frequently and some actions occur frequently. In addition, in case of behavior analysis, it is difficult to collect accurate ground truth data, so it is usually labeled by the person collecting the data. Finally, there is few common dataset, which makes it more difficult to evaluate performance in behavior analysis.

2.4.2 Activity recognition chain

ARC (Activity Recognition Chain) in Fig. 2.7 refers to the process of recognizing behavior using processes of signal processing, pattern recognition, and machine learning techniques.

First, after k sensor raw data comes in, the sensor sync is adjusted during the preprocessing process, or calibration, unit conversion, and resampling are performed. Next is the process of cutting data to make it easy to classify as a process of data segmentation.

Data segmentation

Data segmentation is the process of dividing the pre-processed sensor signal into smaller time segments with information about the behavior. Sliding window, event-based window, activity-based window are typical. Sliding window refers to the use of a fixed length window and has the advantage of being easy to operate in real time. Event-based window requires preprocessing in advance, and when applied to PDR, it is typical to know the heel strike and toe off phase situations and segment data based on this. An activity-based window is a window cut based on when the behavior changes, as used in wavelet analysis.

Feature selection and feature extraction

Both feature selection and extraction methods refers to the process of reducing data to feature points suitable for classifying behaviors. As the dimension of features increases, the classification performance increases until it reaches the optimal number. If exceeding the point, the required number of training samples is increasing as feature dimension gets bigger for classifier performance, which is called as a curse of dimensionality problem [87]. Therefore, features are usually calculated in both time and frequency domains in advance, then the feature reduction process is conducted to select the optimal features that affect the classification results.

The condition of a good feature vector is that the variation is not large when the same operation is performed several times, and even if it is performed by other people, the change should not be large. And among other motions, the feature vector must be clearly distinguishable. In addition, redundancy should be minimized between feature vectors to reduce computational complexity.

There are two possible ways to reduce feature vector dimensions. The first

one is called a feature selection method searching a subset of appropriate features from the original set. The other, a feature extraction, is a combination of original feature vectors.

Feature selection is a process of choosing a subset of relevant features from the original set [88]. Among numerous methods to scale down feature vectors, one of representative algorithms based on information theory called mRMR (minimum Redundancy Maximum Relevance) is introduced in the dissertation [89].

Feature extraction is a projection of a high-dimensional original feature space into a new low-dimensional feature space and usually consists of a linear or nonlinear combination of the original feature space. However, in the case of feature extraction, there is a disadvantage that the physical meaning of the original feature vectors is lost because new feature vectors are created by reducing the dimension. Therefore, it is said that feature selection is preferred to systems that are actually applied. The other feature extraction method is a combination of original features for relevant feature selection [86]. It is useful because it facilitates classification and visualization of high dimensional data, but the physical meaning of features is lost through the process. The common techniques used in feature extraction are PCA, LDA (Linear Discriminant Analysis), and so on. The PCA is one of the most popular and common feature extraction methods, and it is the linear technique transforming inter-correlated features into uncorrelated features.

Training and classification

In terms of training and classification, there are numerous classifiers in machine learning, and two basic classifiers, kNN (k-th Nearest Neighbor) and SVM (Support Vector Machine) as in Fig. 2.8 are adopted for comparison in this dis-

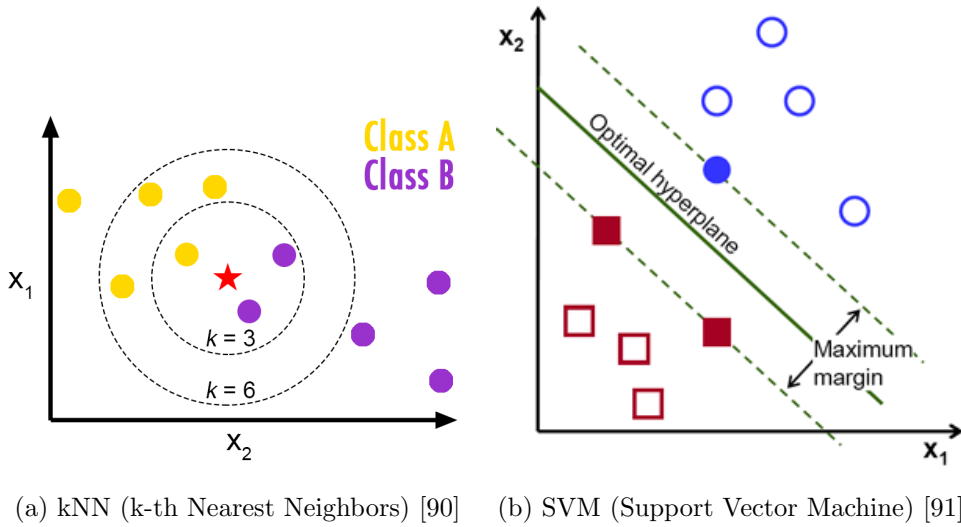


Figure 2.8: Classifiers

sertation. KNN is a principle of similarity between the training set and test data [92]. Since it only determines class through the distance, so it is non-parametric method, and one of the simplest of all machine learning algorithms. The inputs are k closest training examples in the feature space and the output is a class membership for classification.

The powerful SVM is a statistical learning theory-based classifier which minimizes an empirical risk and maximizes the margin between separating hyperplane and the data at the same time [93]. The SVM is a basically linear classifier but non-linear classification can be made using kernel methods, and the basic kernels are linear, polynomial, RBF (Radial Basis Function), and sigmoid. In addition it is able to classify multi-class with pair-wise classification with high accuracy on average, so it is widely used for most of supervised classification methods.

Chapter 3

Attitude Estimation in Smartphone

3.1 Adaptive Attitude Estimation in Smartphone

In this chapter, adaptive attitude estimation using ellipsoidal method is proposed for the pedestrian navigation system. To estimate the heading of a device is an essential part of both PA and IA, which will be combined in the chapter 4. The attitude estimation using the angular rate from the gyroscope, the specific force from the accelerometer, magnetic field from magnetometer are called AHRS. Using the characteristics that gyro measure has a low-frequency component, and accelerometer and magnetometer have high-frequency one, AHRS combines those sensors with filtering methods. However, its performance degrades, especially when the sensor is moving fast or exposed to a magnetic disturbance in hand. In the following sections, the proposed methods will be described in detail.

3.1.1 Indirect Kalman filter-based attitude estimation

Quaternion representation of attitude and sensor model

The body frame is an orthogonal axis set aligned with the vehicle's attitude, and the sensor measurements are collected on this frame. The navigation frame is a coordinate frame fixed to the Earth's surface, having its origin at the

location of the navigation system, and NED (North-East-Down) is used as a navigation frame in this paper. The attitude is defined as the relationship between the two frames, and the attitude between body and navigation frame is represented using quaternion:

$$\mathbf{r}^n = \mathbf{C}_b^n \mathbf{r}^b \quad (3.1)$$

where the \mathbf{r}^n , \mathbf{r}^b are navigation and body vector, respectively, and the rotation matrix \mathbf{C}_b^n is DCM (Direction Cosine Matrix).

The DCM propagates with time by the following equation:

$$\begin{aligned} \dot{\mathbf{C}}_b^n &= \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^b \\ \text{where } \boldsymbol{\Omega}_{nb}^b &= \begin{bmatrix} 0 & -\omega_{nb,z}^b & \omega_{nb,y}^b \\ \omega_{nb,z}^b & 0 & -\omega_{nb,x}^b \\ -\omega_{nb,y}^b & \omega_{nb,x}^b & 0 \end{bmatrix} \end{aligned} \quad (3.2)$$

where the ω_{nb}^n is the turn rate of body frame with respect to the navigation frame, expressed on the body frame.

The three sensors, used for the attitude estimation, are modeled as follows:

$$\begin{cases} \mathbf{y}_g^b = \boldsymbol{\omega}^b + \boldsymbol{\varepsilon}^b + \mathbf{w}_g \\ \mathbf{y}_f^b = \mathbf{f}^b + \mathbf{C}_n^b \mathbf{g}^n + \nabla^b + \mathbf{w}_f \\ \mathbf{y}_m^b = \mathbf{m}^b + \mathbf{C}_n^b \mathbf{m}^n + \mathbf{w}_m \end{cases} \quad (3.3)$$

where \mathbf{y} , $\boldsymbol{\varepsilon}$, ∇ , \mathbf{w} , \mathbf{g} , \mathbf{f} , \mathbf{m} are sensor output, gyro bias, accelerometer bias, noise, gravity vector, acceleration, magnetic field, respectively, and the superscript and subscript represent frame and sensor, respectively. The \mathbf{g}^n and \mathbf{m}^n are given reference gravity and magnetic field vector at the current position.

Indirect Kalman filter

Kalman filter is known to minimize the squared error of the state for linear system and measurement models with model uncertainties described as zero-mean Gaussian white noises [94]. In order to fuse gyro, accelerometer and magnetometer outputs, the system is modeled as gyro integration and the measurements of accelerometer and magnetometer output under no acceleration and magnetic disturbance, respectively. Gyro bias, one of the error sources, is estimated together with the attitude, it is reasonable to model process noise as zero-mean Gaussian white noise. Indirect Kalman filter is used to estimate attitude and gyro bias error in this dissertation and the error states at the k^{th} epoch is the following:

$$\begin{aligned}\mathbf{x}_k &= \begin{bmatrix} \boldsymbol{\varphi} & \hat{\boldsymbol{\varepsilon}}^b \end{bmatrix}^T \\ &= \begin{bmatrix} \varphi_N & \varphi_E & \varphi_D & \varepsilon_x & \varepsilon_y & \varepsilon_z \end{bmatrix}^T\end{aligned}\quad (3.4)$$

where $\varphi_N, \varphi_E, \varphi_D$ correspond to the attitude errors represented as psi-angle in INS. These terms are approximately equal to the roll, pitch, yaw error for small-angle misalignments [95]. The relationship between estimated attitude $\tilde{\mathbf{C}}_b^n$ and true DCM \mathbf{C}_b^n is defined as (3.5):

$$\tilde{\mathbf{C}}_b^n = [\mathbf{I}_{3 \times 3} - [\boldsymbol{\varphi} \times]] \mathbf{C}_b^n \quad (3.5)$$

where the representation $[\cdot \times]$ is a skew-symmetric matrix given as (3.6):

$$[\boldsymbol{\varphi} \times] = \begin{bmatrix} 0 & -\varphi_z & \varphi_y \\ \varphi_z & 0 & -\varphi_x \\ -\varphi_y & \varphi_x & 0 \end{bmatrix} \quad (3.6)$$

The gyro bias process model is a random walk, and the corresponding system matrix in the filter is in (3.7) and Δt is the time difference between two epochs,

which is reciprocal of the sampling rate.

$$\Phi_k = \begin{bmatrix} \mathbf{I}_{3 \times 3} & -\tilde{\mathbf{C}}_{b,k}^n \Delta t \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (3.7)$$

The measurement is the residual; the difference between true value represented in body frame and sensor output, and its measurement matrix is given as (3.8):

$$\begin{aligned} \mathbf{z}_k &= \begin{bmatrix} \mathbf{y}_{f,k}^b - \tilde{\mathbf{C}}_{n,k}^b \mathbf{g}^n & \mathbf{y}_{m,k}^b - \tilde{\mathbf{C}}_{n,k}^b \mathbf{m}^n \end{bmatrix}^T \\ \mathbf{H}_k &= \begin{bmatrix} \tilde{\mathbf{C}}_{n,k}^b [\mathbf{g}^n \times] & \mathbf{0}_{3 \times 3} \\ \tilde{\mathbf{C}}_{n,k}^b [\mathbf{m}^n \times] & \mathbf{0}_{3 \times 3} \end{bmatrix} \end{aligned} \quad (3.8)$$

For the more detailed information about the indirect Kalman filter, see [96]. Besides, the two-stage attitude update filter in indirect Kalman filter is implemented to avoid attitude error from magnetometer [97].

3.1.2 Conventional attitude estimation algorithms

In order to estimate the device attitude, various adaptation approaches are studied to use acceleration and magnetometer measurements. In the earlier stage, adaptation decision is made by comparing the accelerometer norm, magnetometer norm, and inclination angle with its reference values [98]. In [99], the fuzzy rule is implemented for detecting acceleration and adjusting measurement covariance in ARS (Attitude Reference System). Modeling acceleration or magnetometer disturbance is also proposed by several papers [100–102]. Suh proposes the adaptation rule to calculate correct innovation covariance with non-negative measurement covariance angle using Eigenvalue and Eigenvector [97]. In terms of finding the adaptation coefficient of measurement covariance, the method to minimize the Frobenius norm of innovation covariance is also proposed [103]. Li proposes the adaptation method to divide the acceleration condi-

tion with three stages: no, low, and high acceleration [104]. To adjust innovation covariance with the direction of residual or just all directions are proposed by Ghobadi, but the author ultimately applies the latter because of initial bias and attitude error [105]. Kang suggests modeling the dynamic mode probability by HMC (Hidden Markov Chain), and the modes are divided into dynamic and stationary situations [96]. Similarly, adaptive rule is proposed to tune the measurement noise covariance by detecting interference using the HMM (Hidden Markov Model) [106]. The adaptive algorithm using UKF (Unscented Kalman Filter) has been recently proposed in numerous papers [107–114]. Chiella proposes the adaptive UKF to add robustness in measurement covariance change by considering non-Euclidean algebra in unit quaternion [107, 108], and the measurement covariance is adapted by the method of [115]. In [109], the double-step UKF is proposed by dividing accelerometer and magnetometer measurement update in two-stage in order to avoid the wrong attitude correction from magnetometer, and the diagonal covariance inflation in [105] is applied. In robust Kalman filtering, Huber methodology combining l_1 and l_2 norm shows robustness under deviation in Gaussian distribution and dynamic model and its applications are in [110–114]. If this acceleration or disturbance condition lasts for a long time, the error is accumulated by time. All the above methods inflate measurement covariance under the acceleration, which in turn means the gyroscope mainly updates attitude.

3.1.3 Adaptive attitude estimation using ellipsoidal methods

In this section, attitude estimation using the adaptive algorithm is described to handle acceleration and magnetic disturbance. The block diagram for the proposed algorithm is in Fig. 3.1, and the proposed algorithm in this paper is marked as orange. While a pedestrian is moving with the smartphone in hand,

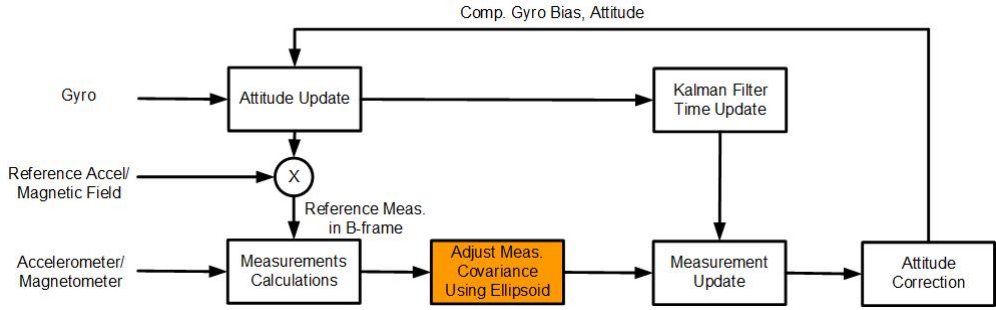


Figure 3.1: Proposed ellipsoidal method-based adaptive AHRs diagram

and the accelerometer in the sensor can be described as:

$$\mathbf{f}^b = \mathbf{f}_{ext}^b + \mathbf{C}_n^b \mathbf{g}^n + \mathbf{w}_f \quad (3.9)$$

When there is an additional acceleration, it is essential to deal with the acceleration residuals. Therefore, the adaptive algorithm using ellipsoidal method for the residual is proposed. The previous work [105] considers generalized covariance union-based covariance inflation using the residual inconsistency measured from Mahalanobis distance. As the author mentioned in the paper, it is sensitive to the wrong initial attitude and gyro bias estimation. Similarly, the proposed algorithm also adjusts the measurement covariance taking into account the direction of the residuals in a heuristic way using the MVCE (Minimum Volume Covering Ellipsoid) [116]. MVCE is to find minimum covering ellipsoid of given ellipsoids using a convex optimization technique. According to the [117], the difference between general covariance union and minimum enclosing ellipsoid problems is the interpretation in a statistical and geometric sense, respectively. Specific proofs and mathematical explanations for the computation and optimization are described in the literature [116–118].

Given two ellipsoids, one is the measurement covariance with no acceleration, and the other is the one with acceleration. Our goal is to find the third

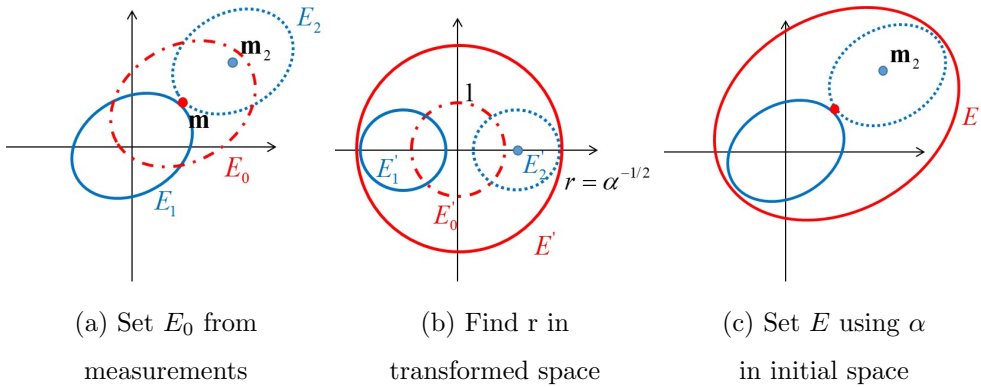


Figure 3.2: Adaptive measurement covariance process

ellipsoid, E , covering both ellipsoids. Following the ideas from the algorithms for ellipsoids [119], the adaptive measurement is proposed as follows. Assuming that the ellipsoid (E_1) with no acceleration is defined as (3.10) that the set of all \mathbf{p} on ellipsoid (E_1) centered at mean (\mathbf{m}_1) with covariance (\mathbf{R}_1).

$$E_1 \equiv \left\{ \mathbf{p} \mid (\mathbf{p} - \mathbf{m}_1)^T \mathbf{R}_1^{-1} (\mathbf{p} - \mathbf{m}_1) \leq 1 \right\} \quad (3.10)$$

The ellipsoid E_2 for acceleration condition is defined in the same way. If E is an ellipsoid that includes two ellipsoids, the corresponding mean and covariance can be calculated by means of E_0 , which is as follows:

$$\mathbf{m} = \mathbf{m}_0 = \frac{1}{2} (\mathbf{m}_1 + \mathbf{m}_2) \quad (3.11)$$

$$\begin{aligned} \mathbf{R}^{-1} &= \alpha \mathbf{R}_0^{-1} \\ &= \alpha \left(\mathbf{R}_1 + \mathbf{R}_2 + \frac{1}{4} [\mathbf{m}_1 - \mathbf{m}_2] [\mathbf{m}_1 - \mathbf{m}_2]^T \right)^{-1} \end{aligned} \quad (3.12)$$

where \mathbf{R}_0 is a covariance of the ellipsoid, the corresponding ellipsoid, E_0 , depicted in Fig. 3.2a. Our goal is to find a positive parameter α . In order to find

the parameter α , the \mathbf{R}_0^{-1} is firstly decomposed by Cholesky factor, \mathbf{L}_0 in (3.13), and E_0 is linearly transformed into a unit sphere using (3.14) in Fig. 3.2b. The ellipsoids, E_1 and E_2 and are accordingly transformed into E'_1 and E'_2 as the frame making the unit sphere E'_0 .

$$\mathbf{R}_0^{-1} = \mathbf{L}_0 \mathbf{L}_0^T \quad (3.13)$$

$$\mathbf{q} = \mathbf{m} + \mathbf{L}_0^{-T} \mathbf{p} \quad (3.14)$$

The covering ellipsoid E' should include the furthest point $\mathbf{q}_1, \mathbf{q}_2$ on the E'_1, E'_2 from the origin, and it is calculated by maximizing the distance r_1, r_2 . It is assumed that those are equal distance because the variances \mathbf{R}_1 and \mathbf{R}_2 are the same. Therefore, the distance maximizing quadratic equation is in (3.15).

$$r^2 = \mathbf{q}_1^T \mathbf{q}_1 = \left(\mathbf{m}_1 + \mathbf{L}_0^{-T} \mathbf{p}_1 \right)^T \left(\mathbf{m}_1 + \mathbf{L}_0^{-T} \mathbf{p}_1 \right) \quad (3.15)$$

The distance has a relationship with the parameter α as (3.16). The parameter is a ratio, so it is directly used in (3.12) and Fig. 3.2c for adjusting measurement covariance.

$$r^2 = \alpha^{-1} \quad (3.16)$$

Instead of calculating \mathbf{R}_0 in the above process, QL decomposition can be used by building a matrix as follows:

$$\mathbf{B}^T = \begin{bmatrix} \mathbf{L}_1^{-T} & \mathbf{L}_2^{-T} & \mathbf{d} \end{bmatrix} \text{ where } \mathbf{d} = \frac{1}{2} (\mathbf{m}_1 - \mathbf{m}_2) \quad (3.17)$$

$$\mathbf{R}_0 = \mathbf{L}_0^{-T} \mathbf{L}_0^{-1} = \mathbf{B}^T \mathbf{B} = \mathbf{L}^T \mathbf{L} \quad (3.18)$$

where \mathbf{L} is QL decomposition of \mathbf{B} .

Due to the relationship in (3.18) that $\mathbf{L} = \mathbf{L}_0^{-1}$, the \mathbf{L}_0 in (3.13)-(3.15) can be replaced by \mathbf{L} . The same process is adopted in the case of magnetic disturbance.

Using the ellipsoidal method-based measurement covariance adjustment, it is possible to deal with the external acceleration and magnetic disturbance in the direction of residuals. The advantage of considering the direction is that the undisturbed axis in residual in (3.8) can be used to estimate attitude when disturbance acts only on the single axis. Equation (3.19) describes the relationship between accelerometer and magnetometer measurements and attitude errors, respectively. When (3.19) is developed, the accelerometer is related to the φ_N, φ_E and the residual axis, and the $\varphi_N, \varphi_E, \varphi_D$ and the magnetic residual are related to each other. It suggests that, in the case of acceleration, φ_N, φ_E can be estimated even if a disturbance occurs in one axis. As long as the φ_N, φ_E is correctly estimated in the case of the magnetometer, φ_D can be estimated even if disturbances are present in two axes. It is because the φ_D is related to three magnetic measurement vector. Therefore, using the proposed method of considering the direction of the residual, the acceleration residual can be used according to the situation in which the disturbance comes in.

$$\begin{aligned}
\mathbf{z}_{f,k} &= \mathbf{y}_{f,k}^b - \tilde{\mathbf{C}}_{n,k}^b \mathbf{g}^n \\
&= \left(\mathbf{C}_{n,k}^b \mathbf{g}^n + \mathbf{w}_f \right) - \mathbf{C}_{n,k}^b (\mathbf{I} + \mathbf{E}) \mathbf{g}^n \\
&= -\mathbf{C}_{n,k}^b [\mathbf{g}^n \times] \boldsymbol{\varphi} + \mathbf{v}_f \\
\mathbf{z}_{m,k} &= \mathbf{y}_{m,k}^b - \tilde{\mathbf{C}}_{n,k}^b \mathbf{m}^n \\
&= \left(\mathbf{C}_{n,k}^b \mathbf{m}^n + \mathbf{w}_m \right) - \mathbf{C}_{n,k}^b (\mathbf{I} + \mathbf{E}) \mathbf{m}^n \\
&= -\mathbf{C}_{n,k}^b [\mathbf{m}^n \times] \boldsymbol{\varphi} + \mathbf{v}_m
\end{aligned} \tag{3.19}$$

The following Table 3.1 shows the comparison of adaptation rules to be

Table 3.1: Comparison of adaptation rules

	Adaptation Rules	Pros	Cons
Eigenvalue method Suh(2010)	$\mathbf{R}_{d,k} = \begin{cases} 0 & , \max_i (\lambda_{i,k} - \mu_{i,k}) < \varepsilon \\ \sum_{i=1}^3 \max [(\lambda_{i,k} - \mu_{i,k}), 0] \mathbf{u}_{i,k} \mathbf{u}_{i,k}^T & \end{cases}$	- Robust under high disturbance	- Adaptation condition is determined by threshold - Large error for yaw
Cost function method Yu(2012)	$\mathbf{R}_{d,k} = \hat{\mu} \mathbf{L}_1$ $\mu = \frac{\text{tr}([\hat{\mathbf{S}} - \mathbf{L}_0] \mathbf{L}_1^T)}{\text{tr}(\mathbf{L}_1 \mathbf{L}_1^T)}$ $\hat{\mu} = \max(1, \mu)$	- Adaptation parameter is robustly working well	- Heuristic decision of lower and upper bound
GCU method Ghobadi(2018)	$\mathbf{S}_k^- = \beta (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_{d,k}) + \mathbf{R}_k$ $\beta = \begin{cases} 2 & r > 1 \\ (1+r)^2 / 1+r^2 & r \leq 1 \end{cases}$	- Inconsistency in residual is considered	- Low accuracy when gyro bias, initial attitude error exist (wrong predicted error)
Huber method Qiu(2019)	$\mathbf{R}_{d,k} = \mu^{-1} \mathbf{R}_k$ $\rho(\tau) = \begin{cases} \tau^2/2 & , \tau \leq \gamma \\ \gamma \tau - \gamma^2/2 & , \tau > \gamma \end{cases}$ $\mu(\tau) \triangleq \frac{1}{\tau} \cdot \frac{\partial \rho(\tau)}{\partial \tau} = \min \left[1, \frac{\gamma}{ \tau } \right]$	- Robust under some modeling errors and noise uncertainty	- Heuristic decision of threshold - Error growth in magnetic disturbance
Proposed method	$\mathbf{R}_k + \mathbf{R}_{d,k} = \frac{1}{\alpha} \mathbf{R}_0$ $\mathbf{R}_0 = \mathbf{R}_1 + \mathbf{R}_2 + \frac{1}{4} [\mathbf{m}_1 - \mathbf{m}_2][\mathbf{m}_1 - \mathbf{m}_2]^T$	- Consider direction of measurement residuals	- Affected by wrong predicted error (residual direction error)

compared in the simulations and experiments. In the case of the eigenvalue technique, this technique can be applied in a situation with extreme disturbance, but there is a big disadvantage in that the threshold must be set and the yaw error using a geomagnetic field is large. In the case of the cost function-based technique, the adaptation parameter μ is operated strongly, but there is a disadvantage that the upper and lower bounds must be specified. The GCU-based technique has the advantage of considering the residual inconsistency, but it can be seen that the error is large when the gyro bias error and the initial attitude error are present. The Huber technique has the advantage of being robust against modeling errors or noise uncertainty, but it has the disadvantage that the error increases in the geomagnetic field and the threshold must be determined heuristically. The proposed method has the advantage of considering

Table 3.2: Sensor specifications

Xsens MTx [120]	Gyro	Accelerometer	Magnetometer
Bias Stability	5deg/s	2mg	0.5mGauss
Noise Density	0.1deg/s/ $\sqrt{\text{Hz}}$	0.2mg/ $\sqrt{\text{Hz}}$	0.5mGauss/ $\sqrt{\text{Hz}}$

the covariance of the measurement value, so that it can be used as much as possible in the situation where the measurement value can be used. However, if the assumption that the predicted error is small is broken, the residual vector value is wrong and the estimation error increases.

3.2 Experimental Results

In order to verify the effectiveness of the proposed algorithm, the reference value with the existing method through simulation and several experiments is compared. The low-cost MEMS IMU used for simulation and tests is Xsens MTX [120].

3.2.1 Simulation

The first simulation trajectory is shown in Fig. 3.3, where the sensor rotates at a constant angular velocity in the y-axis from 30 to 55 seconds, and there is no magnetic disturbance. It means that the acceleration components are only centrifugal force and gravity. In this case, the body frame x, y, and z axes defined as forward, right, and down, respectively. The sensor noise is generated according to the Xsens MTX specification, as in Table 3.2 [120]. In addition, the simulation conditions are listed in Table 3.3 by changing initial attitude, gyro bias, measurement covariance for accelerometer and magnetometer, and each condition is tested for 50 Monte Carlo runs.

Table 3.3: Simulation cases for trajectory #1 and #2

Case #	Gyro Bias[deg/s] [ε_x ε_y ε_z]	Initial Attitude[deg] [roll pitch yaw]	Measurement Covariance
1	[0.0 0.0 0.0]	[0.0 0.0 0.0]	$diag([\sigma_{acc} \ \sigma_{mag}])$
2	[1.0 -1.25 1.5]	[0.0 0.0 0.0]	$diag([\sigma_{acc} \ \sigma_{mag}])$
3	[2.0 -2.5 3.0]	[0.0 0.0 0.0]	$diag([\sigma_{acc} \ \sigma_{mag}])$
4	[4.0 -5.0 6.0]	[0.0 0.0 0.0]	$diag([\sigma_{acc} \ \sigma_{mag}])$
5	[2.0 -2.5 3.0]	[10.0 10.0 10.0]	$diag([\sigma_{acc} \ \sigma_{mag}])$
6	[2.0 -2.5 3.0]	[20.0 20.0 20.0]	$diag([\sigma_{acc} \ \sigma_{mag}])$
7	[2.0 -2.5 3.0]	[10.0 10.0 10.0]	$diag(0.1 \cdot [\sigma_{acc} \ \sigma_{mag}])$
8	[2.0 -2.5 3.0]	[10.0 10.0 10.0]	$diag(10 \cdot [\sigma_{acc} \ \sigma_{mag}])$

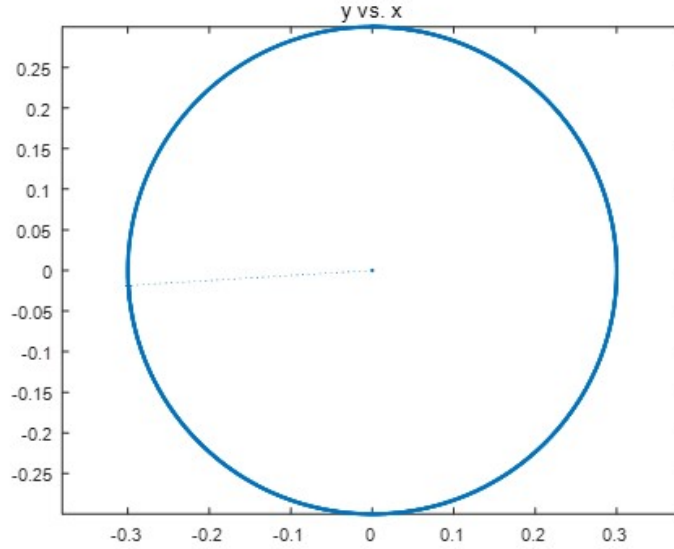
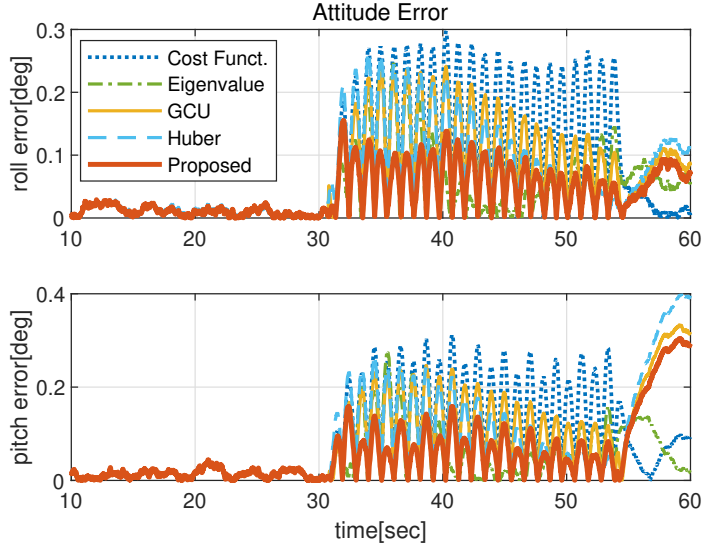


Figure 3.3: Simulation trajectory #1

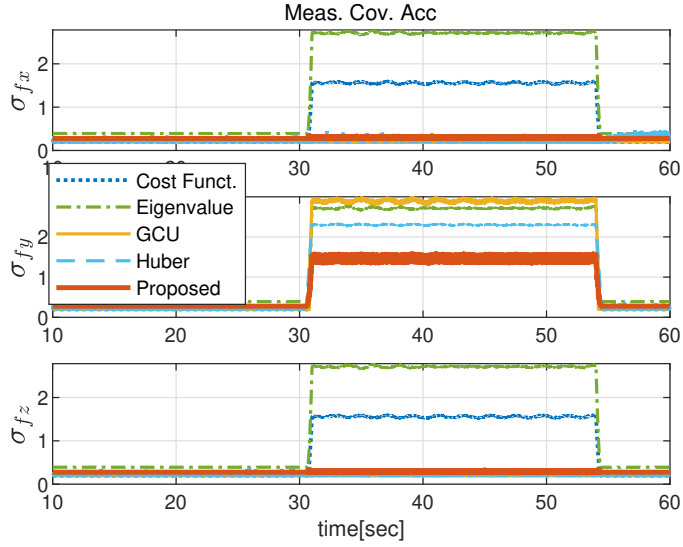
In order to avoid the effects of the magnetometer, the two-stage measurement update in [97] is implemented, and the magnetometer is always updated

Table 3.4: Attitude error in simulation #1

	Error[deg]	Roll[deg]	Pitch[deg]
Case1-4	Cost function method	11.2×10^{-2}	11.2×10^{-2}
	Eigenvalue method	6.53×10^{-2}	6.70×10^{-2}
	GCU method	7.98×10^{-2}	7.97×10^{-2}
	Huber method	5.94×10^{-2}	6.58×10^{-2}
	Proposed method	4.23×10^{-2}	4.37×10^{-2}
Case 5	Cost function method	14.4×10^{-2}	14.4×10^{-2}
	Eigenvalue method	7.81×10^{-2}	7.34×10^{-2}
	GCU method	11.4×10^{-2}	11.4×10^{-2}
	Huber method	7.61×10^{-2}	8.00×10^{-2}
	Proposed method	9.71×10^{-2}	9.59×10^{-2}
Case 6	Cost function method	1.73×10^{-1}	1.73×10^{-1}
	Eigenvalue method	1.88×10^{-1}	1.65×10^{-1}
	GCU method	1.54×10^{-1}	1.53×10^{-1}
	Huber method	28.8×10^{-1}	25.1×10^{-1}
	Proposed method	1.63×10^{-1}	1.58×10^{-1}
Case 7	Cost function method	2.99×10^{-1}	3.00×10^{-1}
	Eigenvalue method	3.04×10^{-1}	3.04×10^{-1}
	GCU method	34.4×10^{-1}	29.9×10^{-1}
	Huber method	1.09×10^{-1}	1.25×10^{-1}
	Proposed method	2.90×10^{-1}	2.90×10^{-1}
Case 8	Cost function method	4.81×10^{-1}	4.15×10^{-1}
	Eigenvalue method	11.9×10^{-1}	11.7×10^{-1}
	GCU method	4.81×10^{-1}	4.15×10^{-1}
	Huber method	8.05×10^{-1}	7.68×10^{-1}
	Proposed method	7.71×10^{-1}	7.22×10^{-1}



(a) Attitude Error



(b) Adapted Acceleration Measurement Covariance

Figure 3.4: Simulation results for trajectory #1

for all comparing adaptive filters. Also, there are some thresholds to be determined for the conventional adaptive algorithms [97,103,105,113]; each algorithm is set to show the best accuracy for the trajectory. Please note that the Huber-based UKF method is modified for the EKF structure for equal comparison. The φ_N, φ_E and measurement covariance for the accelerometer are in Fig. 3.4. The attitudes are compared in Fig. 3.4a and Table 3.4. In the case of gyro bias from cases 1 to 4, all methods estimate the gyro bias well so the results for those cases are compressed in Table 3.4. Considering the direction of the measurement, including the proposed method and [105,113] shows better results overall than the others. It is also depicted in accelerometer measurement covariance in Fig. 3.4b that only the y-axis component is inflated in the case of the GCU method, the Huber-based method, and the proposed algorithm. According to the simulation condition, as the initial attitude errors increase, the estimation

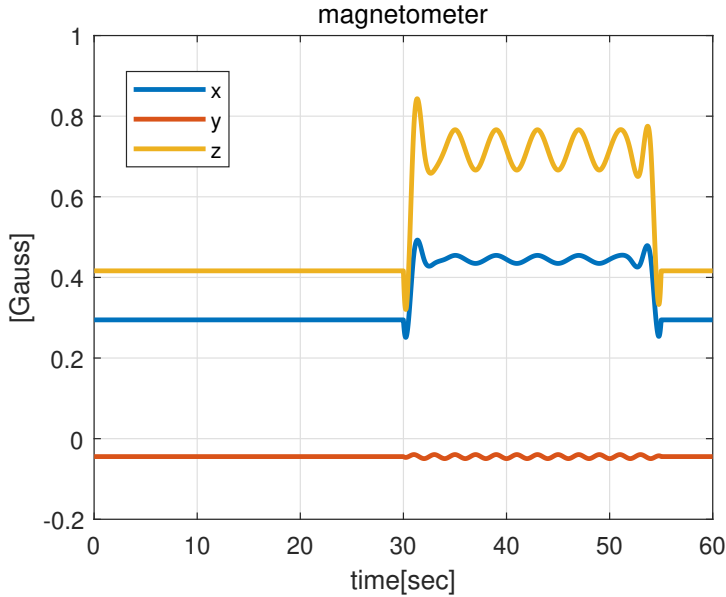


Figure 3.5: Simulation trajectory #2

Table 3.5: Attitude error in simulation #2

	Error[deg]	Yaw [deg]
Case1-6	Cost function method	0.695×10^{-1}
	Eigenvalue method	1.12×10^{-1}
	GCU method	0.545×10^{-1}
	Huber method	1.77×10^{-1}
	Proposed method	0.506×10^{-1}
Case 7	Cost function method	0.720×10^{-1}
	Eigenvalue method	1.18×10^{-1}
	GCU method	16.0
	Huber method	0.164
	Proposed method	12.3
Case 8	Cost function method	0.113
	Eigenvalue method	0.058
	GCU method	0.072
	Huber method	1.44
	Proposed method	0.106

accuracy of all methods is degraded. In addition, the wrong measurement noise covariance, for case 7 and 8, also causes errors in the attitude estimation especially for the methods considering the measurement direction. As the author mentioned in [105], it fails to estimate gyro bias and leads to significant attitude error during the Monte Carlo simulation especially for smaller covariance as case 7.

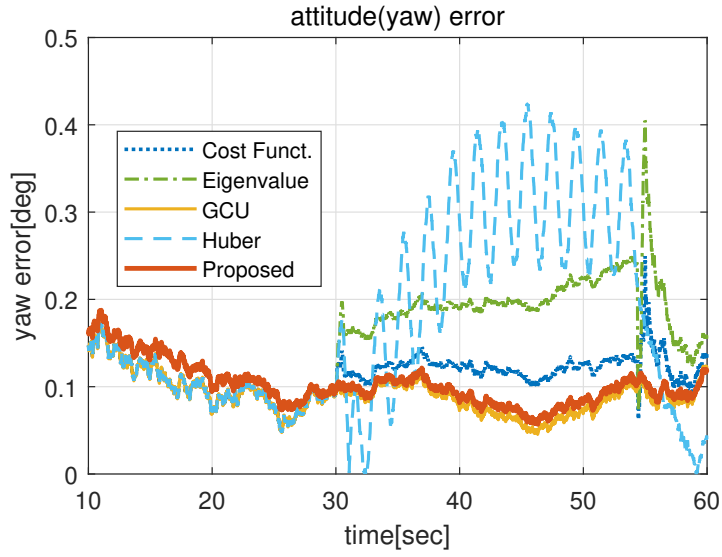
In the case of magnetic disturbance, the second simulation for the stationary condition is generated as Fig. 3.5 that x and y axes are mostly disturbed. It

is because the tilt error could affect the yaw error, and so the accelerometer is always updated for all algorithms without adaptation rules. In addition, initial attitude errors for roll and pitch are set to zero, and the measurement covariance of the accelerometer is also adjusted to the true value. The generated magnetic disturbance has both sinusoidal and constant components, and the degree of it is generated based on experimental data walked indoors.

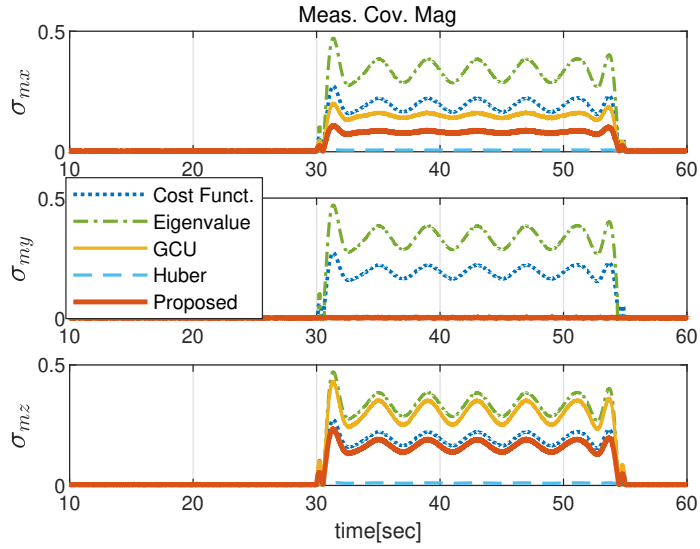
The attitude error and the measurement covariance for magnetometer are in Fig. 3.6. The covariance for the GCU method, the Huber-based method, and the proposed algorithm are different for all axes because of the disturbance in each axis, and considering the direction of measurement is also useful in the magnetic disturbance. The yaw estimation results are in Table 3.5 with the condition of Table 3.3. In the case of gyro bias and initial attitude error changes from case 1 to 6, all methods are not affected by the gyro bias and initial attitude change due to the accelerometer update, so the result for those cases are compressed in Table 3.5.

Case 7 of the simulation shows the limitations of the algorithm considering residual vectors. If the sigma value of the magnetometer is set smaller than the true value by one-tenth, an error occurs because the ellipsoid created from the residual is incorrectly reflected as a smaller sigma. In particular, in case 8, when the sigma is set to 10 times as large as the true value, this measurement is less reliable, so the error increases slightly. However, when set to $1/10$, it is more reliable than it should be, so the attitude error is caused. In turn, all the subsequent residuals have a bigger impact on errors because they are miscalculated. On the other hand, in the case of algorithms that do not consider the vector, even if the residual vector is wrong, the error is not large because the measurement update is not affected.

Across the simulations, the proposed algorithm in yaw estimation has the



(a) Attitude Error



(b) Adapted Magnetometer Measurement Covariance

Figure 3.6: Simulation results for trajectory #2

best performance as long as the measurement covariance is correctly set.

3.2.2 Rate table experiment

The first experiment is performed with an arc trajectory on the rate table as in Fig. 3.7. The sensor is attached to the top of the plate, having a tilt angle of 0 degrees and rotating with a speed of 1.745 rad/s for 5 minutes [121]. The experiment is carried out to confirm whether the proposed adaptive algorithm shows better performance under single-axis acceleration. The initial alignment to set gyro bias and initial attitude is processed for the first 10 seconds. Looking

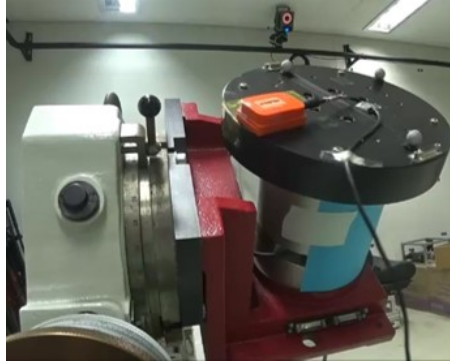
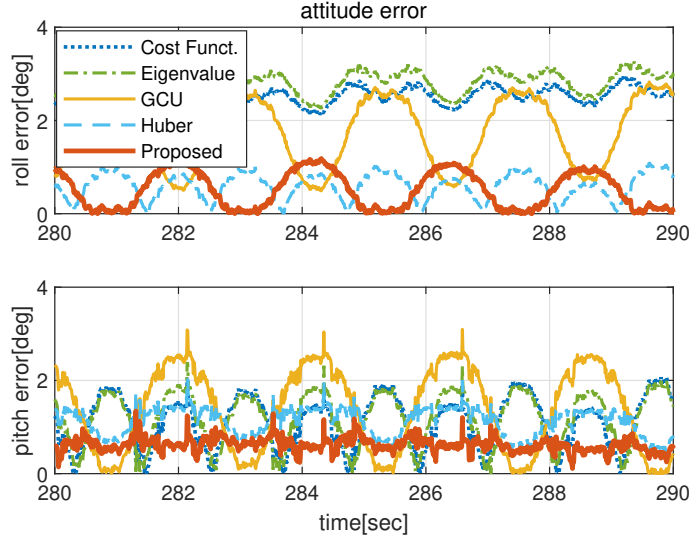


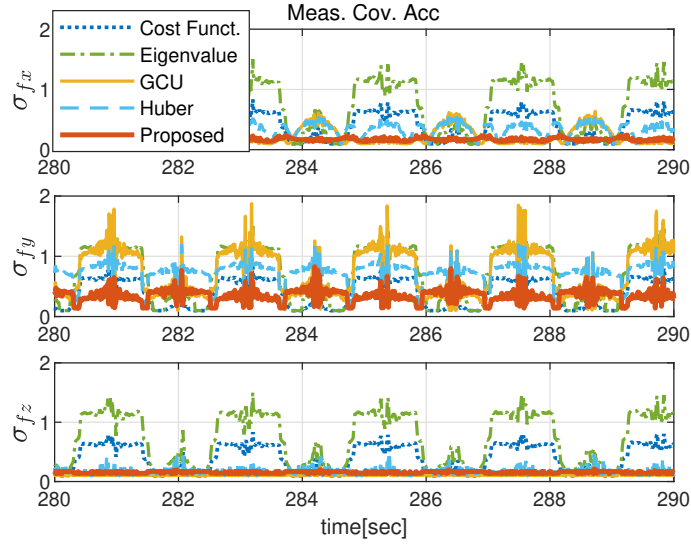
Figure 3.7: The sensor on the rate table

Table 3.6: Attitude error during movement in rate table

Error[deg]	Roll	Pitch
Cost function method	2.00	0.995
Eigenvalue method	2.00	0.959
GCU method	1.47	1.14
Huber method	0.446	0.700
Proposed method	0.460	0.348



(a) Attitude Error



(b) Adapted Acceleration Measurement Covariance

Figure 3.8: Rate table experiment results

at the estimated attitude in Fig. 3.8a and Table 3.6, the attitude with the proposed algorithm has the best accuracy compared to the other algorithms.

Results are shown for roll and pitch only. It is because the magnetometer is unavailable in this case due to the magnetic disturbance from the motor and iron base plate. The corresponding accelerometer measurement covariance is plotted in Fig. 3.8b, and the proposed algorithm and the conventional adaptive algorithm [105] and Huber-based method have different values across three axes following the residual values.

3.2.3 Handheld rotation experiment

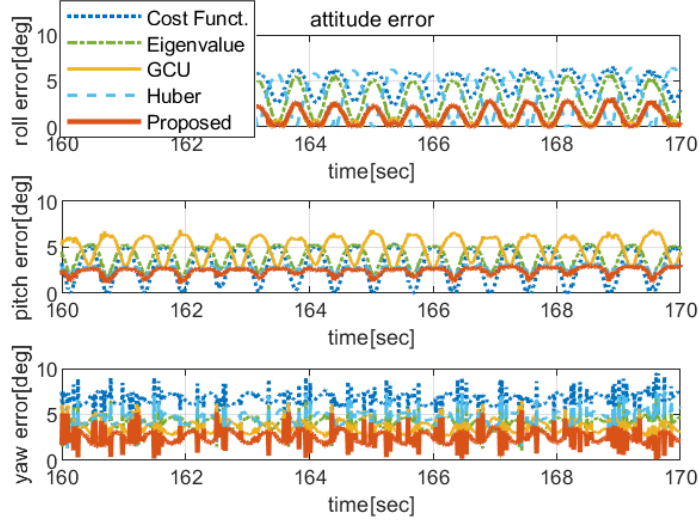
We test the algorithm for the handheld movement. The tester is moving the device in hand in the Fig. 3.9 with a yawing in arc trajectory for two minutes. By this motion, acceleration mainly comes in the x and y-axis. Similarly, the initial alignment process is run the same as the previous experiment. The reference attitude is acquired from the Vicon motion capture system [122]. The Fig. 3.10a and Table 3.7 show the attitude errors of the experiment, and Fig. 3.10b and 3.10c are the corresponding accelerometer and magnetometer covariance, respectively.

In case of tilt angle, the acceleration is mainly occurred in y axis by the

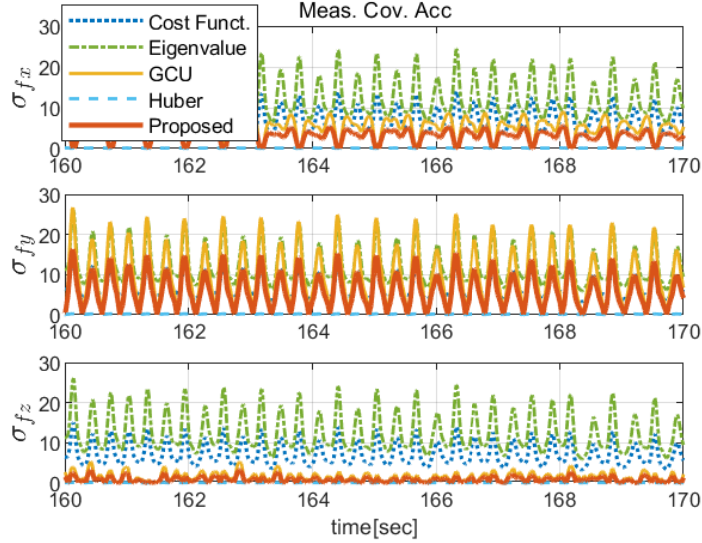


Figure 3.9: The sensor with Vicon markers

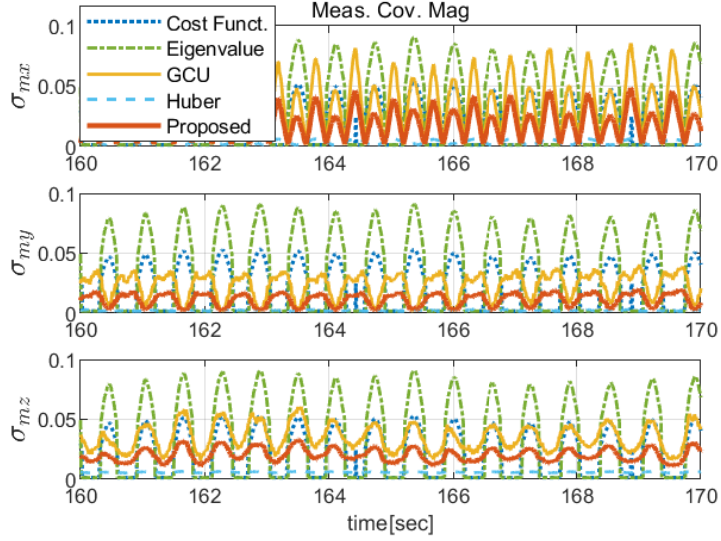
movement, so the acceleration covariance is set differently according to the proposed algorithm. The GCU and the proposed algorithm show better performance in than the others. It is because the tilt angle error affects the cal-



(a) Attitude Error



(b) Adapted Acceleration Measurement Covariance



(c) Adapted Magnetometer Measurement Covariance

Figure 3.10: Handheld experiment results

Table 3.7: Attitude error during hand movement

Error[deg]	Roll	Pitch	Yaw
Cost function method	4.14	2.32	5.89
Eigenvalue method	2.88	2.67	3.47
GCU method	1.23	1.65	2.38
Huber method	2.17	3.05	2.33
Proposed method	1.34	1.45	1.57

culuation of yaw angle, and those two methods has lower tilt angle errors. In order to apply the proposed algorithm in the magnetometer, it is essential to have the correct tilt angle in advance, if not, it is better to apply the adaptive rules having the same covariance in all axes. There is a single dominant acceleration axis in this scenario, so it shows the the effectiveness of the methods

considering the residual vectors. Through the hand experiment, the proposed ellipsoidal method-based adaptation also gives a better performance under the hand-moving scenario compared to other adaptation rules.

3.2.4 Magnetic disturbance experiment

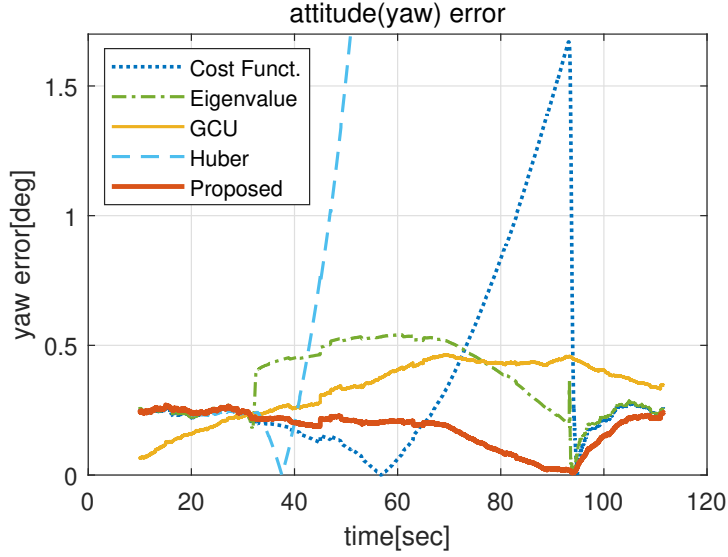
In order to see attitude estimation under only a magnetic disturbance situation, the test is performed with a stationary sensor. To be specific, the sensor is in place, and magnetic source, scissor, in this case, is placed next to it at the time from 35 to 90 seconds. The second scenario with changing magnetic disturbance is tested for the same time as the first one. In order to avoid the effect of tilt errors, the accelerometer is always updated during tests.

Yaw angle error, magnetic residuals, and the corresponding magnetometer measurement covariance for constant disturbance scenario are in Fig. 3.11 and Fig. 3.12. As seen from Fig. 3.11b, the x and y axes of residual has relatively large value compared to the other. Accordingly, the measurement covariance in the proposed algorithm has more considerable value on those axes in Fig. 3.11c. In the case of the second test with changing magnetic disturbance, the residual is plotted in Fig. 3.12b. In Fig. 3.12c, it is also noticeable that the covariance is

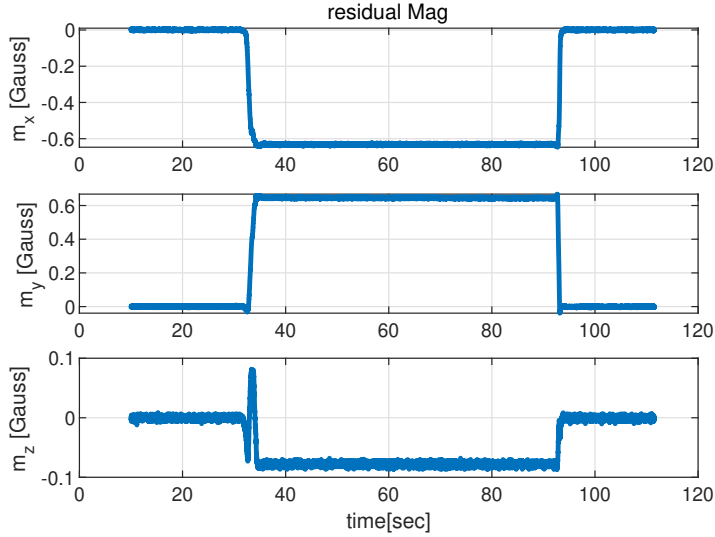
Table 3.8: Attitude error during magnetic disturbance

Yaw Error[deg]	Constant dist.	Changing dist.
Cost function method	0.500	1.04
Eigenvalue method	0.351	6.36
GCU method	0.327	0.214
Huber method	9.93	2.65
Proposed method	0.150	0.147

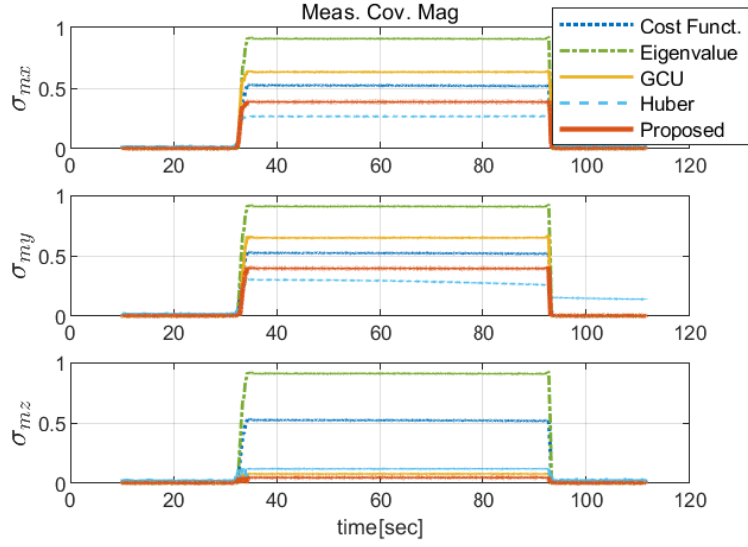
adjusted following the residual. The proposed ellipsoidal method-based adaptation is useful for yaw estimation during the two disturbance situations as in Table 3.8.



(a) Attitude Error

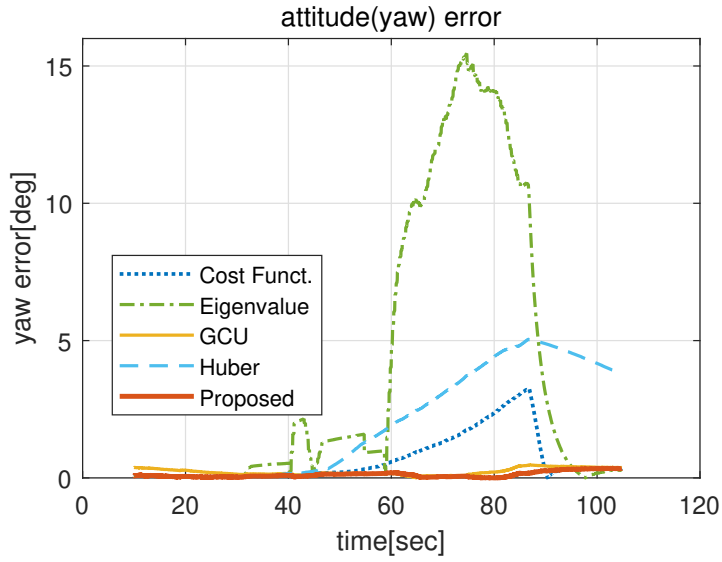


(b) Magnetometer Measurement Residuals

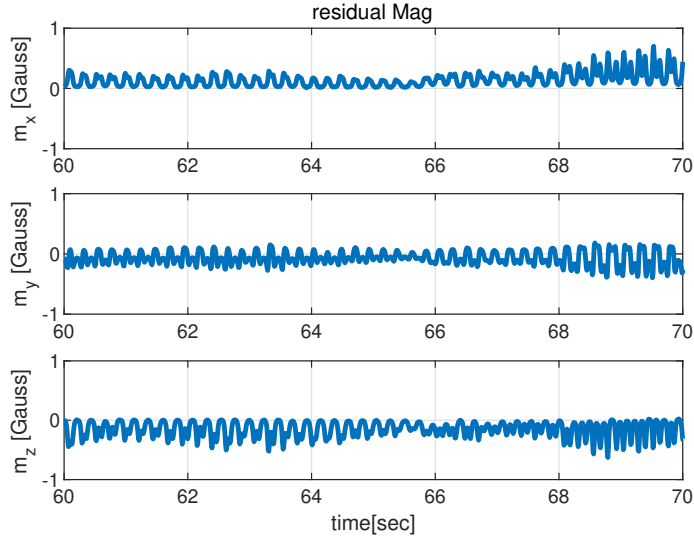


(c) Adapted Magnetometer Measurement Covariance

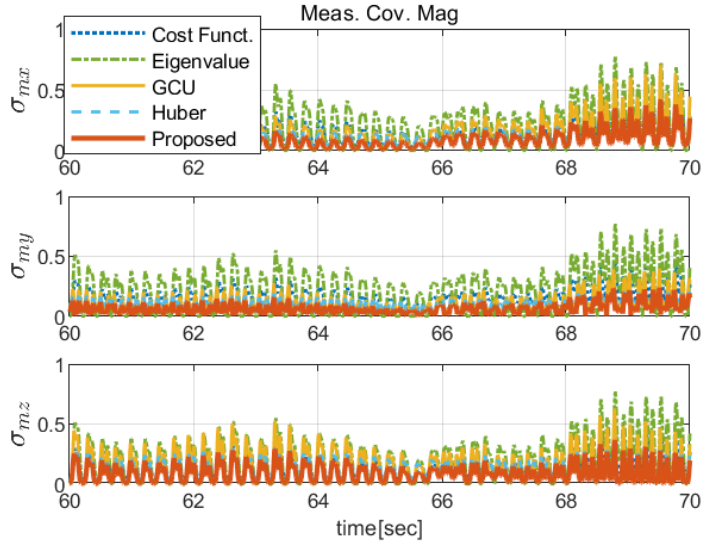
Figure 3.11: Magnetic disturbance experiment results for constant disturbance



(a) Attitude Error



(b) Magnetometer Measurement Residuals



(c) Adapted Magnetometer Measurement Covariance

Figure 3.12: Magnetic disturbance experiment results for changing disturbance

As mentioned before, using ellipsoidal adaptive method for heading estimation under magnetic disturbance, it is essential for tilt angle error should be low. Otherwise, the adaptive algorithm is not observable to estimate attitude errors using magnetometer data.

3.3 Summary

In this chapter, the adaptive attitude algorithm based on the ellipsoidal method with the measurement residuals is proposed. When the sensor keeps moving, especially in hand, attitude estimation using accelerometer measurement is hard to be updated, so conventional algorithms mostly depend on attitude updated by the gyroscope. In order to use the accelerometer or magnetometer measurement at most, the directions of measurement vectors are considered so that the undisturbed axis of the sensor could be used to update the attitude. The adaptive logic considering the measurement vector is based on the ellipsoidal method, and the covariance is adjusted to include the residual vector. Therefore, the proposed algorithm allows us to use more measurement when the measurement vector partially undisturbed. The proposed algorithm shows its effectiveness in simulation and experiments through the case of rate table, handheld, and magnetic disturbances.

Chapter 4

Pedestrian Dead Reckoning in Multiple Poses of a Smartphone

In this chapter, a new approach for IA and PA-based PDR using PCA and pose recognition is presented. Handheld smartphones are usually unrestricted and often have device heading changes that do not match the direction of walking. In order to remove the heading offset between them, some researchers have attempted to solve this problem [38–43]. In [38], the various PCA-based heading estimation methods are compared with broad experimental study in case of pocket. [39] uses a rotating axis for heading estimation dealing with swing, call, trouser pocket poses in the smartphone. Tian proposes adaptive offset compensation using the heading in straight holding mode under the swing, holding, and trouser pocket [40]. PCA-GA (Global Acceleration) method is proposed with the combination of TRIAD heading estimation in [41]. In addition, Deng proposed heading estimation using PCA [42, 43]. The above methods use PCA or fixed offset angle to solve the heading mismatch after the pose from the machine learning such as DT, SVM, or FSM. In the PA-based PDR system, the small heading difference leads to large position error, whereas the PCA-based methods vary heading values significantly depending on the data distribution between two steps. In addition, PDR components of step detection, step length, and heading estimation are different in terms of the classified poses, which also

means that the misclassification leads to position errors in PDR.

In order to solve the heading mismatch errors under multiple positions in PDR, the proposed algorithm takes the advantages of both IA and PA. The IA-based PDR calculates position, velocity, and attitude with acceleration and angular velocity by using integration, so it is able to find out walking direction in addition to the device heading. However, since PDR with IA generates errors quickly when using a low-cost inertial sensor, proper measurement is necessary to estimate the state of speed, position, etc.

Therefore, we combine the advantages of IA and PA in our smartphone PDR system. It is necessary to estimate IA states to suppress the increase in error, so the step length and heading are used as measurements from the PA when the directions match. If the device direction does not coincide with the walking direction, the walking direction is calculated using the PCA of the horizontal acceleration in navigation frame, and then the IA position is updated. In order to accurately understand the mode using two walking direction measurements, pose through machine learning is additionally classified. In addition, the adaptive attitude estimation algorithm based on ellipsoid technique mentioned in chapter 3 is used to estimate the attitude of the device in this system. In the following sections, the proposed methods will be described in detail.

4.1 System Overview

In this section, the IA and PA-fused PDR system is proposed for multiple smartphone poses, and the overall algorithm is in Fig. 4.1. The built-in inertial sensors and magnetometer from the smartphone are the input of the system, and the measurement update components are ZUPT, adaptive attitude estimation, step length from PA, and walking direction from PCA. Pose recognition by

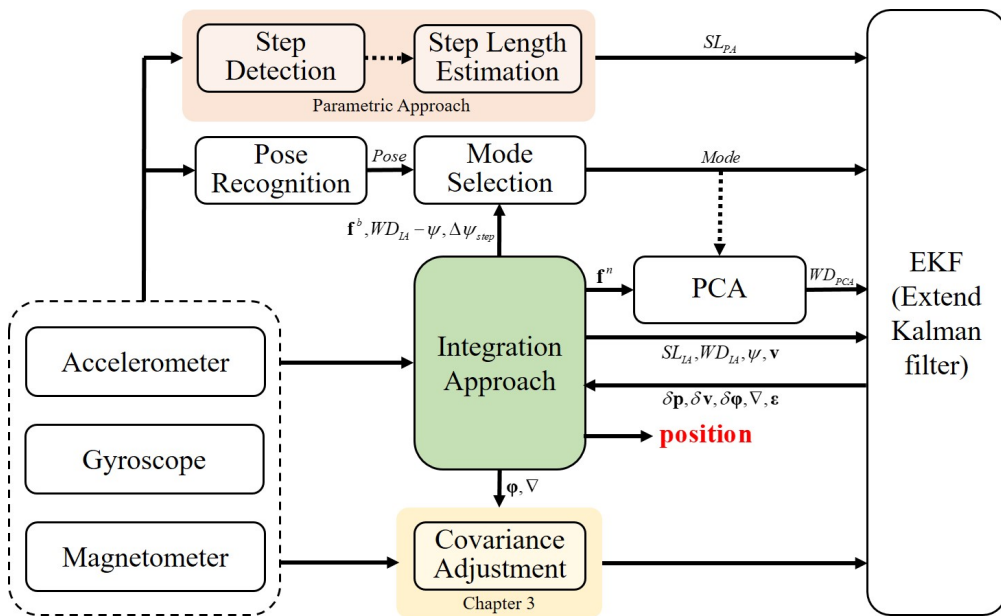


Figure 4.1: Overall IA-PA fusion PDR algorithm

machine learning is added to effectively find the measurement update mode while walking.

The assumptions used in constructing the algorithm in this paper are as follows. One of the biggest assumptions is that in a walking scenario, the starting pose is always text. If the pose is changed while walking with text pose, there is a heading mismatch problem in the direction of walking.

In the following subsections, the pose recognition through machine learning and the IA-PA fusion PDR system are explained in detail.

4.2 Machine Learning-based Pose Classification

Smartphone-based PDR system consists of step detection algorithm with step length estimation and heading estimation [123]. Each method could be changed following different poses, so the correct activity recognition is essential

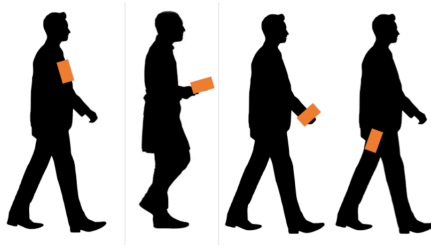


Figure 4.2: Target poses: chest, text, swing, trouser

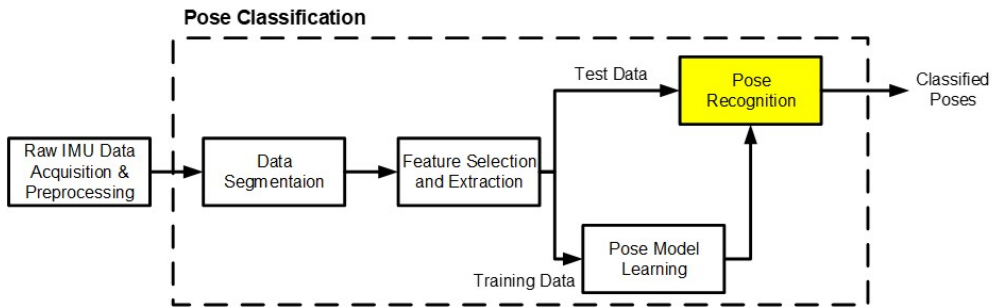


Figure 4.3: Overall pose recognition algorithm

in the PDR system. In this section, the poses in Fig. 4.2 are limited to four different poses possibly happens in a smartphone: text, swinging, shirt pocket, and trouser pocket. The machine learning-based behavior recognition process used in this section is as follows as in Fig. 4.3. After sensor data is collected and post-processed, features are extracted. The algorithms are constructed to select behavioral features that are appropriate for classification and to recognize behavior more efficiently. Then we use these features to train and test data.

4.2.1 Training dataset

Experiments are conducted to generate the training dataset and verify the accuracy and validity of the classification result. The experimental scenario is a 25m straight indoor corridor walking round-trip. Walking speed is 85, 90, 95,

Table 4.1: Sensor specifications for Xsens MTw

	ACC	GYR
Full scale	$\pm 160 \text{ m/s}^2$	$\pm 2000 \text{ deg/s}$
Non-linearity	0.5 % of FS	0.1 % of FS
Bias stability	0.1 mg	10 deg/hour
Noise	$200 \mu \text{ g}/\sqrt{\text{Hz}}$	$0.01 \text{ deg/s}/\sqrt{\text{Hz}}$

100, and 105 BPM (Beats Per Minute). Subjects are 23-40 years old, including 8 men and 4 women. The sensor used is a Xsens MTw with the sampling rate of 100Hz. The sensor specifications are given in Table 4.1 [124].

Figure 4.2 shows the target poses. Those are the shirt pocket, texting, swinging, and trouser pocket, which are common poses in daily smartphone usage. For shirt and thigh pocket situations, the sensor is secured to the pocket to avoid shaking. If done at once, the experiment will be done twice for the hand-held and swing context. We also collected data by flipping the sensor upside down to ensure that the algorithm works well despite changes in sensor posture.

4.2.2 Feature extraction and selection

Human activity recognition from inertial data is usually followed by feature extraction initially. Feature selection is then processed to increase computational efficiency.

Feature extraction

Signal characteristics such as time and frequency domain functions are widely used for feature calculation. Time domain features include mean, median, variance, skewness, kurtosis, range, etc [86]. Peak frequency, peak power, spectrum power and spectrum in different frequency bands, entropy are usually included in the frequency domain feature.

Regarding the training data generation in pose recognition, the 54 feature vectors commonly used are calculated in advance [86]. For the pose classification, mean, variance, minimum, maximum, range, rms, skewness, kurtosis, and correlation of tri-axial accelerometer and gyroscope are used in this dissertation. The frequency domain features are excluded for its irrelevance. In addition, the window size is 140, which is 1.4 second for 100Hz sampling rate, and the step size between windows are 5. This is because the selected window size is suitable to include a single stride.

Feature selection using mRMR

The feature selection is a process of choosing a subset of relevant features from the original set [88]. Among numerous methods to scale down feature vectors, one of representative algorithms based on information theory called mRMR is introduced and implemented in the paper [89].

Maximum dependency represents that the selected features jointly have the largest dependency on the target class. However, the insufficient samples and multivariate density estimation involving the inverse of the high-dimensional covariance matrix [89], which leads to computational burden to the system.

Instead, mRMR using relevance and redundancy is proposed in [89]. A measure of the uncertainty in a discrete random variable, entropy, is defined in (4.1), and the conditional entropy of given another random variable Y is in (4.2).

$$H(X) = - \sum_{x_i \in X} P(x_i) \log(P(x_i)) \quad (4.1)$$

where x_i is a specific value of random variable X and $P(x_i)$ is a probability of x_i over all possible values of X .

$$H(X|Y) = \sum_{y_j \in Y} P(y_j) \sum_{x_i \in X} P(x_i|y_j) \log(P(x_i|y_j)) \quad (4.2)$$

where $P(y_j)$ is a prior probability of y_j .

Using the above information, the information gain also called as mutual information is described as (4.3) meaning the dependency with entropy and conditional entropy.

$$I(X; Y) = H(X) - H(X|Y) = \sum_{x_i \in X} \sum_{y_j \in Y} P(x_i, y_j) \log\left(\frac{P(x_i, y_j)}{P(x_i)P(y_j)}\right) \quad (4.3)$$

With that, the redundancy is defined in (4.4) as the correlation between different features which should be minimized. On the other hand, the correlation between feature vector and class label, relevance in (4.5), should be maximized. By reflecting the redundancy and relevance requirements, the following (4.6) could be made for feature reduction.

$$Redundancy : \frac{1}{|S|} \sum_{X_j \in X} I(X_k; X_j) \quad (4.4)$$

$$Relevance : I(X_k; Y) \quad (4.5)$$

$$\begin{aligned} J_{mRMR}(X_k) &= \max(Relevance) - \min(Redundancy) \\ &= I(X_k; Y) - \frac{1}{|S|} \sum_{X_j \in X} I(X_k; X_j) \end{aligned} \quad (4.6)$$

where S is set of the current selected features that is initially empty, Y represents the class labels, $X_j \in S$ is a specific feature in the current S , and $J(\cdot)$ is a feature selection criterion (score) where, generally, the higher the value of $J(X_k)$, the more important the feature X_k is.

The features sorted in descending order are in Table 4.2. The important thing is that the post-ranking features do not mean that they are the least relevant or the most dependent. The mRMR selects features considering the

Table 4.2: Selected features from mRMR (descending order)

1	Max Accel. X	28	Var. Accel. X
2	RMS Gyro Z	29	Min Gyro X
3	Max Accel. Z	30	Kurt. Accel. X
4	Range Accel. Y	31	Range Gyro Y
5	Range Gyro Z	32	Min Accel. Y
6	Min Accel. Z	33	Skew. Gyro Z
7	RMS Accel. X	34	Mean Accel. Y
8	Range Accel. Z	35	Skew. Accel. Y
9	Kurt. Accel. Y	36	RMS Accel. Y
10	RMS Accel. Z	37	Min Gyro Y
11	Max Gyro Z	38	Max Gyro Y
12	Min Accel. X	39	Kurt. Gyro Z
13	Range Gyro X	40	Var. Gyro Y
14	Var. Gyro Z	41	Skew. Accel. Z
15	Mean Accel. X	42	Corr. Gyro (2)
16	Var. Accel. Y	43	Skew. Accel. X
17	Mean Accel. Z	44	Corr. Accel. (2)
18	Var. Accel. Z	45	Corr. Accel. (3)
19	Min Gyro Z	46	Corr. Gyro (1)
20	Range Accel. X	47	Mean Gyro Z
21	RMS Gyro X	48	Kurt. Gyro Y
22	Max Accel. Y	49	Skew. Gyro Y
23	Corr. Gyro (3)	50	Kurt. Gyro X
24	Kurt. Accel. Z	51	Mean Gyro X
25	Var. Gyro X	52	Corr. Accel. (1)
26	RMS Gyro Y	53	Mean Gyro. Y
27	Max Gyro X	54	Skew. Gyro X

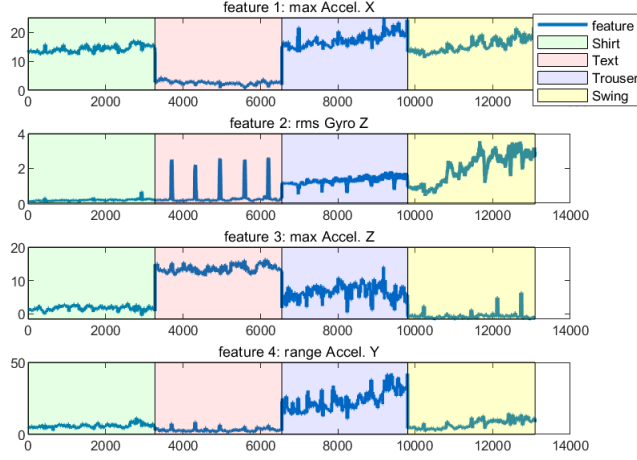


Figure 4.4: Selected features according to poses

relationship between all the features and provides the best combinations of the features in order.

The best combination of feature points according to the mRMR is shown in the Fig. 4.4. The feature is drawn from the combination derived through mRMR to the fourth from the beginning. Is it drawn for a straight round trip, the background color is green for shirt pocket, red for text messages, blue for the pants pocket, and yellow for the swing, respectively. Although there are some poses that can be distinguished by eye, these characteristics may vary from person to person, so I aim to classify the behavior more accurately using a classifier, which will be explained in detail in the next subsection.

4.2.3 Pose classification result using supervised learning in PDR

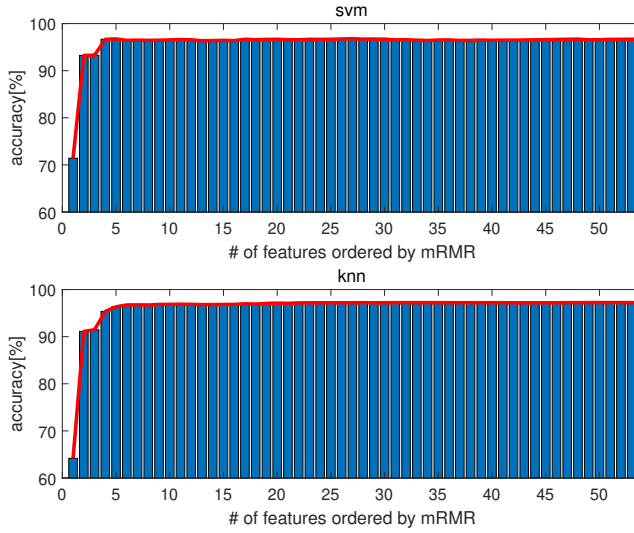
The entire dataset has dimensions of approximately 310,000 by 54 and is computationally inefficient, so mRMR is performed. As mentioned before in chapter 4.2.2, target poses are shirt pocket, text, trouser pocket, and swing.

The results of classification accuracy in the order of by mRMR in Table 4.2 are shown in Fig. 4.5a and the enlarged version is in Fig. 4.5b. As the number of features increases, the accuracy increases, but above a certain level, it can be seen that they are almost similar. Therefore, the best feature combination through mRMR is set to the first 5 for SVM and the first 6 for kNN.

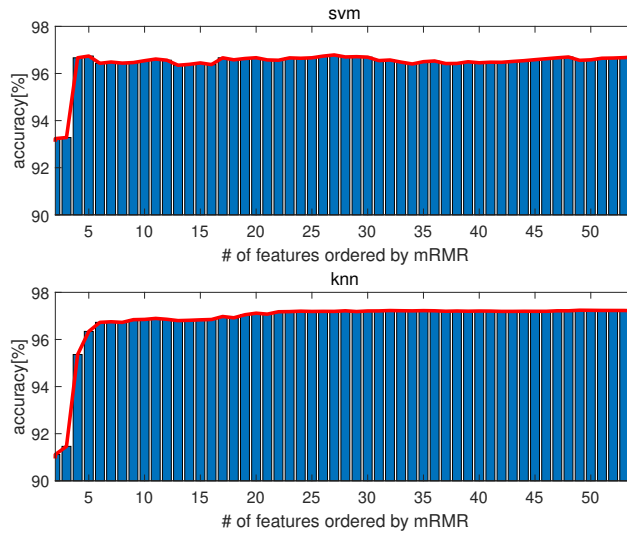
The kNN uses the principle of similarity or distance between the training set and test data, and the input is the k-closest training sets in the feature space, and the output is a target pose for the classification. The SVM is a statistical learning theory-based classifier, and the label is acquired by minimizing an empirical risk as a cost function and maximizing the margin between separating hyperplane and the data. For the parameters of the svm classifier, C-SVC type and radial basis kernel are used [89], and the k of the classifier of kNN is chosen as 4. The classification accuracy is in Table 4.3 for SVM and 4.4 for kNN, and the average accuracy is 96.74% and 96.73%, which is almost the same result for both classifiers.

Table 4.3: Accuracy results-SVM

True \ Classified	Shirt Pocket	Text	Trouser Pocket	Swing
Shirt Pocket	98.51	0	1.12	0.37
Text	0	95.59	4.41	0
Trouser Pocket	1.30	0	98.16	0.54
Swing	0.21	0	5.09	94.70



(a) Classification accuracy according to mRMR feature order



(b) Classification accuracy (enlarged)

Figure 4.5: Classification accuracy by the number of feature ordered by mRMR

Table 4.4: Accuracy results-kNN

Classified \ True	Shirt Pocket	Text	Trouser Pocket	Swing
Shirt Pocket	98.07	0	1.28	0.65
Text	4.17	95.83	0	0
Trouser Pocket	1.10	0	98.46	0.44
Swing	0.43	0	5.01	94.56

4.3 Fusion of the Integration and Parametric Approaches in PDR

In this section, the IA and PA-fused PDR system is specifically described for multiple poses in the smartphone. The IA-based PDR system is a simplified version of the INS for the low-cost inertial sensors, so the basic structure is the same as the INS. The system calculates the position by integrating the acceleration and angular rate at given every sample, and the process consists of five steps: bias compensation, orientation updates, gravity removal, integration, and correction [72].

In bias compensation step, the estimated accelerometer and gyro biases are subtracted in the raw sensor data as (4.7).

$$\begin{cases} \tilde{\omega}_k^b = \mathbf{y}_{g,k}^b - \hat{\varepsilon}_{k-1}^b \\ \tilde{\mathbf{f}}_k^b = \mathbf{y}_{f,k}^b - \hat{\nabla}_{k-1}^b \end{cases} \quad (4.7)$$

where $\tilde{\omega}_k^b$ and $\tilde{\mathbf{f}}_k^b$ are compensated gyro and accelerometer in body frame, respectively, and $\mathbf{y}_{g,k}^b$ and $\mathbf{y}_{f,k}^b$ are raw inertial sensor output in body frame, and $\hat{\varepsilon}_{k-1}^b$ and $\hat{\nabla}_{k-1}^b$ are estimated bias for gyro and accelerometer, respectively.

Next, attitude representing the relationship between body (b, defined as

Forward-Right-Down) and navigation (n, defined as North-East-Down) frame is updated using the angular rate in quaternion attitude representation as (4.8).

$$\mathbf{q}_k = (\mathbf{I} + \frac{1}{2}\mathbf{W}\Delta t)\mathbf{q}_k$$

$$\text{where } \mathbf{W} = \begin{bmatrix} 0 & -\tilde{\omega}_{x,k}^b & -\tilde{\omega}_{y,k}^b & -\tilde{\omega}_{z,k}^b \\ \tilde{\omega}_{x,k}^b & 0 & \tilde{\omega}_{z,k}^b & -\tilde{\omega}_{y,k}^b \\ \tilde{\omega}_{y,k}^b & -\tilde{\omega}_{z,k}^b & 0 & \tilde{\omega}_{x,k}^b \\ \tilde{\omega}_{z,k}^b & \tilde{\omega}_{y,k}^b & -\tilde{\omega}_{x,k}^b & 0 \end{bmatrix} \quad (4.8)$$

\mathbf{q}_k is a quaternion defined as $\mathbf{q}_k = \begin{bmatrix} q_{0,k} & \bar{\mathbf{q}}_k \end{bmatrix}^T = \begin{bmatrix} q_{0,k} & q_{1,k} & q_{2,k} & q_{3,k} \end{bmatrix}^T$.

In the third stage, gravitational components is removed from the compensated accelerometer as (4.9).

$$\bar{\mathbf{f}}^n = \mathbf{C}_{b,k}^n \tilde{\mathbf{f}}_k^b - \mathbf{g}^n \quad (4.9)$$

where $\mathbf{C}_{b,k}^n$ is a DCM transformed from the \mathbf{q}_k , and \mathbf{g}^n is gravity represented in n-frame.

In the fourth stage, velocity and position are calculated through the integration as (4.10).

$$\begin{cases} \mathbf{v}_k^n = \mathbf{v}_{k-1}^n + \bar{\mathbf{f}}_k^n \cdot \Delta t \\ \mathbf{p}_k^n = \mathbf{p}_{k-1}^n + \mathbf{v}_k^n \cdot \Delta t \end{cases} \quad (4.10)$$

where Δt , \mathbf{v}_k^n , and \mathbf{p}_k^n are sampling time, velocity, and position, respectively. The velocity model is simplified as above because the low-cost MEMS IMU does not measure the earth rotation rate and coriolis effect.

The fifth stage is the correction from the EKF error states using the measurements. The EKF is implemented to integrate the IA and PA methods in this dissertation, and the correction is made following the estimated errors, which is discussed in the following subsections. In brief, the position, velocity,

attitude, and biases for accelerometer and gyro are corrected following the EKF estimates.

4.3.1 System model

The built-in inertial sensors and magnetometer from the smartphone are the input of the system, and 15 error states of EKF are position of one stride, velocity, attitude, accelerometer bias, and gyro bias as in (4.11).

$$\delta \mathbf{x} = \begin{bmatrix} \delta \mathbf{p}_{step}^n & \delta \mathbf{v}^n & \delta \boldsymbol{\varphi} & \hat{\nabla}^b & \hat{\boldsymbol{\varepsilon}}^b \end{bmatrix}^T \quad (4.11)$$

where the $\boldsymbol{\varphi}$ is attitude represented in Euler angle. The corresponding system matrix is in (4.12) following the relationship among states.

$$\Phi = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} \Delta t & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \begin{bmatrix} \tilde{\mathbf{f}}^n \times \end{bmatrix} \Delta t & \mathbf{C}_b^n \Delta t & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathbf{C}_b^n \Delta t \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (4.12)$$

where \mathbf{I} is the identity matrix, and the numbers in subscript show its dimension. The zero matrix fit to dimensions are represented as $\mathbf{0}$ in the equation. The measurement updating processes are largely ZUPT, AHRS, and two-dimensional position from PA in the proposed algorithm. The first two processes are done in every sample, the other is every two steps.

4.3.2 Measurement model

ZUPT

In case of ZUPT in the smartphone, the zero velocity phase rarely occurs, but it is performed in the initial alignment and stationary condition. The phase

is detected using the windowed accelerometer z-axis variance, and the corresponding measurement equation is in (4.13) and (4.14).

$$\mathbf{z}_{ZUPT} = \mathbf{v}^n \quad (4.13)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}^T \quad (4.14)$$

AHRS

Using accelerometer and magnetic measurements, attitude of the sensor is determined as follows.

$$\mathbf{z}_{AHRS} = \begin{bmatrix} \tilde{\mathbf{f}}^b - \tilde{\mathbf{C}}_n^b \mathbf{g}^n & \mathbf{y}_m^b - \tilde{\mathbf{C}}_n^b \mathbf{m}^n \end{bmatrix}^T \quad (4.15)$$

$$H = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \tilde{\mathbf{C}}_n^b [\tilde{\mathbf{g}}^n \times] & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \tilde{\mathbf{C}}_n^b [\tilde{\mathbf{m}}^n \times] & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (4.16)$$

The covariance is adjusted following the residuals using the ellipsoidal methods. The specific details are described in chapter 3.

PA measurements - matched case

With the ZUPT and AHRS updates in the smartphone, the IA position is rapidly diverging. The PA position, however, is bounded because the step length is estimated from the parameters. Therefore, when the device heading and walking direction correspond to Fig. 4.6a, the 2D position calculated from PA is used as measurements. In situations such as Fig. 4.6a where the direction of walking matches the direction of the device, the step length obtained from the PA marked in orange and the direction of device can correct the IA position due to the accumulated error indicated in green.

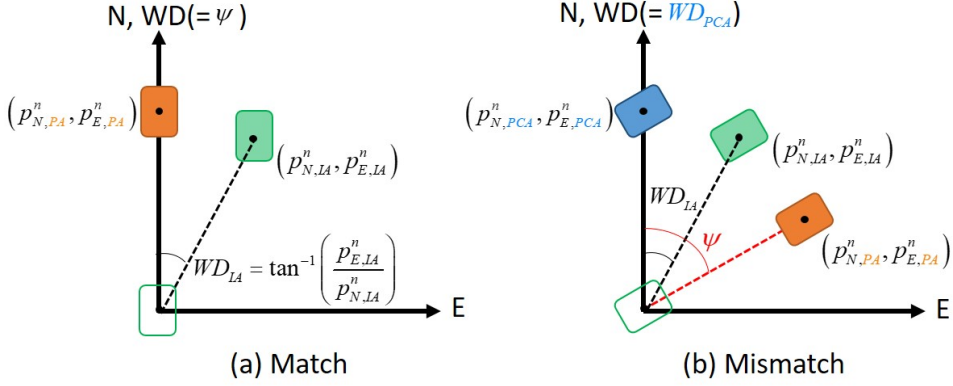


Figure 4.6: Measurement update

The position from the PA-based PDR system consists of three major components: step detection, step length estimation, and heading estimation between two consecutive steps. With the estimated component, the position is calculated by (4.17).

$$\begin{bmatrix} p_{N,k}^n \\ p_{E,k}^n \end{bmatrix} = \begin{bmatrix} p_{N,k-1}^n + SL \cdot \cos(\psi) \\ p_{E,k-1}^n + SL \cdot \sin(\psi) \end{bmatrix} \quad (4.17)$$

where k is k -th step, $p_{N,k-1}^n, p_{E,k-1}^n$ is the previous position in north and east, respectively, and ψ is device heading.

Firstly, accurate step detection is the basis of the PA-based PDR method for accurate position estimation. Even if the step length is estimated correctly, the inaccuracy of the step detection process can result in significant errors due to missing or adding a single step. The peak detection method detects the heel-strike which is the moment the foot touches the ground, allowing to find a step periodically. In this dissertation, considering the advantage of accurate heel strike detection, a peak detection method using acceleration is used. The acceleration data is low-pass filtered with a cutoff frequency set to 5 Hz and windowed to prevent noise effects.

As the name implies, the step length between steps in a PA is determined by its parameters. The IA, on the other hand, causes a significant error due to the double integration of the accelerometer output containing sensor error. Based on the linear relationship between the step length and the step frequency, we use a step length estimation method based on a linear combination similar to the previous work [48]. There are many features and functions for estimating the step length other than the walking frequency, but it is advantageous for the independence of the mounting position as long as the step is accurately detected. The following (4.18) is the step length formula applied.

$$SL = \alpha \cdot WF + \beta \quad (4.18)$$

where WF is walking frequency and α, β are pre-learned parameters according to the pre-calibration.

Lately, in the assumption under the PA-based PDR system, the device heading is the same as the walk direction. In this paper, the heading from the adaptive AHRS algorithm in chapter 3 is used in order to estimate accelerometer and magnetometer measurements efficiently.

In short, the measurement updates for PA positions are performed based on the heading match condition check. It compares the heading difference between the walking direction calculated from two-step positions in IA and the device heading during two steps, and the pose recognition results are additionally used for better decision. The reason for using a stride, two steps, is that the walking direction is oscillating following the walking characteristics of each user. The walking direction and step length for IA are calculated as (4.19) and (4.20).

$$WD_{IA} = \tan^{-1} \frac{\Delta p_E^n}{\Delta p_N^n} \quad (4.19)$$

$$SL_{IA} = \sqrt{(\Delta p_N^n)^2 + (\Delta p_E^n)^2} \quad (4.20)$$

where Δp^n is position between two steps.

For handheld conditions where the walking direction and device orientations match as shown in Fig. 4.6 (a), the heading offset is assumed as zero, so the step length from the PA and device heading are directly used to correct the IA states as in (4.21) and (4.22). In addition, assuming that the height for one stride are the same, the vertical position is also corrected.

$$\mathbf{z}_{match} = \begin{bmatrix} WD_{IA} - \psi & SL_{IA} - SL_{PA} & \Delta p_D^n \end{bmatrix}^T \quad (4.21)$$

$$H = \begin{bmatrix} -\frac{\Delta p_{E,IA}^n}{(\Delta p_{N,IA}^n)^2 + (\Delta p_{E,IA}^n)^2} & \frac{\Delta p_{N,IA}^n}{(\Delta p_{N,IA}^n)^2 + (\Delta p_{E,IA}^n)^2} & 0 & \mathbf{0}_{1 \times 12} \\ \frac{\Delta p_{N,IA}^n}{\sqrt{(\Delta p_{N,IA}^n)^2 + (\Delta p_{E,IA}^n)^2}} & \frac{\Delta p_{E,IA}^n}{\sqrt{(\Delta p_{N,IA}^n)^2 + (\Delta p_{E,IA}^n)^2}} & 0 & \mathbf{0}_{1 \times 12} \\ 0 & 0 & 1 & \mathbf{0}_{1 \times 12} \end{bmatrix} \quad (4.22)$$

The first component in (4.21) uses the heading angle to correct the course angle error calculated from the position. They have different stochastic characteristics, but the long-term characteristics are sufficiently similar that the difference can be neglected. Therefore, the above measurement can be used as above, since the course angle can be assumed to be dominated by the sensor heading error [125].

PCA measurements - mismatched case

The problem in the PA position is that it only considers device heading, not walking direction. If there are different poses such as putting the phone in the shirt or trouser pocket, the walking direction does not match with the device heading as Fig. 4.6b. In this case, there is heading offset between walking direction and device heading. For the position from conventional PA shown in orange, errors are continuously generated because the position is estimated based on the device heading. However, in a mismatch situation, you can update

the position of the IA in green through the position calculated by the PCA of the acceleration vector, this position marked blue in the Fig. 4.6b.

As mentioned, for the mismatched heading case, the walking direction is calculated from PCA. PCA is a technique that finds new bases orthogonal to each other while preserving the variance of data as much as possible and transforms samples from high-dimensional spaces into low-dimensional spaces without linear correlation [92, 126].

This approach takes advantage of the fact that the user's motion axis correlates with the largest variance axis in the horizontal acceleration that can be determined by PCA [38, 127–129]. In order to find the unit vector of walking direction, \mathbf{u}_1 , which is the largest variance in the horizontal acceleration data \mathbf{f}_m^n where $m = 1, \dots, M$. Each data point \mathbf{f}_m^n is projected on to a scalar value $\mathbf{u}_1^T \mathbf{f}_m^n$, and the mean of the projected data is $\mathbf{u}_1^T \bar{\mathbf{f}}^n$ where $\bar{\mathbf{f}}^n$ is sample set means represented as follows.

$$\bar{\mathbf{f}}^n = \frac{1}{M} \sum_{m=1}^M \mathbf{f}_m^n \quad (4.23)$$

The variance of the projected data and the data covariance \mathbf{S} are respectively given by

$$\frac{1}{M} \sum_{m=1}^M \{\mathbf{u}_1^T \mathbf{f}_m^n - \mathbf{u}_1^T \bar{\mathbf{f}}^n\}^2 = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 \quad (4.24)$$

$$\mathbf{S} = \frac{1}{M} \sum_{m=1}^M (\mathbf{f}_m^n - \bar{\mathbf{f}}^n) (\mathbf{f}_m^n - \bar{\mathbf{f}}^n)^T. \quad (4.25)$$

In order to maximize the projected variance in (4.24) with respect to \mathbf{u}_1 , Lagrange multiplier λ_1 is introduced with the unconstrained maximization of (4.26).

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1) \quad (4.26)$$

By setting the derivative with respect to \mathbf{u}_1 equal to zero, \mathbf{u}_1 is an eigenvector of \mathbf{S} .

$$\mathbf{S}\mathbf{u}_1 = \lambda_1\mathbf{u}_1 \rightarrow \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 = \lambda_1 \quad (4.27)$$

Therefore, the variance will be a maximum when \mathbf{u}_1 is equal to the eigenvector having the largest eigenvalue λ_1 .

There are various PCA based variants such as PCA2D, PCA2Df, PCA3Df, and gyroPCA to get the user's motion axis [38]. PCA2D is a PCA applied to the window on the 2D acceleration axis obtained by projecting onto the horizontal plane. The first eigenvector, which is the largest eigenvalue means a walking direction. Next, PCA2Df is almost same as PCA2D, but before applying PCA, the acceleration is low pass-filtered at 5 Hz to remove noise. According to [38] the 5 Hz average filter basically eliminates the body shaking noise, but it shows the best results in a series of tests that maintained the acceleration signal due to body movement. In addition, PCA3Df applies PCA to the 3D acceleration axis (converted to a n-frame) and then projects a third order eigenvector (the smallest eigenvalue) onto the horizontal plane. Lastly, gyroPCA [127] applies PCA of gyroscope measures in n-frame to extract the user's relative orientation over time.

Among those, PCA2Df is proved to be most accurate in [38], so the walking direction is obtained from the PCA2Df in this dissertation. To address the 180 degree ambiguity inherent in the direction coming out of the PCA above, we adjust it with the direction through IA-based PDR.

Using the walking direction obtained from both IA and PCA, the IA states are updated as in (4.28).

$$\mathbf{z}_{mismatch} = \begin{bmatrix} WD_{IA} - WD_{PCA} & SL_{IA} - SL_{PA} & \Delta p_D^n \end{bmatrix}^T \quad (4.28)$$

As seen from the above equations (4.25), sensor errors such as accelerometer

Modes	Description
Standstill (Mode 1)	- During initial alignment - Zero velocity update
Match (Mode 2)	- Walking direction and device heading match (Text pose) - Measurement from <u>PA position</u>
Transition/turn (Mode 3)	- Rotating in place, pose transitions - No measurement update
Mismatch (Mode 4)	- Poses other than text (shirt, trouser, and swing) - Measurement from <u>PCA position</u>

Figure 4.7: Measurement updating mode

bias ∇^n and noise \mathbf{w}_f^n are neglectable if the acceleration by pose is large enough to ignore the errors by the quadratic term inside of summation (4.29).

$$(\mathbf{f}_m^n + \nabla^n + \mathbf{w}_f^n - \bar{\mathbf{f}}^n) (\mathbf{f}_m^n + \nabla^n + \mathbf{w}_f^n - \bar{\mathbf{f}}^n)^T \quad (4.29)$$

However, it should be noted that the PCA is done in the navigation frame. Therefore, it is essential to minimize the error of navigation frame acceleration through the attitude estimation method estimated in chapter 3.

4.3.3 Mode selection

In this paper, there are a total of four modes to decide how to update measurements as in Fig. 4.7. As in Fig. 4.8 and mentioned in chapter 4.1, the user initially stands for about 5 seconds and performs alignment for initial attitude. During this phase, the mode is recognized as standstill, and the zero velocity correction as (4.13) is performed. When the user starts walking with text pose, in which case the state variables of the IA are corrected to the position obtained from the PA as (4.21). These are scenario constraints for this algorithm mentioned in chapter 4.1. In the process of switching from text pose to the others, or in a situation such as rotation in place, the transition

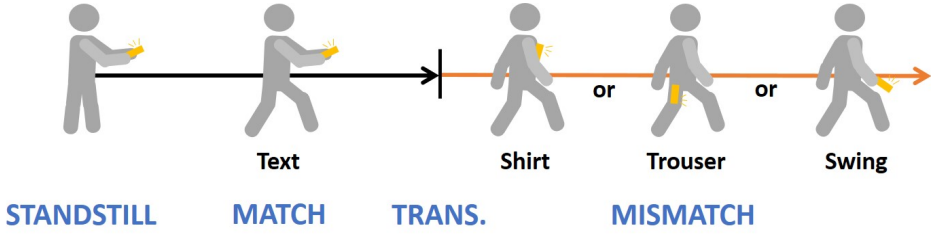


Figure 4.8: Walking scenarios and corresponding modes

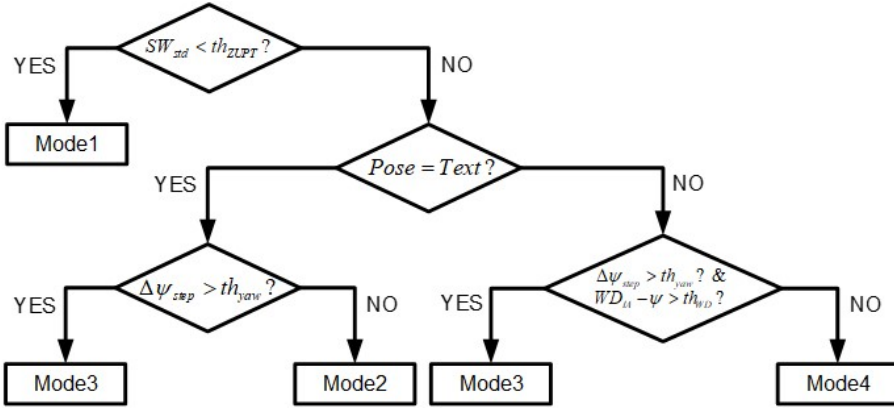


Figure 4.9: Mode decision flow

mode works. In this case, the position is obtained by propagation of IA without performing measurement update. Lastly, if there is a discrepancy between the direction of walking and the direction of the device, the walking direction is calculated as PCA to correct the state variables of IA as (4.28).

In selecting four modes, it uses an accelerometer, classified pose, the difference between the walking direction and the device heading, and the angle difference between two steps as Fig. 4.9. Firstly, the standard deviation of sliding windowed accelerometer data is used to determine standstill mode, which is represented as mode 1. When it is below the pre-determined threshold, it can be seen as standstill. Otherwise, it determines whether the pose is text through

pose recognition. Even if it is decided as text, it still remains whether the user is turning or not. This is because the measurement update is not performed during the turning phase, which is mode 3, and mode 2 updates the position as (4.21). These two modes are chosen using the angle difference between two steps which indicates the turning of the user when it is above the threshold. In case of non-text poses, shirt pocket, trouser pocket, and swing in this dissertation, it can be considered as mismatch phase. Same as before, the measurement is not updated under the transition and turning condition, so it has to be determined. In addition to the angle difference between two steps, a difference between the direction of walking and the device heading is used for deciding mismatch mode. It is noted that each threshold used for mode decision is saved in advance.

4.3.4 Observability analysis

In this section, the observability analysis of the proposed algorithm is performed. One of the reasons for the analysis of the observability of a dynamic system is the need to determine the effectiveness of the Kalman filter designed to estimate the state of that system. The ability to estimate the state of a fully observable system depends only on system driving noise and measurement noise. On the other hand, if the system is unobservable, ignoring the noise level will not give us an accurate estimate of the condition. In other words, observability sets a lower bound on the estimation error, and the lower the limit, the better the chance to get an accurate estimate of the system states.

The analysis of the observability of the constant dynamic system is rather simple, but the analysis of the time-varying system is quite cumbersome and requires the evaluation of the observability Grammian. The calculation of the observability Grammian is performed numerically rather than analytically. As a result, it is difficult to study the properties of a system and derive general

rules governing it.

There are cases where the time-varying system has little loss of accuracy and can be approximated by a PWCS (Piece-Wise Constant System) (e.g. INS during alignment during flight) [130]. When doing the observability analysis of the system, it is the characteristics, not the exact time response that we are interested in. Therefore, the time-varying system can be replaced with PWCS for observability analysis.

Meanwhile, when using a linearized estimator, errors in the linearization process affect measurement Jacobians with respect to the direction of the information. If the incorrect information due to the linearization error exists along an unobservable direction, the uncertainty is small, even if it is impossible to observe.

The observability matrix for the linearized EKF system over the time step $[1, k]$ is defined as a function of the discrete time state transition matrix Φ and the linearized measurement matrix \mathbf{H} :

$$\mathcal{O}(\mathbf{x}) = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \Phi_{2,1} \\ \vdots \\ \mathbf{H}_k \Phi_{k,1} \end{bmatrix} \quad (4.30)$$

To compute and analyze the observability matrix of the proposed system, we first derive the state transition matrix following matrix differential equation:

$$\dot{\Phi}_{k,1} = \mathbf{F}_k \Phi_{k,1} \quad \text{where } \mathbf{F}_k = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \begin{bmatrix} \tilde{\mathbf{f}}^n \times \end{bmatrix} & \mathbf{C}_b^n & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathbf{C}_b^n \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (4.31)$$

By conducting a block-by-block examination of (4.31), its analytic solution can be obtained, which will be used for the observability analysis. The first order linearized solution at \mathbf{k} is in (4.12).

The null space of \mathcal{O} for the proposed algorithm following each measurement in chapter 4.3.2 explains in turn.

ZUPT

A rank test of zero velocity correction results in 8. The null space basis vector for position, heading, gyro bias in down axis of n-frame, and accelerometer bias are as follows.

$$\mathbf{N}_{ZUPT,1} = \begin{bmatrix} \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \end{bmatrix}, \delta \mathbf{x}' = \delta \mathbf{x} + \mathbf{N}_{ZUPT,1} \delta \mathbf{p} \quad (4.32)$$

$$\mathbf{N}_{ZUPT,2} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ 0 \\ 0 \\ 1 \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}, \delta \mathbf{x}' = \delta \mathbf{x} + \mathbf{N}_{ZUPT,2} \delta \psi \quad (4.33)$$

$$\mathbf{N}_{ZUPT,3} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{C}_{n,col3}^b \end{bmatrix}, \delta \mathbf{x}' = \delta \mathbf{x} + (\mathbf{N}_{ZUPT,2\&3}) \varepsilon_D^n = \delta \mathbf{x} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ 0 \\ 0 \\ 1 \\ \mathbf{0}_{3 \times 1} \\ \mathbf{C}_{n,col3}^b \end{bmatrix} \varepsilon_D^n \quad (4.34)$$

$$\mathbf{N}_{ZUPT,4} = \begin{bmatrix} \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{3 \times 2} \\ 0 & 1 \\ -1 & 0 \\ 0 & 0 \\ g & 0 \\ 0 & g \\ 0 & 0 \\ \mathbf{0}_{3 \times 2} \end{bmatrix} \quad (4.35)$$

For simplified representation of null space-based vectors, we assume that acceleromter bias and gyro bias are represented in NED frame. It is also noticed that the acceleromter bias is coupled with attitude.

As a result, given ZUPT measurement, the position, heading, gyro bias (down in n-frame), and coupled accelerometer bias (north and east in n-frame) are unobservable.

AHRS

A rank test of AHRS measurement results in 8. The null space basis vector for position, velocity, and accelerometer bias are as follows.

$$\mathbf{N}_{AHRS,1} = \begin{bmatrix} \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \end{bmatrix}, \delta \mathbf{x}' = \delta \mathbf{x} + \mathbf{N}_{AHRS,1} \delta \mathbf{p} \quad (4.36)$$

$$\mathbf{N}_{AHRS,2} = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \end{bmatrix}, \delta \mathbf{x}' = \delta \mathbf{x} + \mathbf{N}_{AHRS,2} \delta \mathbf{v} \quad (4.37)$$

$$\mathbf{N}_{AHRS,3} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ m_x \\ m_y \\ m_z \\ -g \cdot m_y \\ g \cdot m_x \\ 0 \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (4.38)$$

Similarly, for simplified representation of null space-based vectors, accelerometer bias is again represented in NED frame. It is also noticed that the accelerometer

bias is coupled with attitude. As a result, given AHRS measurements, the position, velocity, and coupled accelerometer bias (north and east in n-frame) are unobservable.

PA measurements

A rank test of IA-PA which is (4.21) and (4.28) results in 11. The null space basis vector for heading, gyro bias in down axis of n-frame, and accelerometer bias are as follows.

$$\mathbf{N}_{IAPA,1} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ 0 \\ 0 \\ 1 \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}, \delta \mathbf{x}' = \delta \mathbf{x} + \mathbf{N}_{IAPA,1} \delta \psi \quad (4.39)$$

$$\mathbf{N}_{IAPA,2} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{C}_{n,col3}^b \end{bmatrix}, \delta \mathbf{x}' = \delta \mathbf{x} + (\mathbf{N}_{IAPA,1 \& 2}) \varepsilon_D^n = \delta \mathbf{x} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ 0 \\ 0 \\ 1 \\ \mathbf{0}_{3 \times 1} \\ \mathbf{C}_{n,col3}^b \end{bmatrix} \varepsilon_D^n \quad (4.40)$$

$$\mathbf{N}_{IAPA,3} = \begin{bmatrix} \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{3 \times 2} \\ 0 & 1 \\ -1 & 0 \\ 0 & 0 \\ g & 0 \\ 0 & g \\ 0 & 0 \\ \mathbf{0}_{3 \times 2} \end{bmatrix} \quad (4.41)$$

As before, for simplified representation of null space-based vectors, accelerometer and gyro bias are again represented in NED frame. It is also noticed that the accelerometer bias is coupled with attitude in (4.41). As a result, given IA-PA for both match and mismatch measurements, the heading, gyro bias (down in n-frame), and coupled accelerometer bias (north and east in n-frame) are not observable.

In conclusion, if all the measurements are used at once, it becomes fully observable. In the case of accelerometer bias associated with attitude in all measurements, the observable varies depending on the attitude.

4.4 Experimental Results

4.4.1 AHRS results

As mentioned earlier, the IA and PA fusion proceeds under the assumption that the attitude estimation algorithm proposed in chapter 3 and the gyro bias error are well estimated. To confirm this assumption, the estimated attitude comparison with the one provided by Xsens MTw is performed. In addition, in

Table 4.5: Attitude error compared with Xsens MTw

[deg]	roll	pitch	yaw
attitude error	1.99	1.59	2.99

Table 4.6: Gyro bias error compared with stationary data

[rad/s]	ϵ_x	ϵ_y	ϵ_z
gyro bias error	0.908×10^{-3}	1.42×10^{-3}	0.715×10^{-3}

Table 4.7: Initial gyro bias set

	ini. ϵ [rad/s]
case 1	$\begin{bmatrix} 0.0 & -0.0 & -0.0 \end{bmatrix} * 1e - 3$
case 2	$\begin{bmatrix} 1.0 & -1.0 & -0.5 \end{bmatrix} * 1e - 3$
case 3	$\begin{bmatrix} 2.0 & -2.0 & -1.0 \end{bmatrix} * 1e - 3$
case 4	$\begin{bmatrix} 3.0 & -3.0 & -1.5 \end{bmatrix} * 1e - 3$
case 5	$\begin{bmatrix} 4.0 & -4.0 & -2.0 \end{bmatrix} * 1e - 3$

the case of the gyro bias, the comparison is performed with the gyro bias as a true value with the sensor resting on the flat floor before the experiment.

The attitude accuracy provided by Xsens is as follows. Roll and pitch are 0.75 degrees in dynamic situations, and yaw is 1.5 degree. From the results in Table 4.5 and 4.6, it can be seen that the attitude and the gyro bias are also well estimated to satisfy the assumptions mentioned in this paper. Also, even if the initial gyro bias value is changed as shown in the Table 4.7, it converges as shown in the Fig. 4.10.

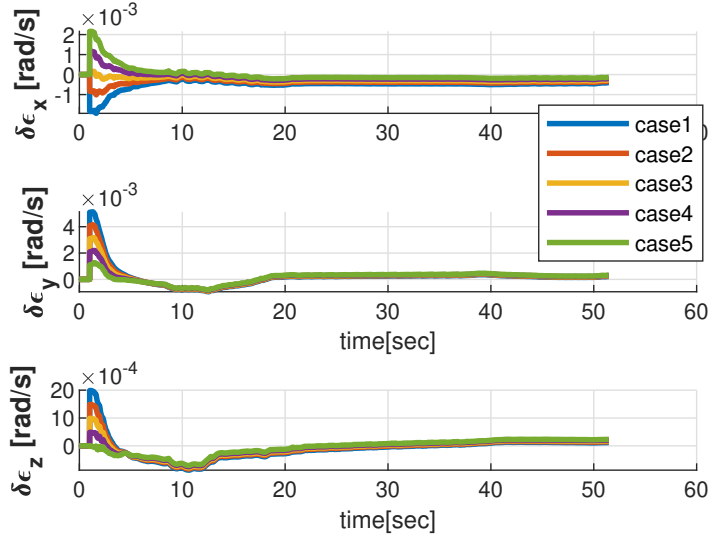


Figure 4.10: Estimated gyro bias error in different initial values

4.4.2 PCA results

To check the performance of the walking direction calculated by PCA, the dataset in chapter 4.2 is used. In short, the experimental scenario is a 25m round trip linear trajectory, and the walking speed is 80, 90, 95, 100, and 105 BPM. The number of subjects participating in this experiment is 8 males and 4 females, a total of 12 subjects. Subjects are from 23 to 40 years of age with no physical disability. The sensor used is the Xsens MTw. This is because this sensor can be wirelessly attached to multiple locations on the body and get synchronized data from all sensors. In this thesis, the sensors are attached to the shirt pocket, hand, trouser pocket, and shoes. The sensor on the shoe is used to get an accurate heel strike point.

When the heel strike point is extracted from the shoe, the acceleration in the navigation frame during 1 stride, \mathbf{f}^n , is calculated using the attitude provided

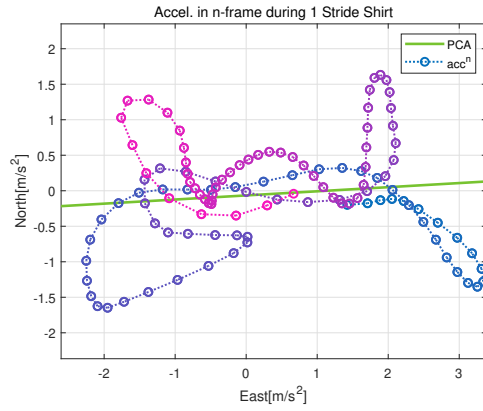
Table 4.8: Walking direction by PCA results

error[deg]	Mean	Std.	25% tile	Median	75% tile	95% tile
Shirt	-0.203	5.64	1.16	3.24	6.37	12.0
Trouser	0.025	3.21	0.706	2.02	3.65	7.00
Swing	-0.100	5.97	1.29	3.39	6.48	13.2

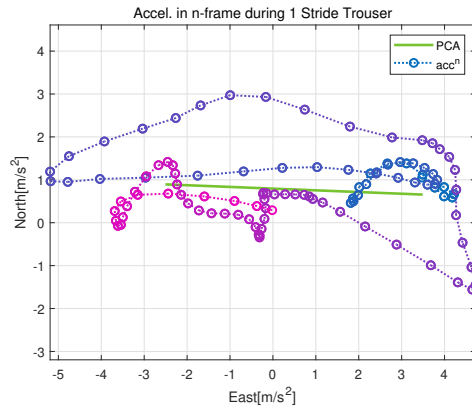
by MTw as in Fig. 4.11. The figure shows the acceleration of the north and earth axes during 1 stride, and over time the acceleration data changes from blue to magenta. The green line shows 180 degree ambiguity in the direction of progress calculated by PCA. Also, as you can expect, the shirt pocket fixed to the upper body has the lowest acceleration and the greatest acceleration in the swing situation where the hand is moved greatly. The result of Fig. 4.11 shows that although there is a difference in the magnitude of the acceleration, the direction of progress can be extracted through PCA.

PCA results from 12 subjects are shown in the Fig. 4.12 and Table 4.8. The results about 120 strides per person are checked, except for errors due to body movement, errors in the starting, ending, and in-situ rotation situations. In each box in the Fig. 4.12, the central mark represents the median, and the bottom and top edges of the box represent the 25th and 75th percentiles, respectively. The whiskers are extended to the most extreme data points that are not considered outliers, and outliers are displayed individually using the + symbol.

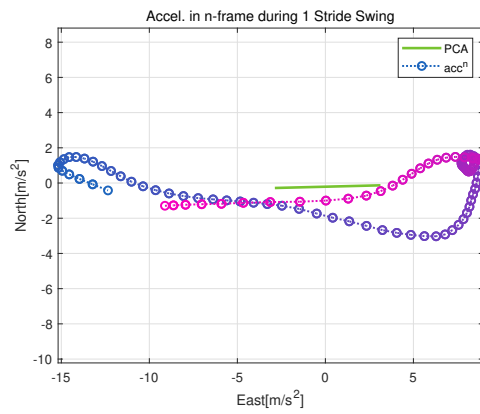
Overall, the best results are when the device is in the pants pocket. This is because in the pocket, the sensor is fixed in the pocket and is completely caused by the movement of the leg, so it can indicate the direction of progress. On the other hand, the shirt pocket is fixed inside the pocket, but it seems that



(a) shirt pocket

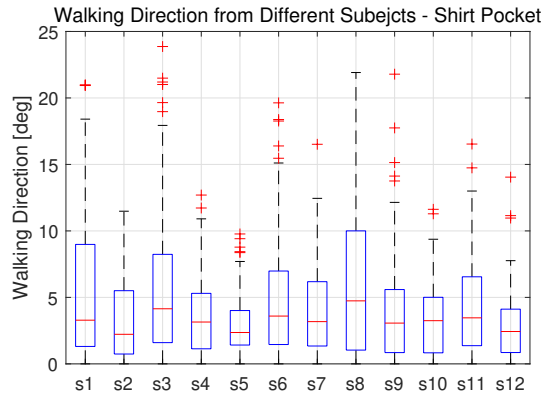


(b) trouser pocket

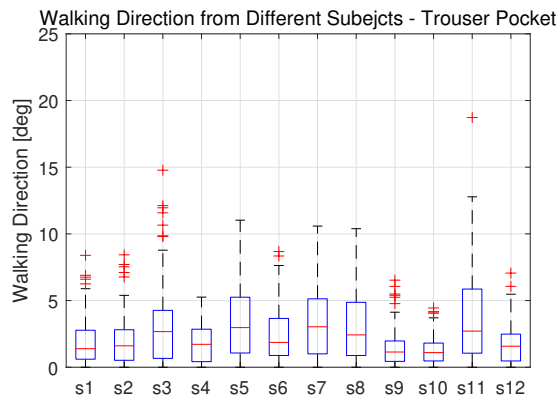


(c) swing

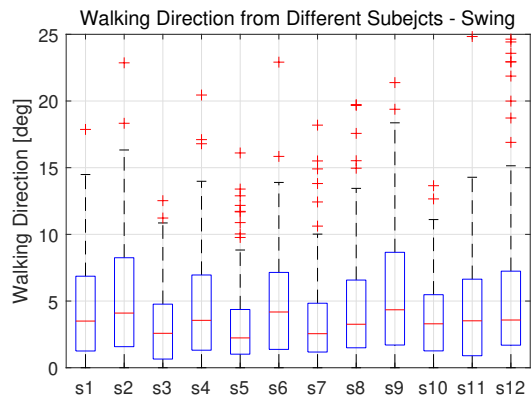
Figure 4.11: PCA and acceleration (n-frame) for 1 stride



(a) shirt pocket



(b) trouser pocket



(c) swing

Figure 4.12: PCA results in box whisker plot

an error occurred because the upper body movement is relatively small. In the case of swing, the result is that the volatility is largely due to the movement of the hand because the degree of freedom is the largest.

4.4.3 IA-PA results

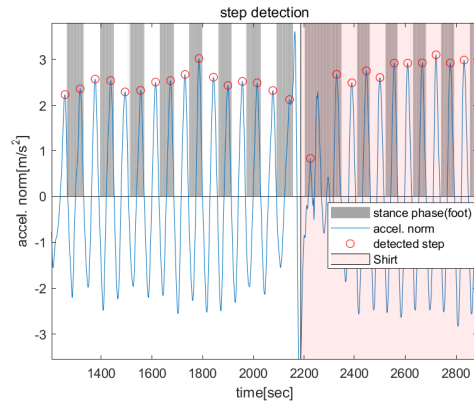
To compare and prove the performance of the IA and PA-based PDR system, the experiment is performed also with the IMU module, Xsens MTw [120, 124]. The three trajectories are tested. The first trajectory is a straight trajectory of 40m one way performed to check the results of the PCA under mismatch case, and the second trajectory is an L-shaped trajectory of 58m round trip to check whether the proposed algorithm works well even when rotation occurs in a pose other than text. The trajectory is a square trajectory of 194.6m one way to confirm that there is no problem even after operating for a long time.

In the IA-PA fusion system, each of IA and PA must be well estimated for the best results, so the results for each component are presented as follows.

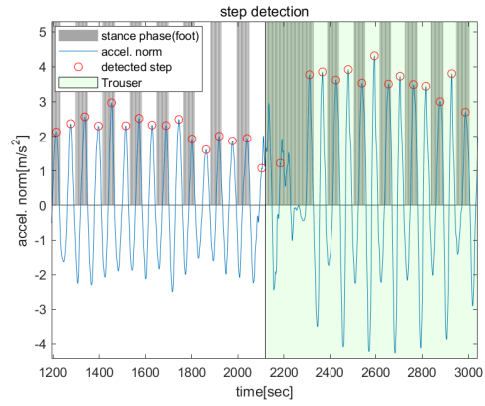
As mentioned before, the trajectory 1 as in Fig. 4.13 is tested to check the feasibility of the proposed algorithm. Firstly, whether step detection is correct or not for each pose is checked as Fig. 4.14. Two MTw sensors are synchronized and tested at the same time, one attached to the right shoe and used to obtain the correct true value, and one operated like a mobile phone. It has already been proven that step detection using the peak value of acceleration magnitude works well in text situations. Similar to other target poses, it is easy to detect



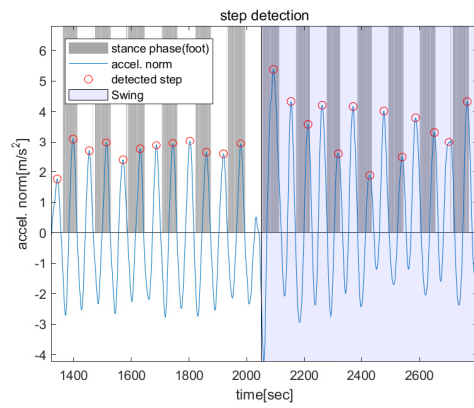
Figure 4.13: Trajectory #1



(a) shirt pocket



(b) trouser pocket



(c) swing

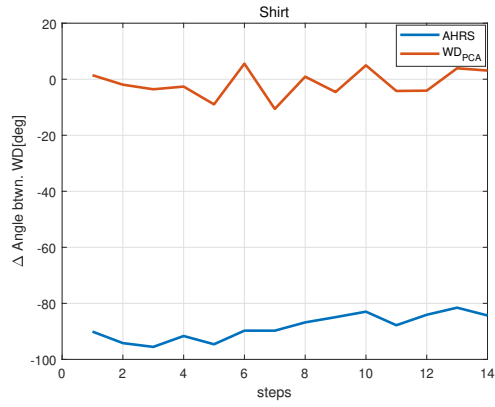
Figure 4.14: Step detection results

steps using the magnitude of acceleration. The gray areas in each figure in Fig. 4.14 are the stance phase intervals (per stride) detected by shoes, the blue is the magnitude of acceleration, and the red is the step detected by the step detection technique. In addition, the shirt pocket section is red, the pants pocket is green, and the swing is blue, respectively. As looking at Fig. 4.14, it can be seen that there is a step misdetection during the transition between poses, but the steps are well detected in other situations.

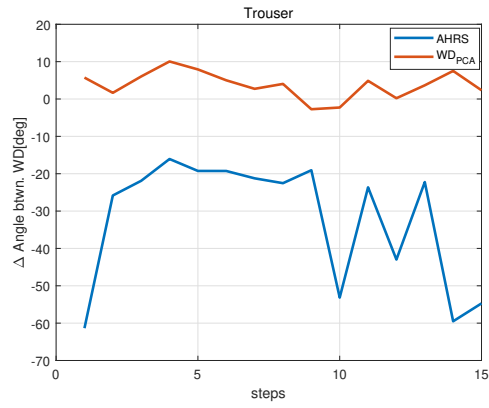
In case of step length and heading, there are many features and functions for estimating the step length other than the walking frequency, but it is advantageous for the independence of the mounting position as long as the step is accurately detected. The (4.18) is the step length formula applied. For heading, the estimated attitude results are compared with the attitude provided by MTw because there are no reference.

Next, it is confirmed whether the walking direction obtained by the PCA in the pose other than the text is correct. Fig. 4.15 shows the result of the difference between the walking direction of each stride and the walking direction calculated by PCA, and the difference between the walking direction and AHRS in non-text pose when walking trajectory 1. When the pose changes as shown in the figure, the position and attitude of the device changes, and the walking direction and the device direction do not match. On the other hand, in the case of walking direction calculated through PCA, you can see that it gives information about the direction of walking.

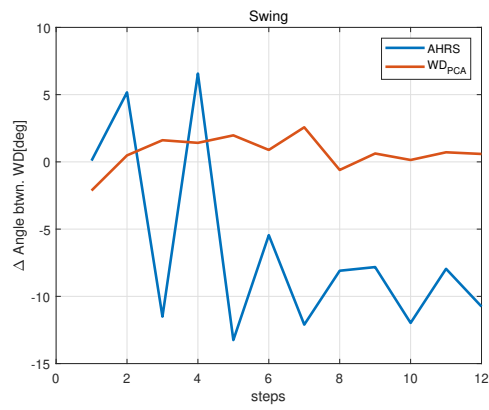
The following Fig. 4.16 and Table 4.9 the results of position estimation calculated for 10 subjects from the proposed algorithm. In Fig. 4.16, red indicates the proposed algorithm, and black indicates the trajectory of the PA-based algorithm. When the pose changed, it is marked with magenta and gray, respectively. Through the proposed method, the problems that occur when the



(a) shirt pocket

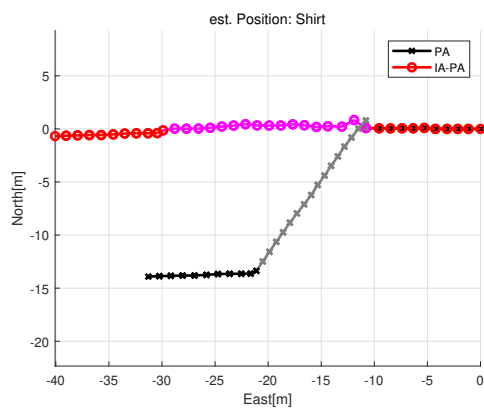


(b) trouser pocket

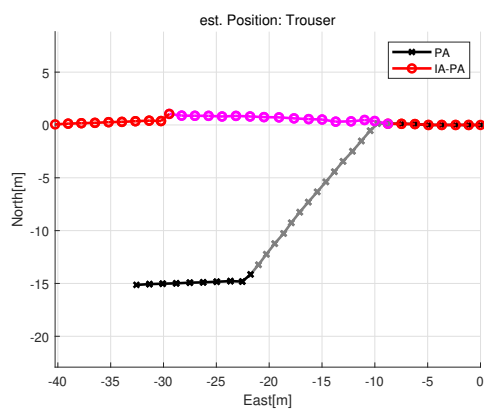


(c) swing

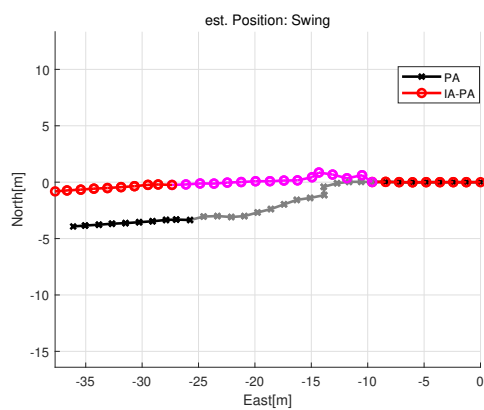
Figure 4.15: Walking direction difference of PCA and AHRS



(a) shirt pocket



(b) trouser pocket



(c) swing

Figure 4.16: Estimated position

Table 4.9: Position results for trajectory #1

Pos. Error[m]	PA only		IA-PA	
	Mean	Std.	Mean	Std.
Shirt Pocket	15.1	3.86	0.881	0.698
Trouser Pocket	8.99	5.72	0.920	0.442
Swing	2.59	2.59	0.999	0.686

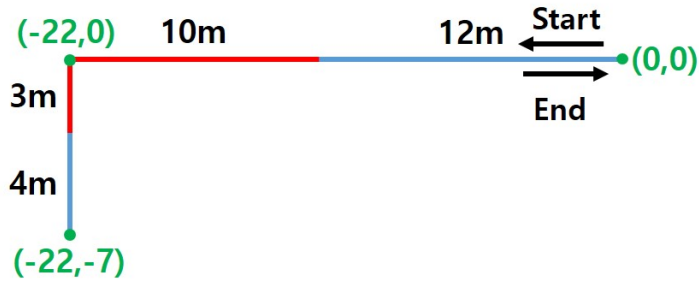
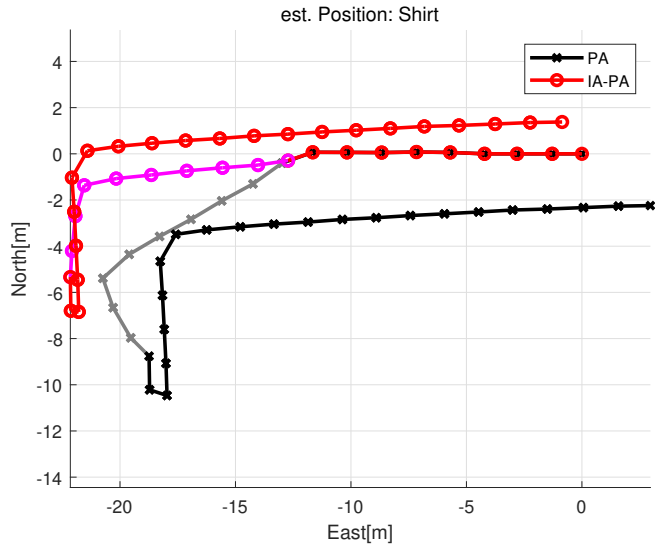


Figure 4.17: Trajectory #2

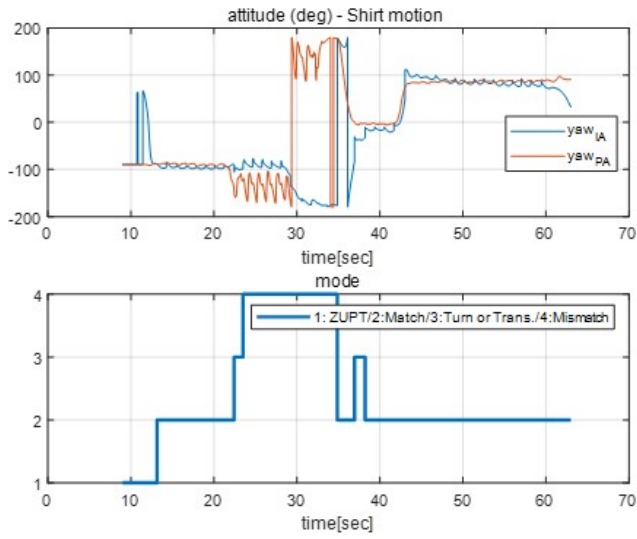
existing PA-based PDR estimation changes its poses can be solved.

The walking scenario 2 is a total of 58 meters of L-shaped round-trip tracks, as in Fig. 4.17, with the subject holding the sensor in hand and walking in the area marked in red so that the walking direction does not match the sensor. In addition, text pose with no mismatch case throughout the experiment is also tested. The position and attitude estimation results for those poses are in Fig. 4.18-4.20.

As seen from the position results in Fig. 4.18a, in the beginning, the results of PA and IA are the same during the text pose. The text pose is a situation in which the direction of movement is the same as the direction of the device, so the IA position is updated by the PA position. Therefore, the walking direction of



(a) Position



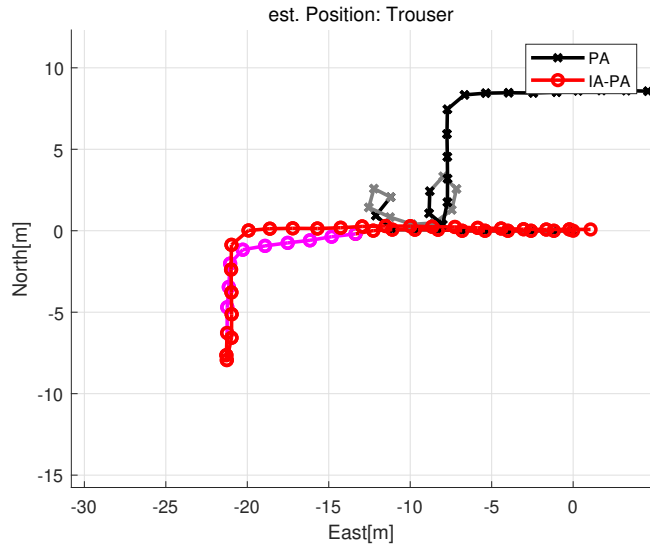
(b) Attitude and mode

Figure 4.18: Position and attitude results for shirt pocket pose

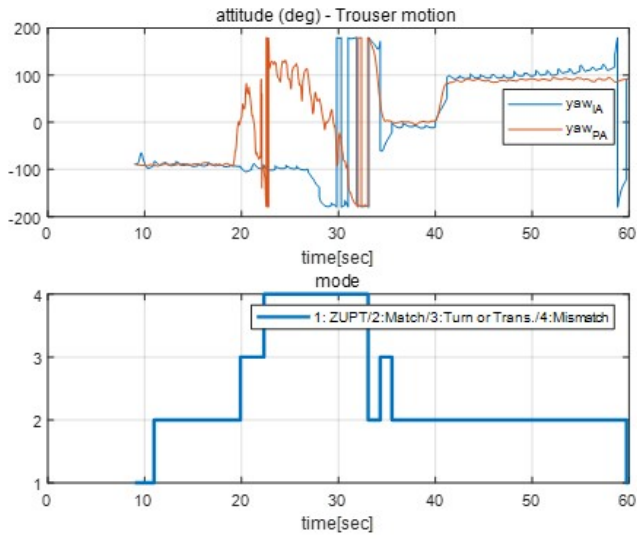
IA calculated from the n-frame position and PA from the AHRS also correspond as in Fig. 4.18b. When the device is put on the shirt pocket, the position of PA is no longer corresponds to the walking direction, so the mode is changed following the difference as transition or mismatch as in Fig. 4.18b. Mode 3 includes the transition of poses or turning of the subject, and it shows a shorter step length than the regular walk between steps. After the transition phase, the tester walks straight with the device tilted, so the mode changes into the mismatch one. In Fig. 4.18b, it is noticeable that the device heading marked as yaw_{PA} does not match with the walking direction represented as yaw_{IA} . When the device returns to the match mode, the position of the proposed method follows the characteristics of the PA. As seen from the results, the proposed algorithm is able to detect the device condition and estimate position by considering the heading difference between walking direction and heading. It is also noticeable that position errors are reduced using the proposed algorithm when the tester is rotating in place. This is because the step length and the walking direction are accurately estimated based on the n-frame position, unlike the period-dependent PA stride.

Similarly, when the device is placed in the trouser pocket, the heading mismatch also occurs as in Fig. 4.19a. When the sensor is placed in the trouser pocket, there are large attitude changes due to the repetitive leg movements. As we are using the mean heading of two consecutive steps, the swaying heading angle from the leg does not affect the position results. In addition, the results show that the proposed algorithm works even under the trouser pocket poses.

This algorithm can be applied to swing pose as shown in Fig. 4.20a. In the case of swing, if the step detection is performed correctly, the walking direction result of the PA is not bad, but there are disadvantages that it is unstable, so the proposed algorithm can obtain more stable position estimation results.

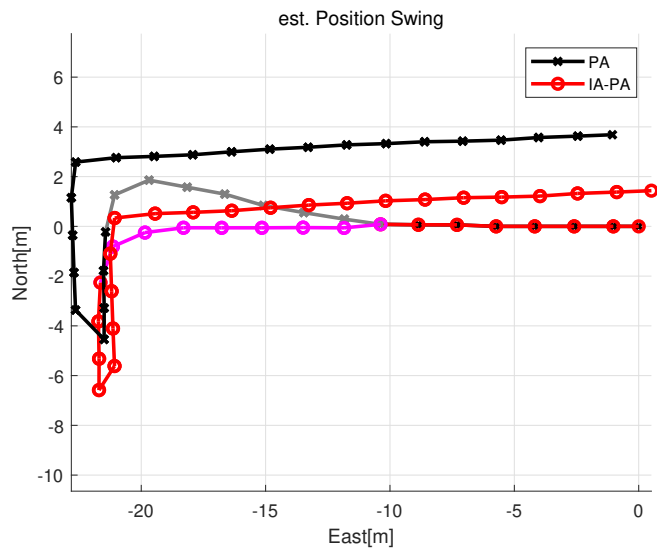


(a) Position

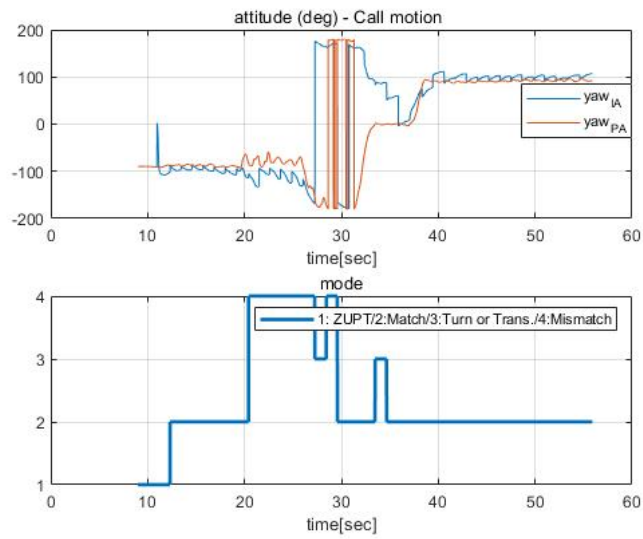


(b) Attitude and mode

Figure 4.19: Position and attitude results for trouser pocket pose



(a) Position



(b) Attitude and mode

Figure 4.20: Position and attitude results for swing pose

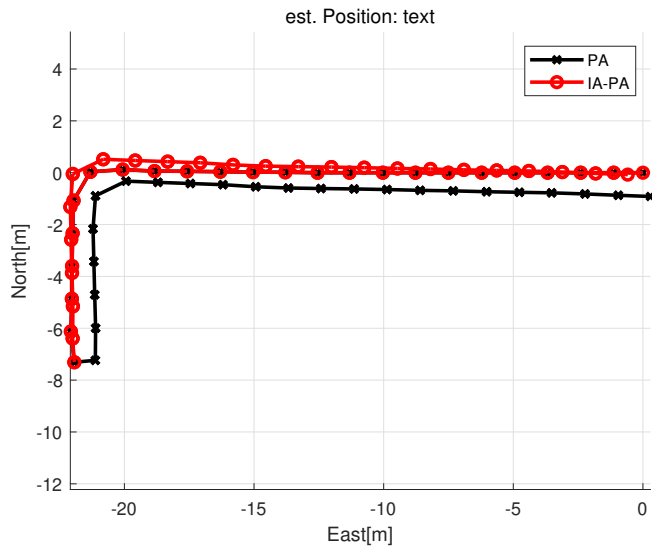
Table 4.10: Position results for trajectory #2

RPE [m]	PA only	IA-PA
Shirt pocket	5.05	0.98
Trouser pocket	15.69	1.42
Swing	2.64	1.05
Text only	0.97	0.43

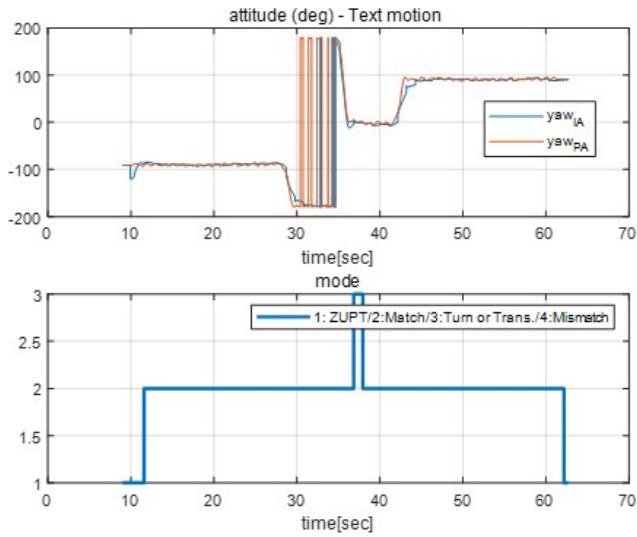
The position result in Fig.4.20a, however, shows one of the limitations of the proposed algorithm that the IA results follow the PA features. To be specific, the step length and heading of the device are incorrect; the measurement updates of those lead to position errors. Therefore, the correct step length and device attitude estimation of PA is the prerequisites of the proposed algorithm.

Lastly, the trajectory with the only handheld case is tested as in Fig. 4.21a. As seen from the figure, the proposed algorithm is also advantageous during rotating in place. The conventional PA algorithm calculates the step length only based on the walking frequency, so errors occur in sections with short strides, such as rotating in place.

The RPE (Return Position Error) results of multiple poses are in Table 4.10. The above results show that the algorithm can be applied in other situations wherever there is a heading mismatch. The proposed algorithm seems to work well, but if the walking direction from PCA is incorrect, the heading error will increase significantly over time. In this case, additional heading measurements, such as the dominant direction, can improve the performance of the IA, which is considered as future work. In addition, since the IA position error is compensated based on the performance of the step length from the PA, an average reasonable length estimation is still required.



(a) Position



(b) Attitude and mode

Figure 4.21: Position and attitude results for text only pose

The trajectory 3 is performed to see that the proposed algorithm works well even on the long trajectory, and the trajectory performed at Building 39 in Seoul National University is rectangular-shaped of 194.6m, as shown in Fig. 4.22a. The text trajectory is blue, and the rest of the poses are in the order of shirt pocket, trouser pocket, and swing, and the section is marked in orange. The result of running the proposed algorithm for a long time is shown in the Fig. 4.22b. In the figure, red is the proposed algorithm, blue is the result of the PA-based algorithm, and when the pose changes, it is drawn with magenta and cyan, respectively. In the case of position error, it is calculated based on four rotating sections and the arrival point, and the proposed algorithm has a position error of **2.13m** and PA of **20.01m**. This shows that even after walking for a long time, the proposed algorithm yields good results.

I would like to mention that the proposed algorithm has some limitations to be improved. As mentioned earlier, the IA states are estimated based on the PA measurements, so the PA position results by itself should be accurate as possible. Therefore, the additional heading information from other sources such as dominant direction of the building, BLE, Wi-Fi, etc. enables to use the proposed algorithm for a long time. In addition, the mode is determined based on walking direction and device heading, so the slight difference between those could leads to the wrong position errors.

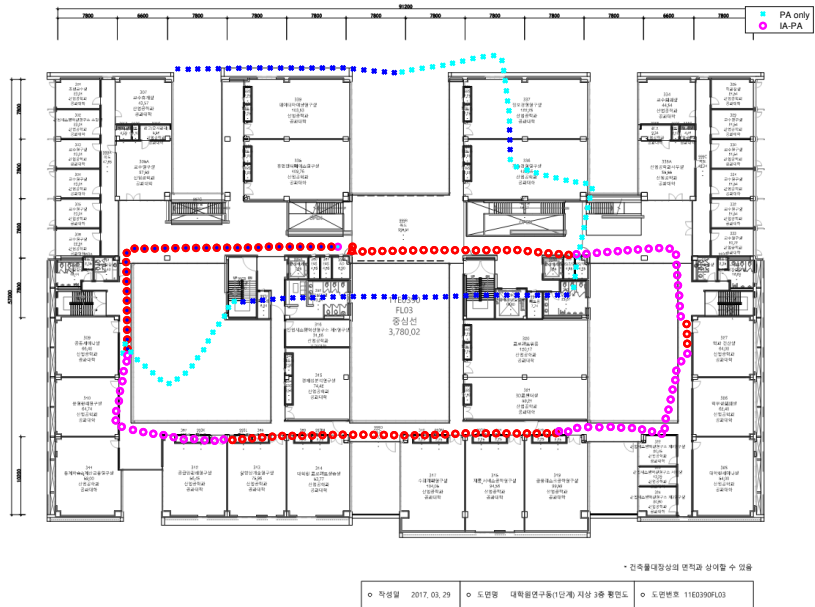
4.5 Summary

In this section, I propose the PDR system with the fusion of the IA and PA for seamless smartphone position estimation. The algorithm is proposed in order to avoid the severe position errors from the difference between the walking direction and the device heading. The fusion algorithm estimates the position

from the IA-based PDR system, with the measurement of step length from the PA and device heading under the correspondence mode. In case of the heading mismatch condition, the PCA from steps provides the walking direction, so the measurement is continuously updated during mismatch case. The proposed algorithm shows its effectiveness in the experiments that position is correctly estimated under shirt pocket, text, swinging, and trouser pocket pose.



(a) Trajectory #3



(b) Estimated Position

Figure 4.22: Position result for trajectory #3

Chapter 5

Conclusions

5.1 Summary of the Contributions

In this dissertation, the fusion of IA and PA-based PDR system is newly proposed for multiple poses in the smartphone, and the adaptive attitude estimation using ellipsoidal method is also proposed. The overview of PDR is presented in chapter 2, and the PA and IA are described. Both for the PA and IA-based PDR system, it is important to have accurate attitude estimation. Therefore, in chapter 3, adaptive attitude estimation using ellipsoidal method is proposed. The proposed algorithm considers the direction of the measurement residuals, so it has a better accuracy than the conventional adaptive estimation technique without considering the direction. The conventional PA for smartphone parametrically estimates the stride based on the step detection and calculates the position using the device heading at that time. It suffers from the attitude and position errors caused by device placements or motions. Especially when the walking direction and the device do not match, it causes severe problems in position estimation. In order to solve this problem, this dissertation proposes the algorithm that constructs state variables based on the IA and uses the position vector from the PA as a measurement in chapter 4. If the walking direction and the device heading do not match based on the pose recognized through machine learning technique, the position is updated in consideration of

the direction calculated using PCA and the step length obtained through the PA.

The results and contributions of this dissertation are summarized as follows:

1. First, in order to accurately and consistently estimate the attitude of a smartphone device for indoor navigation, an adaptive attitude estimation using an ellipsoidal method is proposed. Residual vectors acquired from accelerometers and magnetometers are used to deal with measurement errors from the acceleration and magnetic disturbance in AHRS. When the disturbance occurs, the measurement covariance is inflated to generate covering ellipsoid by comparing the residual vectors and measurement noises. The inflated measurement covariance is used in EKF to estimate attitude errors and gyro bias errors. The proposed attitude algorithm is confirmed to work accurately in the rate table, hand rotation with visual markers, and magnetic disturbing experiments.
2. The fusion of the IA and PA for position estimation in PDR is designed. There are many degrees of freedom when using the smartphone to move indoors, so there is a variety of poses, such as sending text messages and putting it in pants or shirt pockets. In the existing smartphone-based positioning algorithm, the position is estimated by the PA-based PDR, and it can be used only when the pedestrian's walking direction and the device's direction match, otherwise the position error due to mismatch angle error is large. To solve this problem, this paper proposes an algorithm that constructs state variables based on IA and uses the position vector of the PA as a measure. To be specific, the IA-based PDR states are estimated using measurements from PA-based PDR and

PCA. Using the classified poses from machine learning technique and the heading difference, it is possible to choose when to use PCA-based angle measurement. If the walking direction and the device heading do not correspond based on the pose recognized through machine learning technique, the position is updated in consideration of the direction calculated using PCA and the step length obtained through the PA. The EKF uses ZUPT, AHRS, PA-based position, and PCA to estimate and correct the error states of IA. The proposed algorithm, as a result, ensures observability of error states and position accuracy. Through the experiments under various conditions and the four poses (text, shirt pocket, trouser pocket, and swing), the proposed algorithm robustly and continuously estimates the position.

5.2 Future Works

The proposed algorithm can be improved on the two aspects.

- **Step length estimation in the PA-based PDR**

The proposed IA-PA PDR fusion algorithm is dependent on the step length estimation performance of the PA. In proposed algorithm, the step length from the PA is assumed to be accurate, so the position of the IA is limited to the accuracy of the PA position.

- **Adjusting with external information**

If there is a small angle difference between walking direction and device heading during text message pose, the proposed algorithm just follows the attitude of the device. In addition, it is reasonable to use the walking direction obtained from the PCA as a measurement, but it is better to further adjust the heading using other information such as the dominant

direction and multiple virtual tracks. This helps in updating the wrong measurements due to errors in the PCA. In addition, external location information such as Wi-Fi and map information can be used to adjust for accumulated location errors.

As mentioned earlier, the accuracy of the proposed algorithm is limited to the one of the PA-based step length. This means that accumulated step length errors are inevitable. Therefore, the external position source helps in scaling the step length parameters.

Bibliography

- [1] R. Harle, “A survey of indoor inertial positioning systems for pedestrians,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013.
- [2] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.
- [3] L. Mainetti, L. Patrono, and I. Sergi, “A survey on indoor positioning systems,” in *2014 22nd IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2014, pp. 111–120.
- [4] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, “Indoor human navigation systems: A survey,” *Interacting with Computers*, vol. 25, no. 1, pp. 21–33, 2013.
- [5] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “Landmarc: indoor location sensing using active rfid,” in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003).*, 2003, pp. 407–415.
- [6] Y. Gu, A. Lo, and I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks,” *IEEE Communications surveys & tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

- [7] J. Baus, A. Krüger, and W. Wahlster, “A resource-adaptive mobile navigation system,” in *Proceedings of the 7th international conference on Intelligent user interfaces*, 2002, pp. 15–22.
- [8] X. N. Fernando, S. Krishnan, H. Sun, and K. Kazemi-Moud, “Adaptive denoising at infrared wireless receivers,” in *Infrared Technology and Applications XXIX*, vol. 5074, 2003, pp. 199–207.
- [9] L. Ran, S. Helal, and S. Moore, “Drishti: an integrated indoor/outdoor blind navigation system and service,” in *Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the*, 2004, pp. 23–30.
- [10] K. Lorincz and M. Welsh, “Motetrack: A robust, decentralized approach to rf-based location tracking,” in *International Symposium on Location- and Context-Awareness*, 2005, pp. 63–82.
- [11] J. Hightower and G. Borriello, “Location systems for ubiquitous computing,” *computer*, vol. 34, no. 8, pp. 57–66, 2001.
- [12] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, “On indoor position location with wireless lans,” in *The 13th IEEE international symposium on personal, indoor and mobile radio communications*, vol. 2, 2002, pp. 720–724.
- [13] P. Bahl and V. N. Padmanabhan, “Radar: An in-building rf-based user location and tracking system,” in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, vol. 2, 2000, pp. 775–784.

- [14] O. Koch and S. Teller, “A self-calibrating, vision-based navigation assistant,” in *Workshop on Computer Vision Applications for the Visually Impaired*, Oct 2008.
- [15] W. Elloumi, K. Guissous, A. Chetouani, R. Canals, R. Leconge, B. Emile, and S. Treuillet, “Indoor navigation assistance with a smartphone camera based on vanishing points,” in *2013 IEEE International Conference on Indoor Positioning and Indoor Navigation*, 2013, pp. 1–9.
- [16] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, “Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments,” in *Unmanned Systems Technology XI*, vol. 7332. International Society for Optics and Photonics, 2009, p. 733219.
- [17] A. Howard, “Real-time stereo visual odometry for autonomous ground vehicles,” in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3946–3952.
- [18] M. Agrawal, K. Konolige, and R. C. Bolles, “Localization and mapping for autonomous navigation in outdoor terrains: A stereo vision approach,” in *Applications of Computer Vision, 2007. WACV’07. IEEE Workshop on*. IEEE, 2007, pp. 7–7.
- [19] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, “Vision-based hand pose estimation: A review,” *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 52–73, 2007.
- [20] R. Jirawimut, P. Ptasiński, V. Garaj, F. Cecelja, and W. Balachandran, “A method for dead reckoning parameter correction in pedestrian navigation system,” *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 1, pp. 209–215, 2003.

- [21] R. W. Levi and T. Judd, “Dead reckoning navigational system using accelerometer to measure foot impacts,” Dec. 10 1996, uS Patent 5,583,776.
- [22] Q. Ladetto, “On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive kalman filtering,” in *Proceedings of ION GPS*, vol. 2000, 2000, pp. 1735–1740.
- [23] S. Shin, C. Park, H. Hong, and J. Lee, “Mems-based personal navigator equipped on the user’s body,” 2005.
- [24] J. Kappi, J. Syrjarinne, and J. Saarinen, “Mems-imu based pedestrian navigator for handheld devices,” in *Proceedings of the 14th international technical meeting of the satellite division of the institute of navigation (ION GPS 2001)*, 2001, pp. 1369–1373.
- [25] K. Sagawa, M. Susumago, and H. Inooka, “Unrestricted measurement method of three-dimensional walking distance utilizing body acceleration and terrestrial magnetism,” *In Proceedings of the International Conference on Control, Automation and Systems*, pp. 707–710, 2001.
- [26] S. Y. Cho, C. G. Park, and G. I. Jee, “Measurement system of walking distance using low-cost accelerometers,” in *proceedings of the 4th Asian Control Conference*, 2002.
- [27] V. Gabaglio, “Centralised kalman filter for augmented gps,” in *ION 2001 Conference Proceedings, Salt Lake City, Utah, USA*, no. CONF, 2001.
- [28] K. Aminian, P. Robert, E. Jequier, and Y. Schutz, “Level, downhill and uphill walking identification using neural networks,” *Electronics Letters*, vol. 29, no. 17, pp. 1563–1565, 1993.

- [29] S. Cho, “Design of a pedestrian navigation system and the error compensation using rhekf filter,” in *Department of Control and Instrumentation Engineering*. Kwangwoon Univ., 2004.
- [30] S. Y. Cho and C. G. Park, “A calibration technique for a two-axis magnetic compass in telematics devices,” *ETRI journal*, vol. 27, no. 3, pp. 280–288, 2005.
- [31] M. J. Caruso and L. S. Withanawasam, “Vehicle detection and compass applications using amr magnetic sensors,” in *Sensors Expo Proceedings*, vol. 477, 1999, p. 39.
- [32] C. E. White, D. Bernstein, and A. L. Kornhauser, “Some map matching algorithms for personal navigation assistants,” *Transportation research part c: emerging technologies*, vol. 8, no. 1-6, pp. 91–108, 2000.
- [33] M. A. Quddus, W. Y. Ochieng, L. Zhao, and R. B. Noland, “A general map matching algorithm for transport telematics applications,” *GPS solutions*, vol. 7, no. 3, pp. 157–167, 2003.
- [34] T. Judd, “A personal dead reckoning module,” in *ION GPS*, vol. 97, 1997, pp. 1–5.
- [35] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara, “A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu,” in *2009 IEEE International Symposium on Intelligent Signal Processing*, 2009, pp. 37–42.
- [36] E. Foxlin, “Pedestrian tracking with shoe-mounted inertial sensors,” *IEEE Computer graphics and applications*, no. 6, pp. 38–46, 2005.

- [37] H. Ju and C. G. Park, “A pedestrian dead reckoning system using a foot kinematic constraint and shoe modeling for various motions,” *Sensors and Actuators A: Physical*, vol. 284, pp. 135–144, 2018.
- [38] U. Steinhoff and B. Schiele, “Dead reckoning from the pocket—an experimental study,” in *2010 IEEE international conference on pervasive computing and communications (PerCom)*, 2010, pp. 162–170.
- [39] J.-S. Lee and S.-M. Huang, “An experimental heuristic approach to multi-pose pedestrian dead reckoning without using magnetometers for indoor localization,” *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9532–9542, 2019.
- [40] Q. Tian, Z. Salcic, I. Kevin, K. Wang, and Y. Pan, “A multi-mode dead reckoning system for pedestrian tracking using smartphones,” *IEEE Sensors Journal*, vol. 16, no. 7, pp. 2079–2093, 2015.
- [41] B. Wang, X. Liu, B. Yu, R. Jia, and X. Gan, “Pedestrian dead reckoning based on motion mode recognition using a smartphone,” *Sensors*, vol. 18, no. 6, p. 1811, 2018.
- [42] Z.-A. Deng, G. Wang, Y. Hu, and D. Wu, “Heading estimation for indoor pedestrian navigation using a smartphone in the pocket,” *Sensors*, vol. 15, no. 9, pp. 21 518–21 536, 2015.
- [43] Z. Deng, X. Liu, Z. Qu, C. Hou, and W. Si, “Robust heading estimation for indoor pedestrian navigation using unconstrained smartphones,” *Wireless Communications and Mobile Computing*, vol. 2018, 2018.
- [44] I. Skog, P. Handel, J.-O. Nilsson, and J. Rantakokko, “Zero-velocity detection—an algorithm evaluation,” *IEEE transactions on biomedical engineering*, vol. 57, no. 11, pp. 2657–2666, 2010.

- [45] S. Y. Cho and C. G. Park, “Mems based pedestrian navigation system,” *The Journal of Navigation*, vol. 59, no. 1, pp. 135–153, 2006.
- [46] C. Combettes and V. Renaudin, “Comparison of misalignment estimation techniques between handheld device and walking directions,” in *2015 IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–8.
- [47] P. Davidson and R. Piché, “A survey of selected indoor positioning methods for smartphones,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1347–1370, 2016.
- [48] S. Shin, C. Park, J. Kim, H. Hong, and J. Lee, “Adaptive step length estimation algorithm using low-cost mems inertial sensors,” in *2007 IEEE sensors applications symposium*, 2007, pp. 1–5.
- [49] J. C. Alvarez, D. Alvarez, A. López, and R. C. González, “Pedestrian navigation based on a waist-worn inertial sensor,” *Sensors*, vol. 12, no. 8, pp. 10 536–10 549, 2012.
- [50] K.-C. Lan and W.-Y. Shih, “Using simple harmonic motion to estimate walking distance for waist-mounted pdr,” in *2012 IEEE wireless communications and networking conference (WCNC)*, 2012, pp. 2445–2450.
- [51] A. Mikov, A. Moschevikin, A. Fedorov, and A. Sikora, “A localization system using inertial measurement units from wireless commercial hand-held devices,” in *2013 IEEE International Conference on Indoor Positioning and Indoor Navigation*, 2013, pp. 1–7.

- [52] J. Liu, R. Chen, L. Pei, R. Guinness, and H. Kuusniemi, “A hybrid smart-phone indoor positioning solution for mobile lbs,” *Sensors*, vol. 12, no. 12, pp. 17 208–17 233, 2012.
- [53] N. Kothari, B. Kannan, E. D. Glasgown, and M. B. Dias, “Robust indoor localization on a commercial smart phone,” *Procedia computer science*, vol. 10, pp. 1114–1120, 2012.
- [54] S. Shin, M. Lee, C. Park, H. Hong, and J. Lee, “Phone location classification based on mems inertial sensors,” in *Proceedings of 17th GNSS Workshop, Jeju Haevichi Hotel, Jeju, South Korea*, 2010.
- [55] S. Saeedi, A. Moussa, and N. El-Sheimy, “Context-aware personal navigation using embedded sensor fusion in smartphones,” *Sensors*, vol. 14, no. 4, pp. 5742–5767, 2014.
- [56] A. Kharb, V. Saini, Y. Jain, and S. Dhiman, “A review of gait cycle and its parameters,” *IJCEM International Journal of Computational Engineering & Management*, vol. 13, pp. 78–83, 2011.
- [57] S. K. Park and Y. S. Suh, “A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems,” *Sensors*, vol. 10, no. 10, pp. 9163–9178, 2010.
- [58] M. Susi, V. Renaudin, and G. Lachapelle, “Motion mode recognition and step detection algorithms for mobile phone users,” *Sensors*, vol. 13, no. 2, pp. 1539–1562, 2013.
- [59] A. Abadleh, E. Al-Hawari, E. Alkafaween, and H. Al-Sawalqah, “Step detection algorithm for accurate distance estimation using dynamic step

- length,” in *2017 18th IEEE International Conference on Mobile Data Management (MDM)*, 2017, pp. 324–327.
- [60] S. Y. Park, S. J. Heo, and C. G. Park, “Accelerometer-based smartphone step detection using machine learning technique,” in *2017 IEEE International Electrical Engineering Congress (iEECON)*, 2017, pp. 1–4.
- [61] J. Seo, Y. Chiang, T. H. Laine, and A. M. Khan, “Step counting on smartphones using advanced zero-crossing and linear regression,” in *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, 2015, pp. 1–7.
- [62] M.-S. Pan and H.-W. Lin, “A step counting algorithm for smartphone users: Design and implementation,” *IEEE Sensors Journal*, vol. 15, no. 4, pp. 2296–2305, 2014.
- [63] R. Zhou, “Pedestrian dead reckoning on smartphones with varying walking speed,” in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.
- [64] X. Kang, B. Huang, and G. Qi, “A novel walking detection and step counting algorithm using unconstrained smartphones,” *Sensors*, vol. 18, no. 1, p. 297, 2018.
- [65] A. C. Dİrican and S. Aksoy, “Step counting using smartphone accelerometer and fast fourier transform,” *Sigma J. Eng. Nat. Sci*, vol. 8, pp. 175–182, 2017.
- [66] A. Martinelli, H. Gao, P. D. Groves, and S. Morosi, “Probabilistic context-aware step length estimation for pedestrian dead reckoning,” *IEEE Sensors Journal*, vol. 18, no. 4, pp. 1600–1611, 2017.

- [67] Z. Tang, Y. Guo, and X. Chen, “Self-adaptive step counting on smart-phones under unrestricted stepping modes,” in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, 2016, pp. 788–797.
- [68] A. Brajdic and R. Harle, “Walk detection and step counting on unconstrained smartphones,” in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013, pp. 225–234.
- [69] V. Renaudin, M. Susi, and G. Lachapelle, “Step length estimation using handheld inertial sensors,” *Sensors*, vol. 12, no. 7, pp. 8507–8525, 2012.
- [70] J. Borenstein, “Pdr system for firefighters,” in *5th Annual Technology Workshop, Worcester Polytechnic Institute*, 2010.
- [71] L. Ojeda and J. Borenstein, “Non-gps navigation for security personnel and first responders,” *The Journal of Navigation*, vol. 60, no. 3, pp. 391–407, 2007.
- [72] A. R. Jiménez, F. Seco, J. C. Prieto, and J. Guevara, “Indoor pedestrian navigation using an ins/ekf framework for yaw drift reduction and a foot-mounted imu,” in *2010 IEEE 7th Workshop on Positioning, Navigation and Communication*, 2010, pp. 135–143.
- [73] M. S. Lee, C. Park, and C. W. Shim, “A movement-classification algorithm for pedestrian using foot-mounted imu,” in *Proceedings of the 2012 International Technical Meeting of The Institute of Navigation*, 2012, p. 922.

- [74] L. E. Díez, A. Bahillo, J. Otegui, and T. Otim, “Step length estimation methods based on inertial sensors: A review,” *IEEE Sensors Journal*, vol. 18, no. 17, pp. 6908–6926, 2018.
- [75] A. Köse, A. Cereatti, and U. Della Croce, “Bilateral step length estimation using a single inertial measurement unit attached to the pelvis,” *Journal of neuroengineering and rehabilitation*, vol. 9, no. 1, p. 9, 2012.
- [76] E. Bishop and Q. Li, “Walking speed estimation using shank-mounted accelerometers,” in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 5096–5101.
- [77] S. Yang and Q. Li, “Ambulatory walking speed estimation under different step lengths and frequencies,” in *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2010, pp. 658–663.
- [78] Z. Sun, X. Mao, W. Tian, and X. Zhang, “Activity classification and dead reckoning for pedestrian navigation with wearable sensors,” *Measurement science and technology*, vol. 20, no. 1, p. 015203, 2008.
- [79] D. Gusenbauer, C. Isert, and J. Krösche, “Self-contained indoor positioning on off-the-shelf mobile devices,” in *2010 IEEE International Conference on Indoor Positioning and Indoor Navigation*, 2010, pp. 1–9.
- [80] S. Yang, H. Zhu, G. Xue, and M. Li, “Disen: Ranging indoor casual walks with smartphones,” in *2015 11th IEEE International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, 2015, pp. 178–185.
- [81] S. Zihajehzadeh and E. J. Park, “A gaussian process regression model for walking speed estimation using a head-worn imu,” in *2017 39th An-*

- nual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017, pp. 2345–2348.
- [82] Y. Song, S. Shin, S. Kim, D. Lee, and K. H. Lee, “Speed estimation from a tri-axial accelerometer using neural networks,” in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2007, pp. 3224–3227.
 - [83] O. J. Woodman, “An introduction to inertial navigation,” University of Cambridge, Computer Laboratory, Tech. Rep., 2007.
 - [84] O. Woodman and R. Harle, “Pedestrian localisation for indoor environments,” in *Proceedings of the 10th international conference on Ubiquitous computing*, 2008, pp. 114–123.
 - [85] G. Schmidt, “Ins/gps technology trends, advances in navigation sensors and integration technology,” *NATO RTO Lecture series*, vol. 232, 2004.
 - [86] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhelou, and Y. Amirat, “Physical human activity recognition using wearable sensors,” *Sensors*, vol. 15, no. 12, pp. 31 314–31 338, 2015.
 - [87] W. Bouaguel, E. Mouelhi, and G. B. Mufti, “New method for instance feature selection using redundant features for biological data,” in *International Conference on Mathematical Aspects of Computer and Information Sciences*, 2015, pp. 398–405.
 - [88] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” *IEEE Transactions on knowledge and data engineering*, vol. 17, no. 4, pp. 491–502, 2005.

- [89] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [90] “k nearest neighbor classifier (knn)-machine learning algorithms,” <https://medium.com/@equipintelligence/k-nearest-neighbor-classifier-knn-machine-learning-algorithms-ed62feb86582>, accessed: 2020-05-15.
- [91] “Introduction to support vector machines,” https://docs.opencv.org/3.4/d1/d73/tutorial_introduction_to_svm.html, accessed: 2020-05-15.
- [92] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.
- [93] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, “A practical guide to support vector classification,” 2003.
- [94] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.
- [95] D. Titterton, J. L. Weston, and J. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.
- [96] C. W. Kang, H. J. Kim, and C. G. Park, “A human motion tracking algorithm using adaptive ekf based on markov chain,” *IEEE Sensors Journal*, vol. 16, no. 24, pp. 8953–8962, 2016.
- [97] Y. S. Suh, “Orientation estimation using a quaternion-based indirect kalman filter with adaptive estimation of external acceleration,” *IEEE*

- Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp. 3296–3305, 2010.
- [98] A. M. Sabatini, “Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing,” *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.
 - [99] C. W. Kang and C. G. Park, “Attitude estimation with accelerometers and gyros using fuzzy tuned kalman filter,” in *2009 IEEE European Control Conference (ECC)*, 2009, pp. 3713–3718.
 - [100] A. M. Sabatini, “Variable-state-dimension kalman-based filter for orientation determination using inertial and magnetic sensors,” *Sensors*, vol. 12, no. 7, pp. 8491–8506, 2012.
 - [101] A. M. Sabatini, “Estimating three-dimensional orientation of human body parts by inertial/magnetic sensing,” *Sensors*, vol. 11, no. 2, pp. 1489–1525, 2011.
 - [102] J. K. Lee, E. J. Park, and S. N. Robinovitch, “Estimation of attitude and external acceleration using inertial sensor measurement during various dynamic conditions,” *IEEE transactions on instrumentation and measurement*, vol. 61, no. 8, pp. 2262–2273, 2012.
 - [103] M.-J. Yu, “Ins/gps integration system using adaptive filter for estimating measurement noise variance,” *IEEE transactions on aerospace and electronic systems*, vol. 48, no. 2, pp. 1786–1792, 2012.
 - [104] W. Li and J. Wang, “Effective adaptive kalman filter for mems-imu/magnetometers integrated attitude and heading reference systems,” *The Journal of Navigation*, vol. 66, no. 1, pp. 99–113, 2013.

- [105] M. Ghobadi, P. Singla, and E. T. Esfahani, “Robust attitude estimation from uncertain observations of inertial sensors using covariance inflated multiplicative extended kalman filter,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 1, pp. 209–217, 2017.
- [106] X. Tong, Z. Li, G. Han, N. Liu, Y. Su, J. Ning, and F. Yang, “Adaptive ekf based on hmm recognizer for attitude estimation using mems marg sensors,” *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3299–3310, 2017.
- [107] A. C. Chiella, B. O. Teixeira, and G. A. Pereira, “Quaternion-based robust attitude estimation using an adaptive unscented kalman filter,” *Sensors*, vol. 19, no. 10, p. 2372, 2019.
- [108] C. A. CB, T. B. OS, and P. G. AS, “Robust attitude estimation using an adaptive unscented kalman filter,” in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7748–7754.
- [109] X. Tong, Y. Su, Z. Li, C. Si, G. Han, J. Ning, and F. Yang, “A double-step unscented kalman filter and hmm-based zero-velocity update for pedestrian dead reckoning using mems sensors,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 581–591, 2019.
- [110] L. Chang, B. Hu, G. Chang, and A. Li, “Huber-based novel robust unscented kalman filter,” *IET Science, Measurement & Technology*, vol. 6, no. 6, pp. 502–509, 2012.
- [111] C. D. Karlgaard, “Nonlinear regression huber–kalman filtering and fixed-interval smoothing,” *Journal of guidance, control, and dynamics*, vol. 38, no. 2, pp. 322–330, 2015.

- [112] X. Wang, N. Cui, and J. Guo, “Huber-based unscented filtering and its application to vision-based relative navigation,” *IET radar, sonar & navigation*, vol. 4, no. 1, pp. 134–141, 2010.
- [113] Z. Qiu, Y. Huang, and H. Qian, “Adaptive robust nonlinear filtering for spacecraft attitude estimation based on additive quaternion,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 1, pp. 100–108, 2019.
- [114] C. D. Karlgaard and H. Schaub, “Huber-based divided difference filtering,” *Journal of guidance, control, and dynamics*, vol. 30, no. 3, pp. 885–891, 2007.
- [115] R. Mehra, “Approaches to adaptive filtering,” *IEEE Transactions on automatic control*, vol. 17, no. 5, pp. 693–698, 1972.
- [116] E. A. Yildirim, “On the minimum volume covering ellipsoid of ellipsoids,” *SIAM Journal on Optimization*, vol. 17, no. 3, pp. 621–641, 2006.
- [117] O. Bocharadt and J. Uhlmann, “On the equivalence of the general covariance union (gcu) and minimum enclosing ellipsoid (mee) problems,” *arXiv preprint arXiv:1012.4795*, 2010.
- [118] P. Kumar and E. A. Yildirim, “Minimum-volume enclosing ellipsoids and core sets,” *Journal of Optimization Theory and Applications*, vol. 126, no. 1, pp. 1–21, 2005.
- [119] S. B. Pope, “Algorithms for ellipsoids,” *Cornell University Report No. FDA*, pp. 08–01, 2008.

- [120] D. Roetenberg, H. Luinge, and P. Slycke, “Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors,” *Xsens Motion Technologies BV, Tech. Rep*, vol. 1, 2009.
- [121] “Acutronic, one-axis series ac1120s-tilt,” <http://www.acutronic.com>, accessed: 2020-01-17.
- [122] “Vicon, vicon vantage,” <https://www.vicon.com>, accessed: 2020-01-17.
- [123] S. Y. Park, H. Ju, and C. G. Park, “Stance phase detection of multiple actions for military drill using foot-mounted imu,” *sensors*, vol. 14, p. 16, 2016.
- [124] M. Paulich, M. Schepers, N. Rudigkeit, and G. Bellusci, “Xsens mtw awinda: Miniature wireless inertial-magnetic motion tracker for highly accurate 3d kinematic applications,” *Xsens: Enschede, The Netherlands*, 2018.
- [125] J. W. Song and C. G. Park, “Enhanced pedestrian navigation based on course angle error estimation using cascaded kalman filters,” *Sensors*, vol. 18, no. 4, p. 1281, 2018.
- [126] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [127] U. Blanke and B. Schiele, “Sensing location in the pocket,” *Ubicomp Poster Session*, p. 2, 2008.
- [128] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole, “Which way am i facing: Inferring horizontal device orientation from an accelerometer signal,” in *2009 IEEE International Symposium on Wearable Computers*, 2009, pp. 149–150.

- [129] M. Kourogi and T. Kurata, “Personal positioning based on walking locomotion analysis with self-contained sensors and a wearable camera,” in *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2003, p. 103.
- [130] D. Goshen-Meskin and I. Bar-Itzhack, “Observability analysis of piecewise constant systems. ii. application to inertial navigation in-flight alignment (military applications),” *IEEE Transactions on Aerospace and Electronic systems*, vol. 28, no. 4, pp. 1068–1075, 1992.

국문초록

본 논문에서는 저가형 관성센서를 이용한 보행항법시스템 (PDR: Pedestrian Dead Reckoning)의 성능 향상 알고리즘을 제안한다. 구체적으로 보행자가 실내에서 스마트폰을 들고 이동할 때 발생하는 다양한 동작 상황에서도 운용될 수 있도록, 매개변수 기반 측정치를 사용하는 적분 기반의 보행자 항법 알고리즘을 구성한다. 또한 타원체 기반 자세 추정 알고리즘을 구성하여 외란 상황에서도 강인하게 자세를 추정하는 알고리즘을 제안한다. 추가적으로 기계학습 기반의 동작 인식 정보를 이용, 동작에 따른 측정치 업데이트를 달리함으로써 위치 추정 성능을 향상시킨다.

먼저 스마트폰 기기의 이동 방향을 정확하게 추정하기 위해 타원체 기법 기반 적응 자세 추정을 제안한다. 자세 추정 기법 (AHRS: Attitude and Heading Reference System)은 자이로를 기반으로 자세를 계산하고 자이로 센서오차에 의해 발생하는 드리프트를 보정하기 위해 측정치로 가속도계와 지자계를 사용한다. 일반적으로 가속 및 지자계 외란 상황에서는 자세 추정 성능이 떨어지는데, 추정 성능을 효과적으로 향상시키기 위해 본 논문에서는 타원체 기반 적응 자세 추정 기법을 제안한다. 측정치 외란이 들어오는 경우, 외란의 방향을 고려하여 타원체 기법으로 측정치 공분산을 조정해줌으로써 방향을 고려하지 않은 적응 추정 기법보다 정확하게 측정치 업데이트를 할 수 있다. 특히 외란이 한 축으로만 들어오는 경우, 제안한 알고리즘은 방향을 고려해 나머지 두 축에 대해서는 업데이트 해줌으로써 측정치를 부분적으로 사용할 수 있다. 레이트 테이블, 모션 캡처 장비를 통해 제안한 알고리즘의 자세 성능이 향상됨을 확인하였다.

다음으로 다양한 동작에서도 운용 가능한 적분 및 매개변수 기법을 융합하는 보행항법 알고리즘을 제안한다. 스마트폰을 이용해 실내를 이동할 때에는 자유도가 크기 때문에 전화 걸기, 문자, 바지 주머니 넣기 등 다양한 동작이 발생 가능하다. 기존의 스마트폰 기반 보행 항법에서는 매개변수 기법을 기반으로 위치를 추정하

는데, 이는 보행자의 진행 방향과 기기의 방향이 일치하는 경우에만 사용 가능하며 일치하지 않는 경우 자세 오차로 인한 위치 오차가 크게 발생한다. 이러한 문제를 해결하기 위해 본 논문에서는 적분 기반 기법을 기반으로 상태변수를 구성하고 매 개변수 기법을 통해 나오는 위치 벡터를 측정치로 사용하는 알고리즘을 제안한다. 만약 기계학습을 통해 인식한 동작을 바탕으로 진행 방향과 기기 방향이 일치하지 않는 경우, 주성분 분석을 통해 계산한 진행방향을 이용해 진행 방향을, 매개변수 기법을 통해 얻은 보폭으로 거리를 업데이트해 줌으로써 보행 중 발생하는 여러 동작에서도 강인하게 운용할 수 있다.

다양한 동작 상황 및 경로를 고려한 실험을 통해 위에서 제안한 방법이 다양한 실내 환경에서도 안정적으로 위치를 추정하고 성능이 향상됨을 확인하였다.

주요어: 실내 항법, PDR, 확장 칼만 필터, 타원체 기법 기반 적응 자세 추정, 적분 및 매개변수 기법 융합 방법

학번: 2013-20674