NTNU – Trondheim
Norwegian University of
Science and Technology

Human Tracking System

Hedvik Louise Hansen

June 15, 2014

Problem Description

Motivation

Statoil has a large number of industrial facilities onshore and offshore. Operator health and safety is one of the company's key performance indices. The need for a system for localizing people that are operating inside a production facility are due to the following challenges:

- In case of emergency, one need to know the whereabouts of people to be able to effectively assess the risk situation and to carry out evacuation
- The operator exposure to health stress factors, such as noise and poisonous substances, should be monitored to avoid short and long term damages

A small and lightweight localization unit that communicates with a wireless network can be a key element in meeting these challenges. A suggestion for

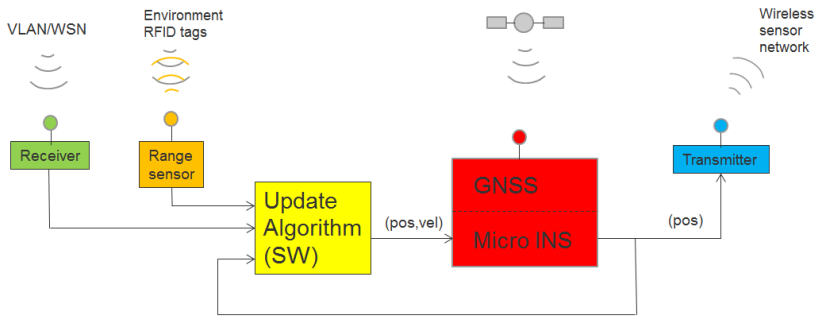


Figure 1: WiLoc localization system

such a system was given by Statoil, as illustrated in Figure 1, in addition to the following list of wanted goals:

1. Develop and demonstrate a prototype system:
   (a) Practical implementation of a system that can operate indoors as well as outdoors
   (b) Documented in a report with theory and discussion on technology choices
2. Investigate and document the feasibility of different sensor configurations:
   (a) INS and GNSS (reference case, outdoor)
   (b) INS with a number of RFID and/or WSN nodes
   (c) INS with a pedometer algorithm
   (d) Combining all of b and c.
3. The system should be mobile and communicate via a wireless network

Goals

The candidate should:

- Look into necessary background theory
- Look into previous work
- Suggest design and implementation strategies
- As far as time permits, implement and evaluate solution

Summary

This thesis is motivated by Statoil's wish to localize people operating inside an onshore or offshore production facility. There are especially two benefits of such systems. By monitoring an operator's exposure to noise, vibration and harmful gases, short and long term damages can be avoided. In addition, a localization system would provide employees with a higher degree of safety in case of emergency.

The aim of this thesis is to develop a prototype system for localizing people indoors as well as outdoors with use of an inertial measurement unit. As a consequence, challenges such as drift and signal noise, as well as magnetic interference are directed. Since the IMU is subject to drift, investigations on different sensor configurations must be conducted.

Collected sensor data from the hip-mounted IMU are used for step detection, step length estimation and heading determination. By combining the results from these three approaches, a positioning estimate for the human operator is calculated using dead reckoning. Dead reckoning is a positioning method for determining an objects location based on a former position, velocity and direction.

Experiments indicated satisfying performance by both the implemented step detection algorithm and the step length estimation model. However, the heading determination is subject to magnetic interferences when the sensor is applied in an industrial area. In an indoors environment are position error 2-4 meters after 22 meters walking. Performance in an outdoors environment are significantly better, with a position error of 10 meter after 400 meters.

Integrating the suggested solution with another positioning system will increase system performance indoors and thus fulfill the aim of this thesis.

Sammendrag

Den overordnende bakgrunnen for denne hovedoppgaven er Statoils ønske om å lokalisere mennesker som jobber på et produksjonsanlegg, på land eller offshore. Et slikt system vil kunne overvåke en operatørs eksponering for støy, vibrasjoner og skadelige gasser, og man kan dermed unngå kortsiktig og langsiktige skader. I tillegg vil lokaliseringssystemet gi de ansatte en høyere grad av sikkerhet hvis det oppstår behov for evakuering.

Målet med oppgaven er på implementer et prototype system for lokalisering av folk, både innendørs og utendørs ved bruk av en sensor kalt IMU. Sensoren gir utfordringer både med tanke signalstøy, drift og magnetisk interferens. Siden posisjoneringsmetoder basert på en IMU opplever drift må det undersøkes om andre metoder for innendørs posisjonering kan kombineres med sensoren.

Sensoren er festet på hoften til brukeren, og samler data som brukes til å detektere skritt, estimere skrittlengde og bestemme retning på skrittet. Ved å kombinere resultatene fra hver og en av disse metodene kan man gi et estimat på operatørens nåværende posisjoon ved bruk av en metode kalt dd-ead reckoning". Denne metodene estimerer et objekts posisjon ved bruk av en tidligere posisjon, i tillegg til retningen og farten brukeren har.

De utførte eksperimentene viser gode resultater for både skrittdetektsjon og skrittlengde estimering. Dessverre er bestemmelsen av retning påvirket av forstyrrelser som oppstår i magnetfeltet innendørs. Tester innendørs ga en posisjonsfeil på 2-4 meters. Utendørs var ytelsen bedre, med en posisjonsfeil på 10 meter etter 400 meter gange.

Ved å integrere den foreslåtte løsningen med en annen posisjoneringssystem, vil systemytelsen øke innendørs, dermed oppfylle målet med denne oppgaven.

Contents

List of Figures

List of Abbreviations

AGPS Assisted Global Positioning System

AHRS Attitude and Heading Reference System

DR Dead Reckoning

ECEF Earth Centered Earth Fixed

ECI Earth Centered Inertial

GLONASS Global Orbiting Navigation Satellite System

GNSS Global Navigation Satellite System

GPS Global Positioning System

HSGPS High Sensitivity Global Positioning System

INS Inertial Navigation System

IPS Indoor Positioning System

NED North East Down

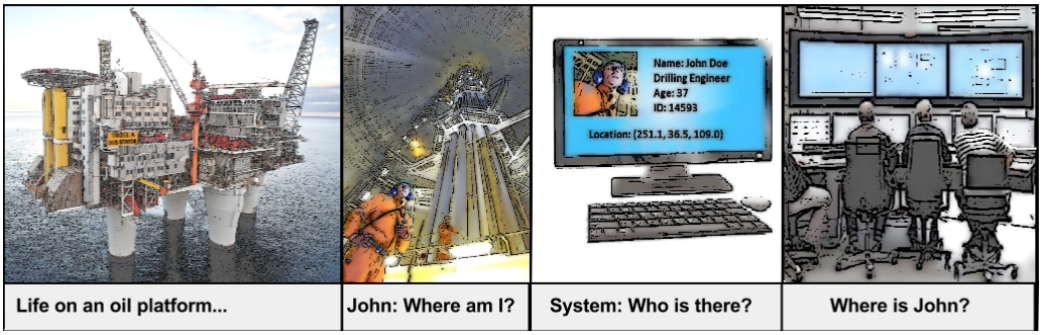
PDR Pedestrian Dead Reckoning

RFID Radio Frequency Identification

WLAN Wireless Local Area Network

Chapter 1

Introduction



1.1 Background

Knowing where you are, how fast you are moving and in which direction is the art of navigation, and has been of interest for thousands of years. The simplest form of navigation is based on pure observations. The ancient Polynesians are known for such navigation skills when making voyages on open sea in search for inhabited islands. They remembered the motion of stars, looked for wildlife species and observed the direction, speed and size of waves and the colors of the sea and sky. Sailors with more land in sight than the Polynesians used landmarks such as mountains, rivers and beaches when navigating from harbor to harbor. Introducing scientific methods and instruments made navigation better and safer. Angular measurements between the horizon and celestial bodies made it possible to decide the position

in north-south direction, and an accurate time-keeping device further determined the east-west position. Today we still use the ancient methods of charts, compass, sextants and celestial bodies, but they are often combined with modern techniques such as radio, radar and satellite navigation.

Knowing your own location is not always sufficient. The first example dates back to World War 2 where radar was used to find enemy's aircrafts, fleets, artillery and even buried mines. Apollo 11 used a video camera to broadcast the location of Buzz Aldrin and Neil Armstrong on live TV. Examples from present time include tracking mail orders, registration of cars passing a toll booth and aircrafts seeking people with the use of thermographic camera. As sensors become smaller and cheaper, software is getting more intelligent, processing is faster, and network communication is available world-wide, sensors sharing information about location and movements of an object is part of everyday-life.

The possibilities for localization and tracking are many, but challenges arise when an object is moving from outdoors to indoors. No mountain can be seen in the distance, nor a clear view to the sky. Concrete and steel beams give a harsh environment for radio transmission, and signals can easily be corrupted or completely blocked. Being virtually immune to external disturbance, an inertial navigation system (INS) is a possible solution to indoor navigation. INS relies on inertial acceleration and angular velocity when calculating an object's position and rotation. Thus, INS are not dependent on external references, but use Newton's laws of classic mechanics.

This thesis is part of a project for Statoil referred to as WiLoC. WiLoc is a localization system providing real time estimates of the localization of a moving target, such as a human operator. Statoil is a leading energy company in oil and gas production, and has a large number of industrial facilities onshore and offshore. In order to provide their employees with the highest degree of safety there is a need to know the localization of people operating inside a production facility. Such a system will in case of emergency help determine the risk situation, and effectively carry out evacuation. A challenge of more mundane nature is the danger of noise, vibration and harmful gases. By knowing the location of a human operator one can monitor an operator's exposure to such health stress factors and thereby avoid short and long term damages.

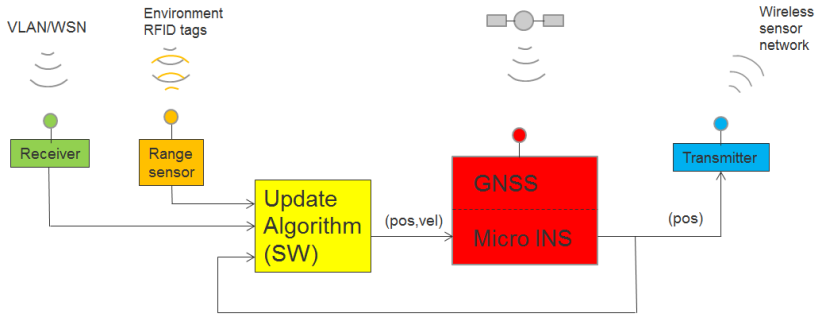


Figure 1.1: WiLoc localization system

1.2 System Overview

Figure 1.1 illustrates the possible outline of a portable sensor system for accurate position localization given in the assignment text, and will henceforth be referred to as a WiLoc node.

The red block in Figure 1.1 symbolizes an off-the-shelf IMU, Spatial. Spatial combines the ordinary features of an IMU, such as acceleration measurement and attitude rates, with an advanced GNSS receiver. The IMU will for convenience be placed around the operator's hip, collecting sensor data for the purpose of indoor navigation. The collected data are sent to an update algorithm, marked in yellow in Figure 1.1. The update algorithm use the data from the IMU's sensors to calculate a real time position estimate. The green and orange block symbolize sources that can provide the update algorithm with reliable position updates, in order to cancel drift errors caused by the IMU. Such external correction signals should not be expected regularly, though a frequent update is beneficial. Relevant update sources can be WLAN, WSN or RFID, velocity sensors, such as odometer and pedometers, or any other update function that gives map-like information feedback. The calculated position of the operator will be transmitted via WSN to an overall system, symbolized by the blue block in Figure 1.1.

WiLoc localization system has the following requirements:

- Position error less then 2 m (good) up to 5 m (ok)
- ATEX zone 0 certified

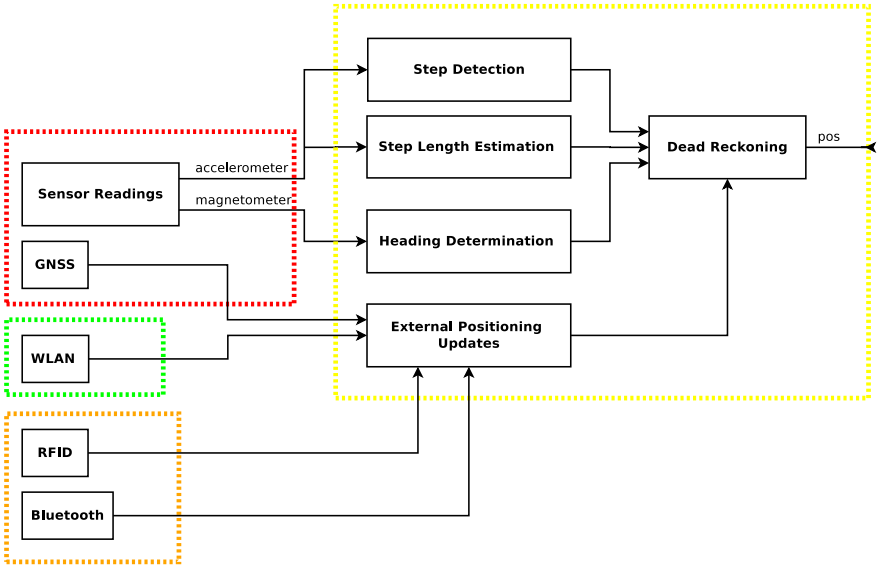


Figure 1.2: Suggested solution to the WiLoc localization system, where the colored frames correspond to Figure 1.1

This thesis focus on the design and implementation of the red and yellow block of the WiLoc localization system, and will therefore solely rely on measurements from the IMU for the purpose of localizing the human operator. However, a selection of indoor positioning methods are introduced and discussed by means of providing the suggested INS with external positioning updates. The blue block is completely excluded.

The suggested solution can be viewed in Figure 1.2. Collected sensor data from the hip-mounted IMU are used for step detection, step length estimation and heading determination. By combining the results from the three approaches, a positioning estimate for the human operator is calculated using dead reckoning (DR). DR is a positioning method to determine an objects location based on a former position, velocity and direction. Since possible external positioning updates are discussed, these positioning technologies are added in the figure.

1.3 Thesis Outline

The organization of this thesis represents the suggested solution to the WiLoc localization system. Though the demands of the system are given will this thesis provide theory for the purpose of giving the reader an opportunity to evaluated the solution, but also to gain a more deeper understanding of the subject. Evaluations and discussions will be given along the way, but the thesis also include an overall conclusion.

This introduction is followed by a brief overview of selected methods for indoor positioning, along with their advantages and disadvantages for providing the INS with positioning updates. Chapter 3 goes in-depth in the theory of inertial navigation systems. Furthermore are chapters 4-6 dedicated to the theory behind the implemented update algorithm. Chapter 4 explains the theory behind step detection and gives an overview on two popular algorithms, Pan-Tompkins and Libby peak. Chapter 5 look into step length estimation, while Chapter 6 discuss heading determination. In chapter 7 is the suggested solution and implementation of this thesis presented. Experiments and achieved results are presented in Chapter 9, where discussions on the individual parts of the system are added. Discussion on overall system performance are given in Chapter 9, along with concluding remarks on the presented work. Furthermore, proposals for future work are also made in Chapter 9.

Chapter 2

Indoor Positioning

The topic of this thesis is a system for localizing people that are operating inside a production facility. The main task is to determine the position of a human operator wearing a WiLoc node with use of an indoor positioning method based on inertial sensors. Inertial sensors and the art of inertial navigation systems (INS) are thoroughly elaborated throughout this thesis, but a brief introduction is given in this chapter, in addition to related work in the field of indoor positioning systems (IPS) using inertial sensors.

To provide an overview of the subject of indoor positioning, a short introduction is given in the first section of this chapter, followed by brief descriptions and, when possible, comparisons of a selection of indoor positioning technologies. Each technology is evaluated for the purpose of providing the INS with positioning updates. The selection was made based on the assignment text, which focus on Global Positioning Systems (GPS), Wireless Local Area Network (WLAN), and Radio Frequency Identification (RFID). Bluetooth was in reviewed literature found to be a relevant technology for indoor positioning, and is therefore also included. Though this thesis could not examine all IPS, Table 2.1 provides a high-level overview of existing technologies.

The motivation for this chapter is threefold:

- Introduction to Indoor Positioning
- Description of selected methods for indoor positioning and related work
- Discussion on the subject of feasible integration solutions

2.1 Introduction to Indoor Positioning

Indoor positioning is a technique that provides location dependent information about objects or people inside buildings. In some aspect, indoor positioning is simpler than outdoor positioning. For instance, the geographical area to be covered is much smaller, and expected speeds of objects are lower. Nonetheless, other aspects make indoor positioning much more challenging. Radio transmission signals can easily get corrupted or completely blocked in an indoor environment. Consequently, popular positioning systems for outdoor use, such as GPS, can not be expected to work properly indoors. In addition, positioning estimates must be more accurate indoors since the topology constraints often are more complex indoors than outdoors. For example, an vertical positioning uncertainty of several meters does not pose much problem for a navigation system in a car moving at 60 mph. For an IPS will the same error translates into uncertainty regarding floor position. Thus, there is a need for systems designed explicitly to coop with the challenges of indoor positioning.

The list of applications for indoor positioning is astonishing long. Examples include the ability to navigate in public buildings and offices, provide shopping customers with location based advertisement, detecting lost items in private homes, emergency or fall detection on elderly people living at home, location tracking of medical personnel in a hospital, locating firemen in a burning building, industrial robots and intelligent transportation. In 2012 presented [1] a four-page long survey on indoor positioning applications, but with the evolving development new applications can surely already be added.

As mentioned, there exists no universal solution to the field of indoor positioning. Therefore is it important to gain an overview of existing technology's capabilities and match them with the requirements of the system at hand. Such specifications are accuracy, coverage, integrity, precision, availability, update rate, latency, costs, infrastructure, privacy, robustness etc. In the remainder of this chapter, different technologies for indoor positioning are viewed and discussed based on such requirements in order to highlight the advantages and disadvantages of the selected technologies. Two parameters are of particular importance: coverage and accuracy. A graphical overview of indoor technologies in dependence of accuracy and coverage was illustrated in [1], and can be viewed in Figure 2.1. It can be seen that INS are within the

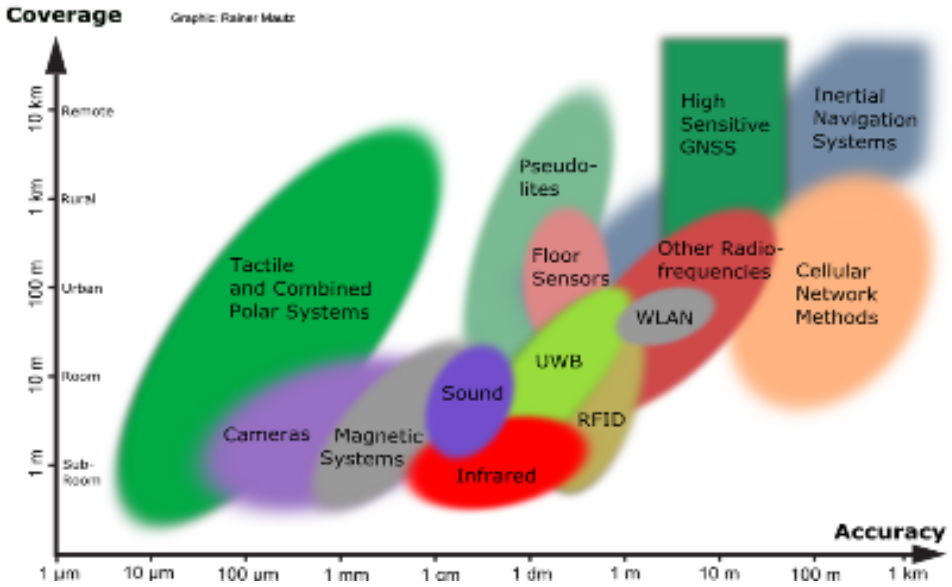


Figure 2.1: Overview of indoor technologies in dependence on accuracy and coverage [1]

range of 1 m to 1 km accuracy, and with 10 m to 10 km coverage. High Sensitive GNSS, WLAN, RFID and other radio-frequencies are evaluated in this chapter, while remaining technologies will not be considered in this thesis.

Indoor positioning technologies use different methods for obtaining indoor navigation, such as proximity, lateration, angulation, dead reckoning, location fingerprinting and pattern recognition. Some of these positioning methods will be briefly explained later in this chapter. However, an in-depth description of such indoor positioning methods expands the subject of this thesis, but further research can be conducted in papers such as [12].

2.2 Indoor Positioning Technologies

This section starts with a brief introduction to inertial sensors and INS, in addition to reviewed literature related to the suggested solution for the WiLoc localization system. Following "Inertial sensors and INS", are a selection of possible external positioning updates for INS. As mentioned in the preceding of this chapter the selected indoor positioning technologies are based on both the assignment text and reviewed literature. Some technologies, such as GNSS, are complex systems developed over decades, while technologies such as bluetooth and RFID are simple, cheap and relative novel systems. After giving a brief introduction to the workings of each technology, will suitability for INS integration be evaluated. If possible the different technologies will be compared in regard to capabilities, and integration feasibility.

2.2.1 Inertial Sensors

As stated in the introduction, inertial navigation systems (INS) use inertial acceleration and angular velocity to calculate an object's position and rotation. These forces are determined by accelerometers and gyroscopes respectively, which are the most common inertial sensors. These inertial sensors measure forces of the object it is mounted on, with reference to an inertial frame [**?**]. Inertial frames are any coordinate frames in which Newton's laws of motion are valid, and are hence neither rotating nor accelerating with respect to global reference frames. A global reference frame has coordinates defined from a fixed point of origin, and include coordinate frames such as earth centered coordinate systems or geographic coordinate systems.

Commonly, there are two ways to compute position, velocity and attitude in an INS. The first methods is based on a set of equations that convert measurements from the inertial sensors into position, velocity and attitude information, defined in a global frame of reference [13]. These equations are called navigation equations [2], and are quite intuitive. The sensed acceleration is first integrated into velocity. The velocity is then integrated into position. However, to integrate acceleration in the right direction, attitude is needed. This is obtained by integrating angular velocity, obtained by the gyroscopes [2]. An INS based on these navigation equations can be viewed in Figure 2.2. Due to measurement noise, the integration process will enhance

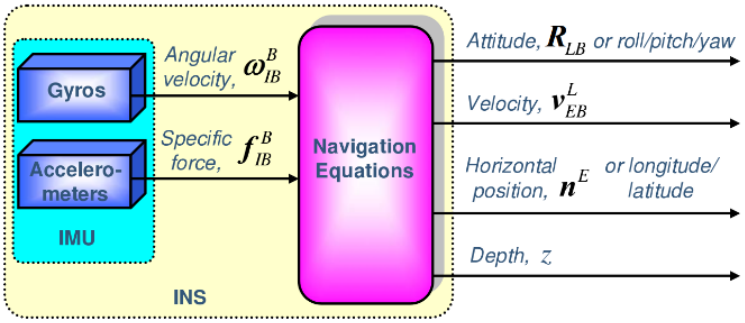


Figure 2.2: INS based on navigation equations [2]

errors that occur in an INS. Thus, INS have a low relative error over short time periods, but can over long time periods have a dramatical error increase [8]. This method will not be further discussed in this thesis, but research can be conducted in papers like [13].

The second method try to restrict compounding errors through integration by recognizing that people move one step at a time [14]. This method was therefore suitable for the work presented in this thesis. Measurements from inertial sensors are used to detect steps, and provide a means of estimating the distance and direction in which the step was taken. In this way error is proportional to the number of steps taken, rather than propagating on a fixed time interval. The latter approach is used in the presented work, and will therefore be utterly described throughout this thesis.

Given the second method, several reviewed papers indicate an accuracy within the requirements given for the WiLoc localization system. [15] use step detection and an adaptive step length estimation to compute walking distance. The step detection algorithm was 98 % accurate, while accuracy of the step length estimation algorithm was 95 % at the worst case. [16] also propose a method for INS using step detection and stride determination, in addition to an integration method of gyroscopes and magnetic compass for heading determination. Indoor walking tests gave less than 1 % step error, 5 % error for traveled distance and 5 % heading error. [17] used a chest-mounted IMU platform and microprocessor to test indoor positioning for pedestrians walking in urban or indoor environments. The approach investigated step detection and step length estimation for a DR system. Results

showed that the proposed algorithm for step detection performed well, with an overall system performance of 2 % distance error while walking forwards. Backward and sideways walk had a distance error of around 6.5 %. [14] also developed an algorithm to propagate a position by detecting stride events, measuring their length and estimating their heading with low-cost accelerometers and magneto-resistive sensors, but mounted the sensors on the user's foot. Through treadmill tests and field trials, the method was found useful for navigation, but could not compare with conventional methods of pedestrian navigation and satellite positioning. Improving signal conditioning and system monitoring of existing sensors can increase measurement accuracy, or alternatively can higher accuracy be achieved by applying higher quality, higher cost sensors. The author also suggest measuring common pedestrian motions to make the system more useful. Another paper that has looked into a foot-mounted IMU is [18]. The authors implemented two methods for distance measurements. The first method calculated orientation using quaternion output from the gyroscopes, and double integration of acceleration signals to estimate position. To better the estimation accuracy was a smoother algorithm based on Kalman filter used. The second method was an algorithm for distance measurements based on counting the number of steps, as described previously. Both methods indicated an average position error rate of less than 5 %. According to [**?**] was positioning error below 5 % of total traveled distance when comparing several of the most relevant algorithms for step detection, stride length and heading estimation.

Using step detection, step length estimation and heading determination in a DR-fashion, has several advantages compared to other methods for indoor positioning. Once the starting point of the object is set, the method does not require any outside information and is not affected by changes in the environment in which it operates, such as bad weather. One the other hand, since DR are based solely on internal sources, are errors that occur in the INS cumulative, increasing exponentially over time [19].To correct for cumulative error, an INS is dependent on external sources to provide the system with a correct positioning update. For this reason are INS often integrated with other positioning technologies. A goal for the work presented in this thesis, and this chapter in particular, was to investigate and document the feasibility of integrating an INS with different sensor configuration in order develop an accurate indoor localization system. The remainder of this chapter is devoted to this topic.

2.2.2 Global Navigation Satellite System, with focus on GPS

Global Navigation Satellite Systems (GNSS) are a constellation of satellites that broadcast navigation signals which allow any receiver on, or near the Earth, to determine its position with a precision ranging from within one meter to several meters, depending on the particular technology used. There are two GNSS mainly in use today, the United State's Global Positioning System (GPS) and the Russian Federation's Global Orbiting Navigation Satellite System (GLONASS). In addition, The European Union's Galileo and the Chinese's BeiDou are examples of GNSS which are currently under construction, both expected to be fully completed within the year 2020. The focus in this section is on GPS since it is available world-wide and free.

A short description on how GPS works follows. The satellites in a GPS send continuous signals towards the Earth, containing the satellite's' position and time of transmission. In order to determine an object's position one must have an antenna which receives the signals, and a receiver that translates the signals. The position of the antenna will be determined based on measurements of the time delay between the emission time and the reception time for at least 4 signals coming from different satellites [3]. This positioning method is a version of triangulation. Triangulation determines an object's location based on intersecting spheres, and the method is illustrated in Figure 2.3. One satellite places the object somewhere on a spherical surface, two satellites place the object somewhere along a circle between the two satellites, three satellites will place the object at the two points where the three spheres intersect, and a fourth satellite will therefor place the objects at just one point. The Earth's surface is often used as the fourth sphere if the object is on Earth.

Since GPS is a well established system, having receivers in a large number of devices, it would be beneficial for use in indoor positioning systems. However, there are two problems to be addressed. Firstly, GPS satellite signals are weak and construction materials therefore cause signal attenuation, ergo a reduction of signal strength, making it harder for GPS receivers to gain contact with the required number of satellites. Secondly, roofs, walls and other surfaces will cause the signal to propagate in multiple paths, giving uncontrollable errors.

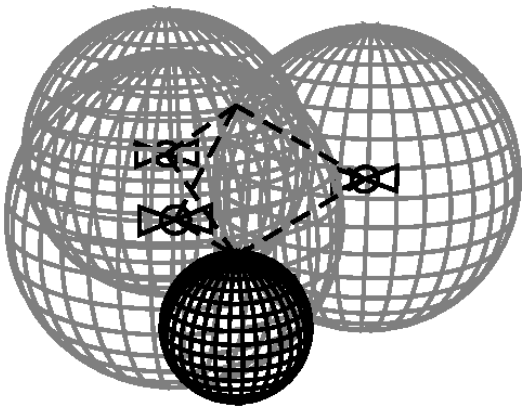


Figure 2.3: Triangulation: Using the Earth as the fourth sphere [3]

Despite this discarding GPS too quickly would be naive. Signal attenuation problems can be solved by high sensitivity receivers (HSGPS), and attempts to determine 3D position indoors have in fact been successful [20]. HSGPS receivers may further be assisted by external sources, such as Assisted GPS (AGPS). AGPS receivers use a cellular network to speed up the start up performance of a GPS when signal conditions are poor. When AGPS is not able to receive signals directly from satellites, the receiver connects to a server which always have clear line of sight (LOS) views of available satellites. The server provides the AGPS receiver with the same information as if the receiver where directly connected to the satellites themselves. A comparison of GPS and AGPS can be viewed in Figure 2.4. AGPS was found to improve GPS in weak signal environments when evaluated in [21]. Further, [20] list attempts for solving the problem of multipath interference, including proper siting of the antenna, antenna selection, proper receiver design and also postprocessing techniques.

Thus, according to reviewed literature there exist several solutions to the problem of signal attenuation and multipath interference in GPS. However, since a WiLoc node are dependent on low-cost products, a small form factor, and real time data, are solutions like HGPS and AGPS not optimal for providing the system with external positioning updates. The suggested solution therefore use a regular GPS antenna, which can provide the system with accurate positioning updates when possible, but still being able to function when GPS signal are unavailable.

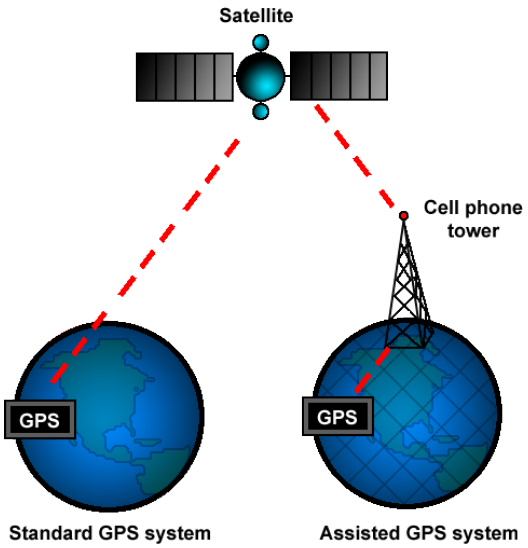


Figure 2.4: Difference between standard GPS and Assisted GPS [4]

2.2.3 Bluetooth

Bluetooth is a standard for exchanging data using low-power radio waves over short-ranges. Communication is achieved in the frequency range from 2.4000 GHz to 2,4835 GHz, which is within the 2.4 GHz ISM band, a license-free frequency band globally set aside for industrial, scientific and medical (ISM) devices. Bluetooth is divided into three power classes with different operating ranges. Class 1 have an operating range of 100 meters, class 2 can operate over a range of 10 meters, while class 3 have a operating range of 1 meters. Bluetooth coverage is therefore device dependent. However, the majority of available device are of class 2. Properties such as low-cost, small size, and high availability has lead to the development of several indoor positioning systems relying on the bluetooth standard [22].

Bluetooth communicate in a master-slave structure, where a master can be connected to a total of seven slaves at the same time, without the need for user interaction. For a system such as the one described in this thesis it would be natural that the master device was located at the human operator while slave devices was positioned at known locations inside the facility. While walking, the master device will start communicating when in range of any stationary devices. Since every bluetooth device has its own unique ID, one

can calculate the position of the master device based on communication data between the master and its connected slaves. Triangulation is a common method for determining the location of a master by measuring the distance from the master to at least three slaves, as well as fingerprinting, described Section 2.2.4. Unfortunately, bluetooth provides no accurate way for measuring distance, and an estimation is therefore performed based on various signal parameters. Such parameters include Angle of Arrival (AOA), Cell Identity (CI), Time of Arrival (TOA), Time Difference of Arrival (TDOA), and RX power level, which are all described in [23].

The focus of this section will be RX power level based positioning since this method have been applied in several of the reviewed approaches for indoor positioning [24]. RX power level based positioning use a signal parameter known as Received Signal Strength Indicator (RSSI) for distance estimation. RSSI states weather the signal power is above, below, or within a described ideal range for signal strength. A RSSI value of zero indicates an optimal signal power with regards to power consumption during transmission. Based on the laws of electro-magnetic waves is the expected behavior of the RSSI parameter a monotonically decreasing function for increasing distance [25]. A propagation model can therefore estimate the relation between signal strength and distance. Further, with the use of a positioning method, such as triangulation, the estimated distance can be used to calculate the position of the master device.

A problem with RSSI is the fact that the value is hardware dependent, and hence will the relationship between a RSSI value and distance vary for different Bluetooth devices. [23] presents a Bluetooth Local Positioning Application (BLPA) which uses RSSI and an extended Kalman filter to computer 3D position with a mean absolute error of 3.76 m. [24] describe Topaz, a bluetooth positioning solution with a 2 m accuracy with 95 % reliability. In addition to the problem of hardware dependency, will accuracy depend on the number of slaves deployed in the positioning system. While increasing the number of slave devices may result in higher accuracy, complexity and cost will likewise increase.

Since research papers state different results on accuracy, is it difficult to give a concluding remark.. What is absolutely certain is that Bluetooth can provide room-level granularity. Since Bluetooth is cheap, simple to use, and has low power consumption is this technology definitely suitable for providing an INS

with external positioning updates.

2.2.4 Wireless Local Area Network

Wireless Local Area Network (WLAN) is an umbrella term for technology providing high-speed wireless date connections between mobile devices in close range to each other. A WLAN typically consists of several devices wishing to communicate with each other and the outside-world, and access points (APs) making this communication possible. The IEEE 802.11 standard defines the specifications for implementing different WLAN communication. WLAN has become extremely popular and are already available in many indoor environments. Taking advantage of an already existing infrastructure is the main reason why WLAN is growing in popularity for indoor positioning.

The approach of WLAN indoor positioning is similar to Bluetooth positioning, where using RSSI as signal parameter is the most popular method [1]. APs are stationary devices located at known positions inside a facility. A human operator wearing a mobile device determines the received signal strength of all visible APs. The positioning system further use the RSSI in a positioning method to determine the operator's position. Positioning methods using RSSI can be subdivided into four stategies: Propagation modeling, Cell of Origin (CoA), Fingerprinting (FP) and multilateration. Fingerprinting based on RSSI is the prevalent positioning method of using WLAN and will therefore be described further in this section.

Fingerprinting is divided into two phases, an offline and an online phase. During the offline phase a radio map will be created. A radio map is a defined grid of points where an operator can be positioned, and each point contain several fingerprints. A fingerprint is a set of RSSI values and device IDs for a particular location. There are two reasons for the use of multiple fingerprint sets at each location. The first reason is the necessity of a statistically valid sample of measurements. The second reasons is that RSSI is dependent on the orientation of the human operator. In the online phase will the measured RSSI value be compared with the radio map and the closest match is determined thorough a pattern recognition algorithm. Figure 2.5 illustrate the offline phase of fingerprinting.

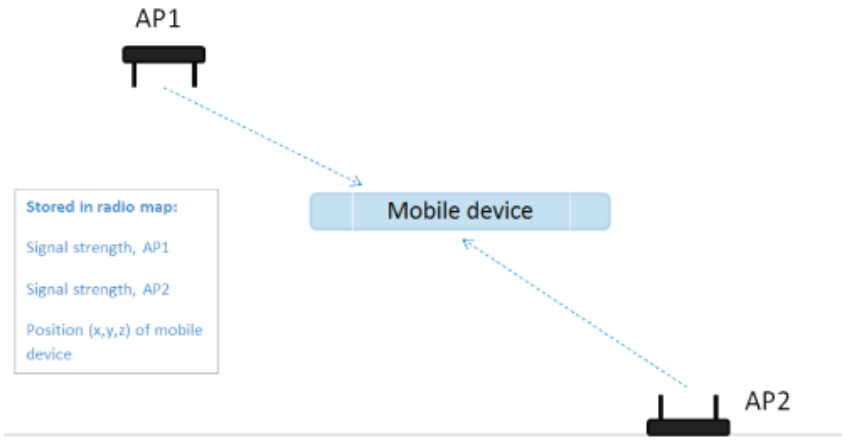


Figure 2.5: Illustration of the offline phase in a WLAN indoor positioning system [5]

[26] describe the positioning method of fingerprinting perfectly. The system start by building up a database of signal strength from different access point in the area of interest. An user then scans the signal strength in the wireless network and send the data back to the database where the closest match are found. The database returns the most likely position of the user. The paper investigate different approaches for fingerprint matching and database generation, all with a room-level accuracy. RADAR is another localization system using WLAN fingerprinting [27]. RADAR use signal strength information gathered at multiple receiver locations to triangulate the user's coordinate. Experimental results show that with high probability is RADAR able to estimate an user's location to within a few meters of actual location. Findings in [27] show that the performance depends on APs density, number of data points taken, and the orientation and speed of the user. [1] include a table of localization approaches using WLAN, and depending on the density of calibration points, reaches fingerprinting accuracies of 2 m to 50 m. Disadvantage of fingerprinting is that is requires a previous set of measurements, which must be recalculated if the environment changes. Further, as previously mentioned is RSSI hardware dependent, making comparisons of different devices difficult.

An argument against a WLAN indoor positioning system is the fact that

it may be time consuming and will be multipath influenced. However, if a human operator is using a WiLoc node inside a facility with existing WLAN infrastructure, literature review indicate that WLAN could be used for indoor positioning, either alone or integrated with other technologies. An advantage of WLAN over Bluetooth is that the probability of an already existing infrastructure is higher, and WLAN will thus already be part of the communication structure, making it a cost-effective positioning system. Further, unlike GNSS does not WLAN require a line of sight. Given that WLAN is already installed, this technology is a good option for providing an INS with external positioning updates.

2.2.5 Radio Frequency Identification

Decreasing size and low-production costs have increased Radio Frequency Identification's (RFID) popularity in indoor positioning. RFID technology is similar to Bluetooth and WLAN, and use radio waves to transfer data between two types of devices, a RFID tag and a RFID reader. The concept is simple. A reader sends out radio signals, listen for response from nearby tags, and read the data sent back from responding tags. Tags contain unique stored information, which are used to determine position. Tags can use an internal battery, or harvest power from the RFID reader in order to respond to a request. RFID tags are labeled either active, passive or semi-active. Active and semi-active tags have internal batteries applied to run their circuitry. Active tags also use the battery to broadcast a signal to a reader, whereas semi-active tags draw power from the reader. Batteries increase the possible reading distance of a tag, and active and semi-active tags can consequently be read 30-100 meters away, depending on battery capacity. Passive tags have no battery and must draw power from the reader. These tags are read up to 10 meters. Such tags have lower production cost and are smaller in size since they have less hardware than active and semi-active tags. Coverage of a RFID indoor positioning system is therefore device dependent.

There are two ways to implement a RFID positioning systems indoors. The first approach is to apply stationary readers and mobile tags, and the second, opposite approach, use stationary tags and mobile readers. [28] place passive RFID tags on small and inexpensive objects, and present a system that generates reminders of objects being left behind, with RFID readers positioned

in the environment frequented by users. This approach will be practical for application in which several objects move around well-arranged and separate areas through some checkpoints with RFID readers. However, if the number of checkpoints increases, or the positioning requirement is higher, will a "movable tag, fixed reader"-system become expensive and impractical. For this reason is this approach not suitable for the system presented in this thesis.

Contrary, [29] describes a RFID-method for indoor positioning with a "mobile readers, scattered tags"- infrastructure. In this approach the moving objects are equipped with readers, while the tags are distributed in the environment. The paper presents a pattern recognition and classification method to estimate an object's physical location, where real-time readings are compared to predefined coverage samples. This predefined sample contain information about which RFID tags are reachable by a reader at a certain point. Figure 2.6 illustrate this positioning method, where a RFID reader are within reach of tags B, C, an E, and must therefore be located within the gray area. The resulting estimation error is within one meter, with an accuracy of 97 %. Since pattern recognition requires preallocated information, a simpler approach is to use single tags with location information that are sent to readers within reach of the tags. Advantages of this approach is a low-cost and simple system. However, when positioning information are based only on the reading of a single tag the system will be prone to high error rate due to unstable communication in an industrial environment.

[30] compared RFID, Bluetooth and WLAN technology. RFID and Bluetooth obtained a tracking accuracy of half a meter or better, while WLAN measurements yielded an accuracy of a few meters. The paper further conclude that a combination of all three technologies would improve positioning performance. Taking advantage of all three technologies would be beneficial, but increase system cost, maintenance, and complexity, and are not realistic for the INS prototype system presented in this thesis.

Similar to Bluetooth and WLAN is a concluding remark complex, since accuracy is related to the density of tag deployment and reading ranges. [1] states an accuracy range from 15 cm to room-level positioning, and thus are RFID a good option for providing an INS with external positioning updates.

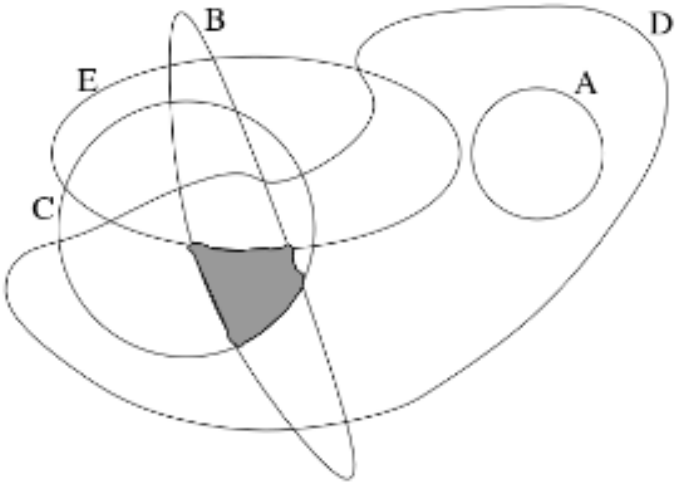


Figure 2.6: Pattern recognition. RFID reader are within reach of tag B, C, and E, are must therefore be located within the gray area. [**?**]

Table 2.1: Overview of indoor positioning technologies. Coverage refers to ranges of single nodes. [1]

Technology	**Typical Accuracy**	**Typical Coverage (m)**	**Typical Measure Principle**	**Typical Application**
Cameras	0.1mm -dm	1-10	angle measurements from images	metrology, robot navigation
High Sensitive GNSS	10m	'global'	parallel correlation, assistant GPS	location based services
Inertial Navigation	1 %	10-100	dead reckoning	pedestrian navigaiton
Infrared	cm -m	1-5	thermal imaging , active beacons	people detection, tracking
Infrastructure Systems	cm - m	building	fingerprinting, capacitance	ambient assisted living
Magnetic Systems	mm - cm	1 -20	fingerprinting and ranging	hospitals, mines
Other Radio Frequencies	m	10 - 1000	fingerprinting, proximity	person tracking
Pseudolites	cm - dm	10 - 1000	carrier phase ranging	GNSS challenged pit mines
RFID	dm - m	1- 50	proximity detection, fingerprinting	pedestrian navigation
Sound	cm	2 - 10	distance from time of arrival	hospitals, tracking
Tactile & Polar Systems	ym - mm	3 - 2000	mechanical, interferometry	automotive, metrology
Ultra-Wideband	cm - m	1 - 50	body reflection, time of arrival	robotics, automation
WLAN/WiFi	m	20-50	fingerprinting	pedestrin navigation, LBS

Chapter 3

Inertial Navigation System

This chapter starts by explaining the different components of an inertial navigation system. Following the first section is an explanation to the concept "Frame of reference" given" before the indoor positioning method pedestrian dead reckoning is presented. The chapters ends with introducing the selected IMU for the presented work of this thesis.

3.1 Inertial Systems Definitions

Inertial navigation systems (INS) are systems using measurements from an inertial measurement unit (IMU) to compute position, velocity and attitude of an object, as illustrated in Figure 3.1. An IMU contains an inertial sensor assembly (ISA), in addition to interfacing hardware, low-level software for down-sampling, temperature calibration and vibration compensation [6]. Inside the ISA is a cluster of mounted inertial sensors, usually with three orthogonal accelerometers and gyros installed on each axes, x, y and z. The gyroscopes and accelerometer measure angular velocity and linear acceleration, respectively in all three dimensions. Some IMUs also contain three orthogonal magnetometers. If magnetometers are applied can the magnetic field around the IMU be measured. By processing signals from these sensors is it possible to track the position and orientation of a device.

An INS can be design with various performance characteristics, but is classi-

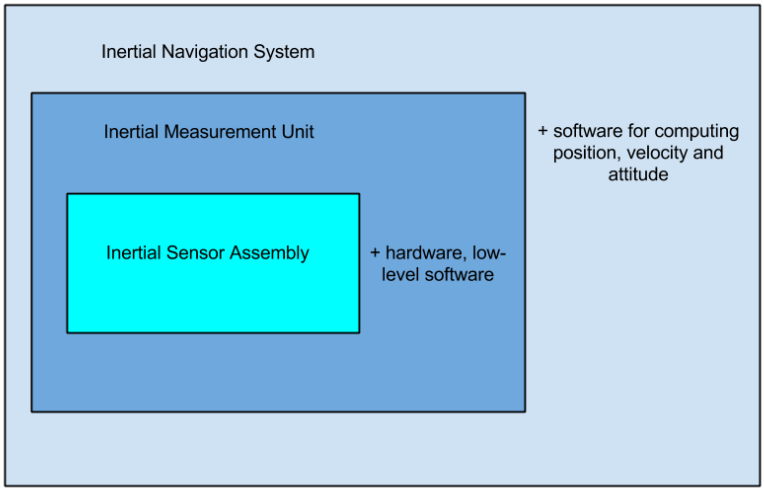


Figure 3.1: Definition of ISA, IMU and INS. [6]

fied based on the frame of reference in which the inertial sensors operate [7]. Generally, an INS fall into two categories:

- Stable platform system (Gimbal)
- Strapdown

A stable platform system isolate the inertial sensors from the angular movements of the object itself. This is achieved by means of gimbals and torque servos which null out the rotation of a stable platform on which the inertial sensors are mounted. Generally speaking can gimbals be characterized as devices that allows rotation around a center point, and will for an INS be recognized as a ring or frame mounted on an axis, allowing an object to pivot freely on that axis. For an INS will three gimbals be used, one for each axis, mounted as shown in Figure 3.2, allowing the platform six degrees of freedom. The outermost frame will mount to a larger surface, such as instrument panels on boats. The stable platform is mounted on the innermost frame and contain the inertial sensors. When an object is moving will it cause rotation on the platform. This rotation is detected by the platform's gyroscopes, which cause signals to be fed back to the torque servos in order to keep the platform stable [7]. Thus will the measured values from the inertial sensors be in another frame of reference than the moving object.

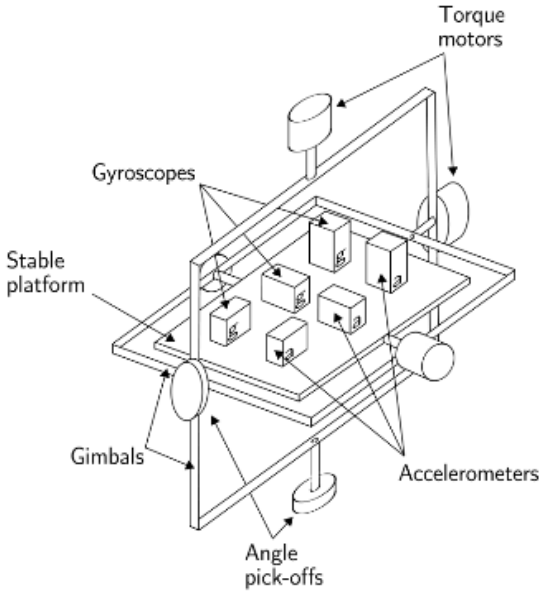


Figure 3.2: A stable platform IMU [7]

Contrary to a gimbal system is a strapdown system firmly attached to the object, and moves with it. The inertial sensors will therefore experience the same changes in rotation as the object in motion, and values from the inertial sensors are therefore measured in the same frame of reference as the moving object. A strapdown system maintains attitude analytically by adjusting rotation matrices or other attitude representations [6]. The IMU used in the presented work of this thesis are based on MEMS sensor, and are therefore classified as a strapdown system.

3.2 Frame of reference

In the preceding section was the phrase frame of reference used several times. This section explains the concept of frame of reference, in addition to the most common coordinate frames used in navigation.

An object's motion is usually described as movements in a coordinate system with two or three different axes. But where is the origin? What are the

directions of the axes? A frame of reference defines how to mathematically represent and measure properties of an object in the physical world. If the task were to describe the motion of a person on a bus moving with constant speed, the motion equations would be completely different, depending on the frame in which we are looking at the problem. One coordinate system could be define as moving with the bus, another with center at the person on the bus, and a third coordinate system in an area where the bus is passing by. Motion is definitive relative, and dependent on the frame of reference. In the same way as the bus example, most INSs have measurements in different coordinate systems. The choice of coordinate systems is made by the system designer, and is dependent on the application and the designer's priorities [6]. The most common coordinate frames used in navigation can be classified into three categories: Earth Centered Coordinate Systems, Geographic Coordinate Systems and Vehicle Coordinate Systems.

Earth Centered Coordinate Systems include the Earth Centered Inertial (ECI) frame and the Earth Centered Earth Fixed (ECEF) frame. As the name implies, both coordinate frames have their origins at the center of the earth, and in addition, a z-axis pointing along the Earth's rotation axis, extending through the North Pole. For the ECI frame the x-y plane coincides with the Earth's equatorial plane, with the x-axis pointing towards the vernal equinox and the y-axis completing the right handed orthogonal coordinate system, defined in [31]. An equinox is a phenomenon that happens twice a year, a day when day and night are theoretically of equal duration because the Earth's north and south poles are not tilted towards or away from the sun. A vernal equinox is the spring equinox. At this time will the x-axis be defined by the line segment pointing from the center of the Earth towards the center of the sun, when the sun crosses the Earth's equator moving north.

What is interesting with the ECI frame is that it can be considered an inertial frame since the axes is fixed and therefore not experiencing any acceleration or rotation. An inertial frame of reference is a frame in which the "law of inertia" and the Newtonian laws of motion are valid. The law of inertia state that an object in motion or in a state of rest will continue if not acted upon by an external force [32]. Outside an inertial frame, ie a frame which are experiencing acceleration, will fictitious forces act upon the object and the Newtonian laws of motion will not take the simple form they do in an inertial frame. Hence, in a non-inertial reference frame will the laws of motion vary from frame to frame depending on the acceleration on that frame with respect

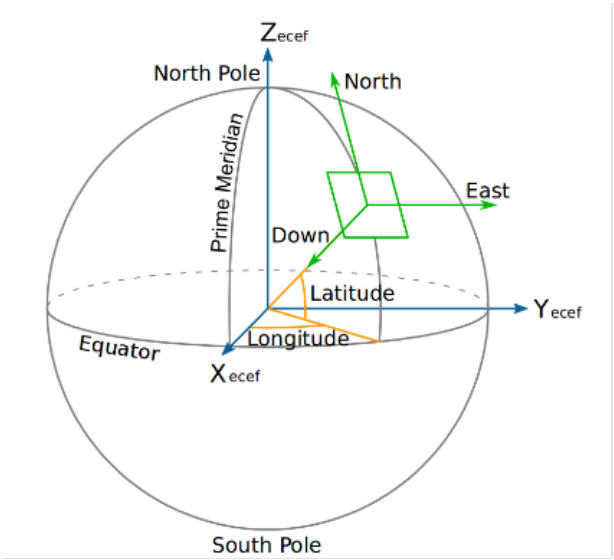


Figure 3.3: Graphic showing Geodetic, NED and ECEF coordinate frames [8]

to an inertial frame. Fictitious forces act on all masses in the non-inertial reference frame of an object, and examples of such forces include centrifugal force and Euler force.

The ECEF frame uses the geodetic coordinate system, combining latitude and longitude angels with a height z, as defined for the ECI frame. Latitude is the angle that specifies the north to south position of a point on the Earth's surface, and longitude is the angle that specifies the east to west position [8]. For the ECEF frame, the x-axis intersect the sphere at O°latitude and 0°longitude. The y-axis complete the right handed orthogonal coordinate system. The ECEF frame rotates relative the ECI frame with the Earth rotation rate ω_e, and is therefore not an inertial frame. The ECEF is used to represent absolute positions on the Earth and can be viewed in Figure 3.3.

Geographic Coordinate systems include the North East Down (NED) frame, tangent frame (t-frame) and the Azimuth reference frame (AZI). This section focus on the NED frame, which is used to express velocities and relative positions with the origin at current position. The x-axis point towards true north, parallel to the line of latitude at that point, the y-axis point east, parallel to the line of longitude at that point. The z-axis points down, towards

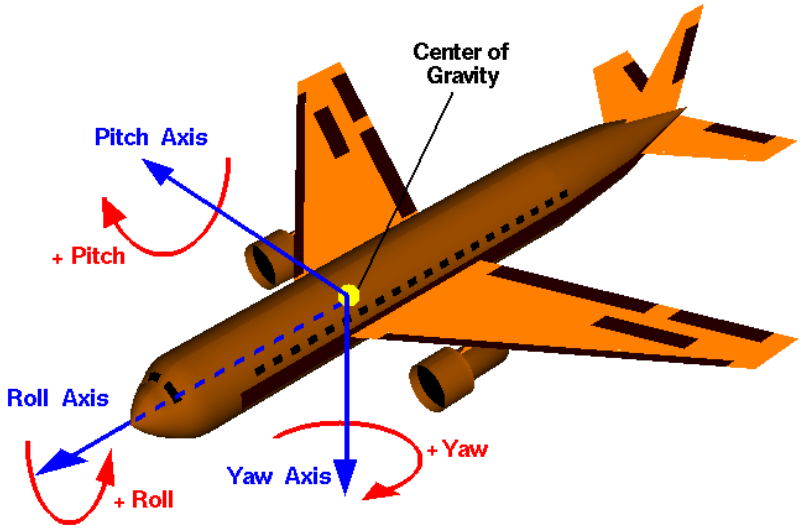


Figure 3.4: Body frame of a plane, indicating roll, pitch and yaw angles. [9]

the centered of the earth. Figure 3.3 also illustrate the NED frame at a position on the Earth.

The last frame to consider is the BODY frame. The BODY frame is a vehicle coordinate system, meaning that the frame moves and rotates with the object. The origin often coincides with the object's center of gravity. The x-axis points in a forward direction, the y-axis to the right side and the z-axis downwards. The BODY frame is related to the NED frame through the Euler angles roll, pitch, yaw [6]. The three angles are often used to describe orientation, where roll is the angle around the x-axis, pitch is the angle around the y-axis and yaw is the angle around z-axis, sometimes called Heading [8]. A body frame system can be viewed in Figure 3.4

3.3 Pedestrian Dead Reckoning

The INS system presented in this thesis detect steps, estimate step length and determines heading for each step in order to compute position. Step detections and step length estimations are performed solidly on sensor readings, while heading determination must relate the frame of reference in which the sensor measurements are applied, to a geographical frame in order to determine an user's global position.

For each detected step n, with step length d_i at a heading of θ_i measured counterclockwise from North, the Eastward displacement Δx_i of a walking person is

$$\Delta x_i = d_i \cos\theta_i \tag{3.1}$$

Similarly, the Northward displacement is

$$\Delta y_i = d_i \sin\theta_i \tag{3.2}$$

Thus, after several steps, n, will the person's position with respect to an initial position $[x_0, y_0]$ be

$$x_n = x_0 + \sum_{i=0}^{i=n} d_i \cos\theta_i \tag{3.3}$$

$$y_n = y_0 + \sum_{i=0}^{i=n} d_i \sin\theta_i \tag{3.4}$$

This method for computing position of a pedestrian is known as pedestrian dead reckoning.

3.4 Advanced Navigation and Spatial IMU

Statoil selected an off-the-shelf IMU delivered by advanced navigation in order to develop a prototype WiLoc localization system. Advanced Navigation is a privately owned Australian company that specializes in the development

Figure 3.5: Spatial IMU (v3). Released on the 17. th of September, 2013. [8]

of 3D navigation technologies [**?**]. The selected IMU is named Spatial (v3), and can be viewed in Figure 3.5.

Spatial weighs 37 grams, and has 30x40.6x24 mm dimensions, including tabs. Operating voltage are 5 to 36 V, with a power consumption of 100 mA at 5 V, typically. This minimal size, weight and power requirements allow for easy integration into almost any system. In addition, Spatial's aluminum enclosure is waterproof, dirtproof and shock-proof to the IP67 standard and shockproof to 2000 g.

Spatial is an inertial navigation system (INS) and attitude and heading reference system (AHRS) providing position, velocity, acceleration and orientation for reliable navigation. Spatial contain temperature calibrated MEMS inertial sensors (accelerometers, gyroscopes, magnetometers and a pressure sensor) combined with a GNSS receiver. The inertial sensors are calibrated to provide consistently accurate data over an extended temperature range of -40 °C to 85 °C. The GNSS receiver included in the spatial supports all of the current and future satellite navigation systems including GPS (USA), GLONASS (Russia), GALILEO (Europe) and BeiDou (Chinese). When Spatial can not get a satellite fix, it continues to navigate using inertial navigation. In addition, spatial contains a battery backup that allows it to hot start inertial navigation from its last position in 500 milliseconds and obtain a GNSS fix in approximately 3 seconds. Spatial contains Advanced Navigation's new sensor filter which is more intelligent than the typical extended kalman filter and is able to extract significantly more information from the

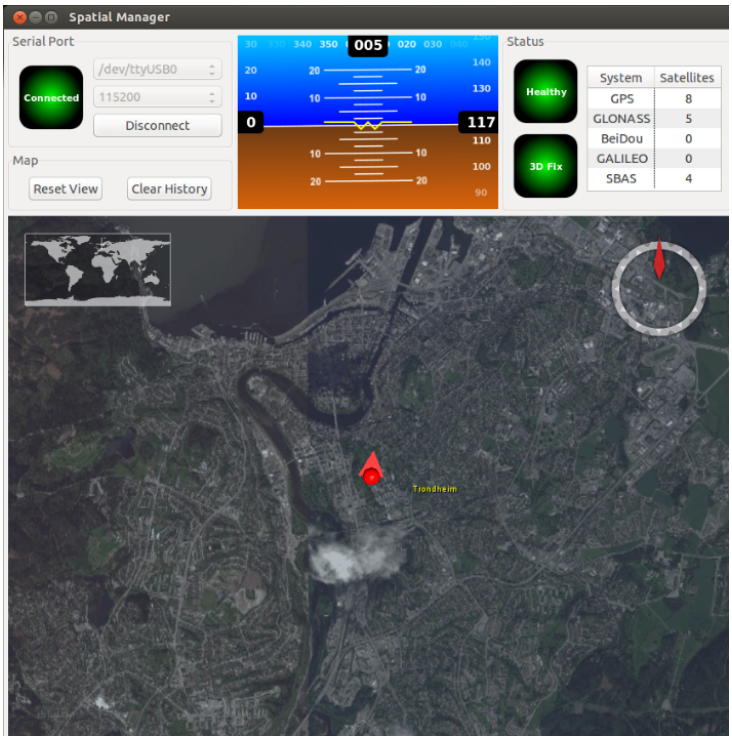


Figure 3.6: Spatial IMU (v3). Released on the 17.th of September, 2013. [8]

data by making use of hum inspired artificial intelligence. Spatial navigation, sensors, GNSS, communication and hardware specifications can be viewed in Appendix A.

3.4.1 Spatial Manager

The Spatial IMU is delivered with a program called "Spatial Manager" which is a fully operational INS and AHRS. Figure 3.6 shows the GUI for this program. Position and heading is calculated using either sensor data from the IMU, or with satellite signals from the GNSS antenna. However, for the presented work of this thesis are only the raw sensor data collected in order to implement the suggested solution to the WiLoc localization system.

Chapter 4

Step Detection

By dividing the challenges of indoor navigation into small sub-problems the task becomes less complicated and easier to carry out. The first step, pun intended, in this divide-and-conquer strategy is to make a step detection algorithm based on raw sensor data. Several types of sensors can be used in step detection algorithms, including GPS, camera-based systems and ultrasonic, infrared, optical, inertial or electromagnetic sensors [33]. Implementing the best algorithm for different situations is a challenging task, and is not only dependent on the sensors, but other factors as well, like sensor placement on the user's body, walking environment, individual gait pattern, data quality, and computational restraints. Since this thesis present inertial navigation using an IMU will the reviewed literature in this chapter focus on step detection algorithms based on sensor data from the IMU's accelerometer.

As mentioned will the accuracy of a step detection algorithm depend on different factors. This chapter will explain how human gait characteristics and sensor placement can effect the design choices of an algorithm. Following these two sections is a literature review on a selection of step detection algorithm. The selection aims to cover the main basics of step detection, and ends with a in-depth description of two step detection algorithms, the Pan-Tompkins Algorithm and a Libby-Peak based algorithm, which both are implemented and tested during the work on this thesis.

4.1 Human Gait Characteristics

Gait is the medical term to describe how humans walk, and is an individual and unique pattern. Analysis of the human gait shows that the human locomotion is cyclic, repeatable and consistent between individuals [34]. While walking, humans generate periodic series of motions where the center of gravity of the body maintain an approximately constant horizontal velocity, with a smooth vertical oscillation of just a few centimeters [35]. The horizontal velocity of the foot on the other hand, varies each stride from stationary to over twice the velocity of the torso [14]. Due to physiological limitations will the motion not be completely periodic but incorporate some randomness in it, caused by internal and external perturbations. The amount of randomness in the human gait reflects the quality of neuromuscular control of the human being [34]. Injuries, diseases and aging are some factors that affect the neuromuscular system and can degenerate walking capabilities of a human being. Thus, a step detection algorithm must be design to manage different gait cycles.

The human gait can be divided into two sub phases; the stance phase and the swing phase. The stance phase is described as the time when the reference foot is touching the ground, and is initialized when the heel of the foot contacts the ground. The stance phase ends when the toe leaves the ground, thereby initializing the swing phase. The swing phase continues, as the foot pushes forward to begin the next stride, until the heel again touches the ground. The foot strike initialize a new stance phase, completing the gait cycle, which can be viewed in detail in Figure 4.1.

Step detection algorithms must not only address the challenge of gait pattern variations from one individual to another, but also take into account that humans gait pattern are dependent on walking velocity and direction. As a person increase step frequency, going from slow walking to fast walking and running, the gait cycle will change. While a walking gait is characterized by the fact that there is always a foot touching the ground, the contrary is true for running, where booth feet are in the air during a certain amount of time. A running gait also have a different change in the center of gravity, which is at its lowest when the leg is vertical. For a walking gait will the center of gravity reach a peak when the leg is vertical. Further, if a person starts walking backwards or sideways it will additionally effect the gait cycle.

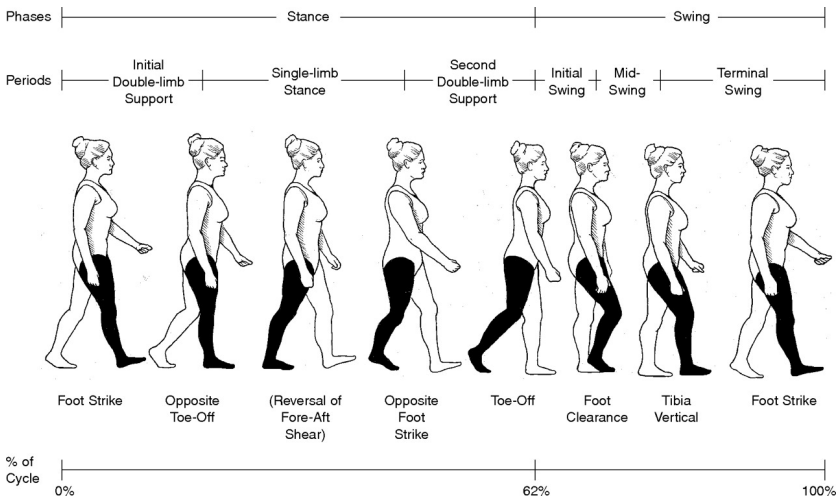


Figure 4.1: Gait cycle [10]

For theses reasons have some reviewed literature proposed using a motion classifier [17]. Based on results from the classifier can the navigation system then optimize system coefficients to the particular setting. It was not found time to developed a motion classifier in this work, but possible advantages are discussed when looking at overall system performance in Chapter 9.

4.2 Sensor placement

The sensors used for step detection can be located at any convenient point on the user's body as long as the accelerometers can sense harmonic motions and the impact accelerations from the user's gait [36]. However, as discussed in [13], the sensor placement has a major influence on step detection algorithm design. Various algorithms take advantage of the fact that though all sensor location experience the same displacement, they will be subject to different cyclic dynamics [34]. Using the knowledge of these dynamics will make it easier to understand the information obtained from the sensors.

[37] summarize some studies which have investigated accelerometer placement for detecting everyday activities. Such studies indicate that some body parts, such as foot, thigh, wrist, hip, lower back and chest, are more favorable than others for activity detection. The Spatial IMU was therefore

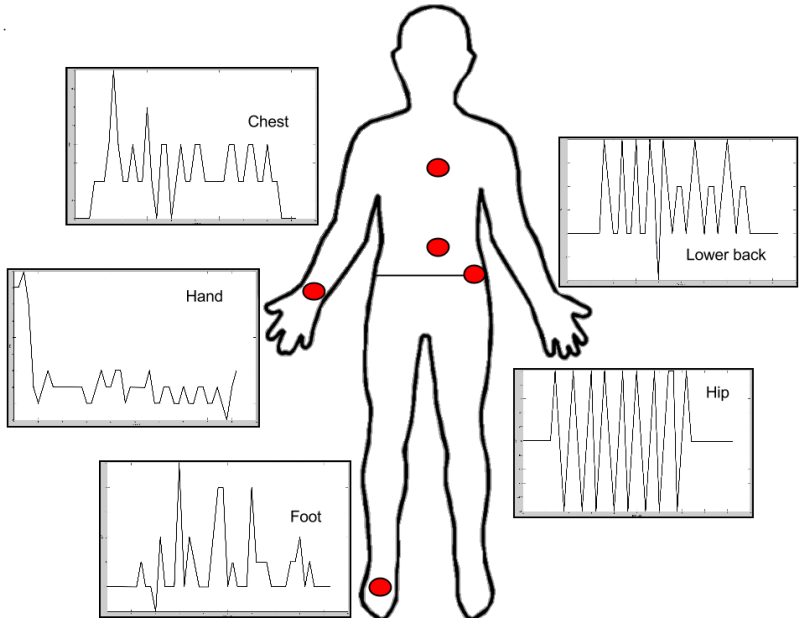


Figure 4.2: Ten second recording of vertical acceleration obtained from the Spatial IMU placed on the foot, hip, lower back, chest and wrist

placed at the foot, hip, lower back, chest and wrist to illustrate the variation in cyclic dynamics sensed by the accelerometers. The vertical accelerations obtained are illustrated in Figure 4.2. In order to help the reader understand accelerometer readings, signals from both the foot and the hip will serve as examples.

Starting at the bottom, the cyclic dynamics which characterize foot-mounted accelerometers is the experience of zero-velocity every time the foot strikes the ground. As viewed in the lower, leftmost graph in Figure 4.2, the periodically intervals where the signal from the accelerometers are constant represent the time at which the reference foot comes to rest, indicating that the user has taken a step. The peaks that follow each step are the acceleration that occur when the reference foot pushes from the ground (toe-off). Similar peaks can be seen right before a new step interval, indicating that the reference foot touches the ground again (foot strike). Between a toe-off and a foot strike is the swing phase, as described in Section 4.1, and can be recognized by a gradually increase in acceleration. The accelerometer signal will only be constant when the reference foot is at rest, and thus is the

number of steps twice the number of intervals with constant acceleration. Unfortunately are not each phase of a gait cycle identical in the graph, or in real life, and the graph is accordingly a perfect illustration on how the sensor can be effected by noise.

The lower, rightmost graph in Figure 4.2 show the cyclic dynamic from the sensed accelerometer at the hip. The periodically intervals of positive peaks represent the time at which a foot strike the ground. Similar will the intervals of negative peaks represent the end of a step (toe-off). This graph is much more periodic and clean, which indicates that for the case of the Spatial IMU and the test setting used, the hip is less influence by noise.

It can additionally be mentioned that acceleration amplitude is dependent on the force used to move your feet. For this reason will the accelerometer signal from a person running have higher amplitude than a person walking.

Going back to the literature, [37] searched an optimal placement of sensors for identification of activities like walking, running, lying, standing and walking up and down stairs, testing various machine learning algorithms. The article conclude that the hip will be the best location for a tri-axial accelerometer for detecting the studied range of everyday activities, though data from all mentioned locations provided similar levels of accuracy. [38] compare activity classification accuracy for accelerometers attached on three different locations, right wrist, left hip and chest. The results from this study show that the right wrist discern activities such as walking and performing hand movements, while the left hip played the most important role when classifying activities such as sitting, running, crawling and lying down. The accelerometer at the chest was best for classifying activities such as squatting and standing. [39] used a single accelerometer attached to the hip to recognize user activities like standing, walking, running, climbing up and down stairs, sit-ups, vacuuming and brushing teeth. Except activities limited to only movements for mouth and hands, the overall performance was good with use of only a single hip-mounted triaxial accelerometer.

It is also reasonable to study the effect of using multiple accelerometers attached to different body parts for the purpose of activity analyzing. [40] placed accelerometers at the upper arm, lower arm, hip, thigh and ankle, and by testing a number of activity classifiers achieved overall accuracy rate of 84 %. Further investigation showed that with reducing the number of accelerometers to only the thigh and wrist reduced the accuracy with only

3 %. [37] argue that incorporating multiple accelerometers became cumbersome for the user and increased the complexity of activity classification, but would on the other hand decrease the number of recognizable activities. This indicates that it is sufficient to use a single sensor for the purpose of doing step detection.

In addition to examine signal characteristics for various sensor locations, practical considerations for those locations are also important. Such considerations include sensor weight, size, temperature, power source, communication and cabling to a potential user interface, and attachment method. A big and heavy sensor will be more suitable mounted at the hip rather than at the foot since the impact will be less on the hip. Further, a torso-mounted sensor can have an integrated user interface and can easily be attached to the user's body, clipped to clothing for instance. Foot-mounted sensor often contain a separate user interface, making the sensor impractical with regards to communication since the sensor must either be wireless or use wires up the user's leg. On the other hand, when looking at signal characteristics, can foot-mounted sensors measure step length directly. In the end will application requirements determine the balance between advantages and disadvantages of different sensor locations.

Based on these studies it seems reasonable to use a single IMU mounted on the user's hip. Several sensors will increase system costs and make the system more complicated, and since the activities at interest is similar, such as walking, running and standing, should one sensor be able to identify these activities with the required accuracy. The sensor is place on the hip for practical reasons. Using Spatial IMU at the hip is proven to obtain good accelerometer readings. In addition, several studies report good results with use of a hip-mounted IMU. The challenge now is to design an accurate step detection algorithm.

4.3 Step Detection Algorithms

Inertial sensors have become popular due to their low-cost, small size, light weight, and limited power requirements, and many papers have been published to demonstrate that accelerometers and gyroscopes can be used to detect simple temporal or spatial features [41].

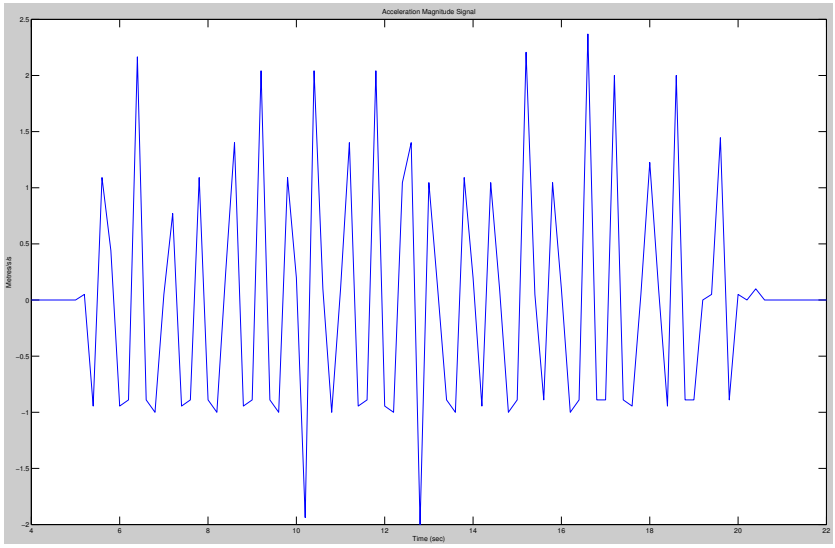


Figure 4.3: Magnitude of triaxial acceleromter signal

Figure 4.3 demonstrate how a triaxial accelerometer signal from a hip-mounted IMU of a walking pedestrian can appear. The human gait's periodic, cyclic pattern, as described in Section 4.1, can be seen from the accelerometer signal and step counting is easy to do, for instance by looking at the peaks. Though step detection in some cases can be easy for the naked eye, it turns out to be somewhat harder to implement in an algorithm. A step detection algorithm must deliver a satisfying accuracy independent of the different signals that arise as a result of individual gait patterns, surface conditions and noise.

Step detection is achieved by identifying various events of a gait cycle. The goal is therefore to decide what you are looking for, and then highlight it. As described in Section 4.2, sensor placement is one way of helping yourself to identify particular gait cycle events, since some body parts will give a cyclic dynamic which highlights a particular gait cycle event more than others. For example, the foot-mounted sensor had intervals with constant acceleration, indicating a step on the reference foot. The hip-mounted IMU showed positive peaks to indicates steps. Then, when receiving the measured acceleration from a particular part of the body, those signals can be mathematically manipulated to further highlight gait cycle events. Such manipulation is achieved through the use of numerous mathematical operations or applying various filters.

After highlighting particular gait cycle events, it is time to start detection these events. Step detection algorithms generally use one or several of the listed approaches to detect a step:

- Peak Detection
- Zero-crossing Detection
- Stance-phase Detection

Peak detection is based on the assumption that local maximums correlate to footfalls [42]. Maxima can easiest be explained as a point of greater magnitude than proceeding and following points, and local maxima can therefore be found by checking the following statement:

$$X_{i-1} \text{ is a maximum if: } X_{i-1} > x_i \cap x_{i-1} > x_{i-2} \tag{4.1}$$

Zero-crossing Detection, as the name implies, is an approach that assume that a step occur when the measured acceleration crosses zero, and find these points [43]. A challenge of zero-crossing is that the acceleration signal is not a continuous function, but rather a set of discrete points. The signal will therefore not necessarily be be defined when crossing zero. This problem can be solved by the method of interpolation. Interpolation constructs new data points between a discrete set of known points [44], and thus makes it possible to estimate a zero-crossing from the discrete accelerometer data points. Different interpolations methods exists, include piecewise, polynomial, spline and linear interpolation, to mention some, where each method should be evaluated based on accuracy, cost and the number of needed data points. Linear interpolation have been used in this work and are therefore briefly explained. The method use two data points (x_a, y_a) and (x_b, y_b) to calculate point y at point x, and is a simple and quick method. The linear interpolation is given by Equation 4.2.

$$y = y_a + (y_b - y_a)\frac{x - x_a}{x_b - x_a} \text{ at the point } (x, y) \tag{4.2}$$

A disadvantage of linear interpolation is that it is not a very accurate method. The interpolation error is given in Equation 4.3, and show that the error is proportional to the square of the distance between the data points. g is the

function one wishes to interpolate, and is twice continuously differentiable. x lies between x_a and x_b [44].

$$|f(x) - g(x)| \leq C(x_b - x_a)^2 \text{ where } C = \frac{1}{8} \max_{y \in [x_a, x_b]} |g''(y)| \tag{4.3}$$

Stance-phase detection find the time at which the foot is touching the ground and assume then that a step has occurred. This method is most beneficial when the sensor is mounted on the user's shoe since the accelerometer readings are constant during this phase, as shown in Figure 4.2. Though the sensor may be effected by noise, is the stance-phase almost indisputably constant, and can easily be determined by periodically checking the accelerometer variance against a certain threshold [13]. Stance-phase detection benefits of Zero Velocity Updates (ZUPT) that can reset foot speed at every step, and by that keep the INS errors bounded in the absence of other correction signal, such as GPS, Bluetooth or RFID. Because of these advantages are stance-phase detection popular and several papers use stance-phase detection with foot-mounted IMUs [13], [14], [34], [16], [18], [45], [46]. Unfortunately, since the work presented in this thesis use a hip-mounted IMU, can this approach not be used. In order to compensate for the lack of ZUP, waist-worn systems most have more sophisticated algorithms [47].

As already indicated are the step detection algorithms implemented in this work not based on stance-phase detection since it is unsuitable for hip-mounted IMUs. However, both peak detection and zero-crossing are implemented and tested. [43] states that a disadvantage of peak detection is the fact that the accelerometer output is effected by the user's walking velocity, while as zero-crossing is not. This statement, as well as other disadvantages and advantages of the two implemented algorithms will be discussed in Chapter 9.

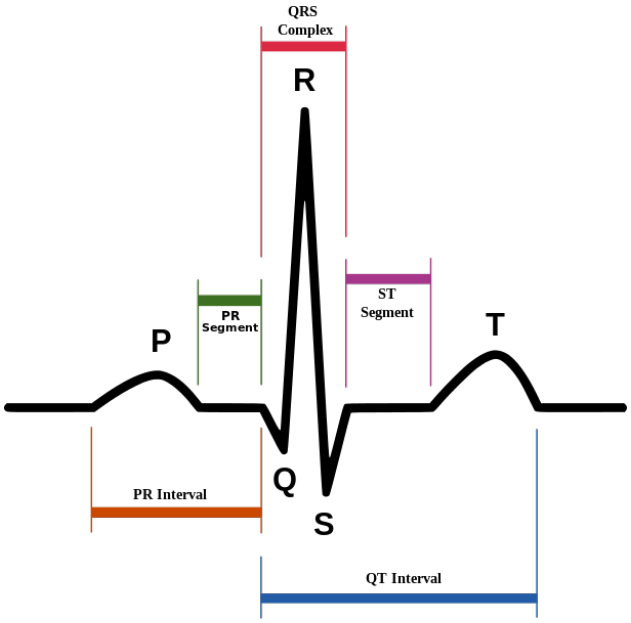


Figure 4.4: QRS complex of ECG signal

4.4 Implemented Step Detection Algorithms

4.4.1 Pan-Tompkins Algorithm

Pan-Tompkins algorithm is a concept of using a set of filters to highlight peak events of a gait cycle, resulting in signals that are very easy to analyze [48]. The idea dates back to Jiapu Pan and Willis J. Tompkins work from 1985, viewed in [49], and was originally used for detecting the QRS complexes of an electrocardiography (ECG) signal. ECG is the recording of the heart's electrical activity and a QRS complex of such a signal can be viewed in Figure 4.4. Studies have with good result investigated if the R-peak detection in ECG signals describe by Pan and Tompkins also can be used for step detection in accelerometers signals [11] [48]. Comparing Figure 4.4 with an accelerometer signal from a foot-mounted IMU, as viewed in Figure 4.2, some similarities can be seen. The QRS complex looks is comparable to a push-off phase, and the QT interval is similar to a swing phase where the T-peak indicates the end of this phase.

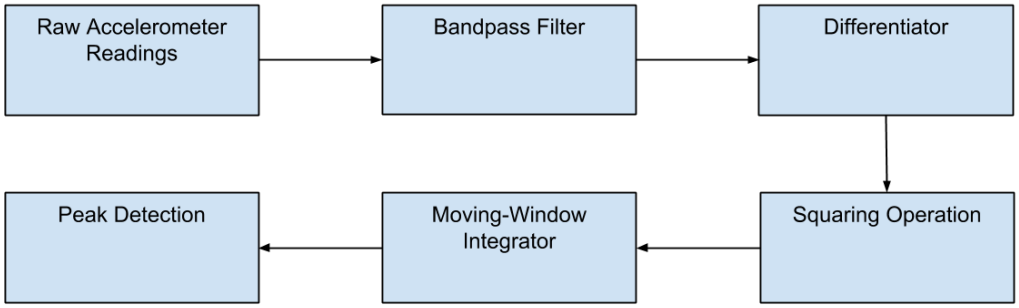


Figure 4.5: Possible Block Diagram of a Pan-Tompkins Algorithm [11]

Figure 4.5 show a possible series of filter used in a Pan-Tompkins algorithm. A bandpass filter is first applied to the signal to cancel out noise. The next process is differentiation, which will obtain information about the slope of a peak. Following the differentiator is squaring, a process that intensify the peaks with high values more than the peaks with small values, and thus helps restrict false peaks. The output of the squaring process may contain gait-cycles with multiple peaks. The moving-window integrator will smoothen this output, making sure that a gait cycle only contain one peak, and thereby only on step. If the four processes described are implemented with coefficients that are suitable for the particular setting, combining them should make it easier for the peak detection search. The peak search procedure can be performed in several ways. The reviewed literature favor an adaptive threshold, which detect steps when a peak is above a certain value.

The implementation of the algorithm is case-dependent and requires tweaking of filter coefficients in order to accomplish satisfying results. Because of this have reviewed literature designed different approaches to the Pan-Tompkins algorithm. The implemented algorithm in this thesis is based on [11], [49] and [50], and use the horizontal acceleration (x-direction).

Bandpass Filter: The bandpass filter is a 6th order Butterworth filter with passband frequency $w1 < \omega < w2$.

Differentiator: The differentiator was tested with both a five-point derivative, as described by the difference Equation 4.4, as well as a two-point

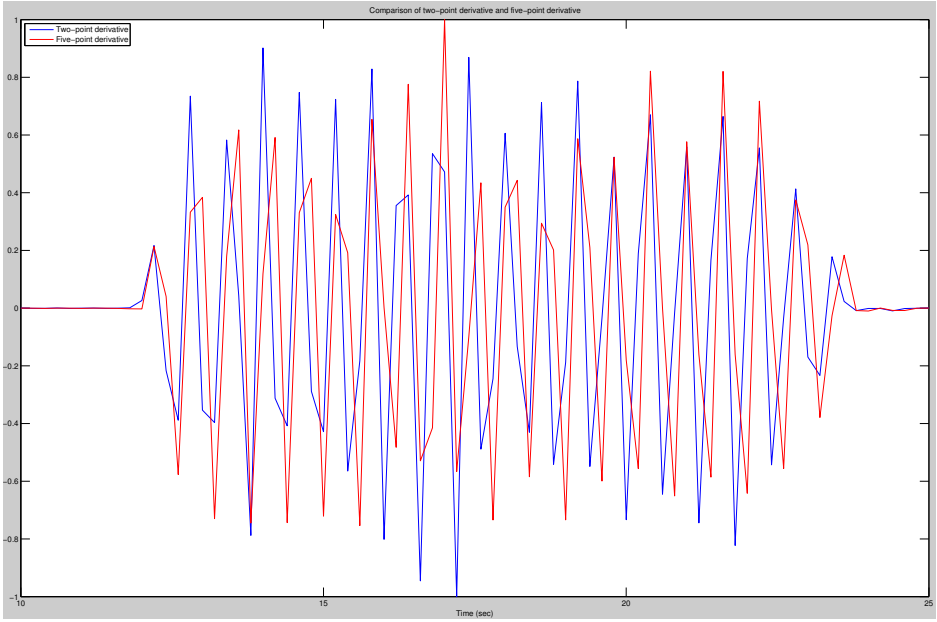


Figure 4.6: Comparison of two-point and five-point derivative

derivative, described in Equation 4.5. A comparison of the two methods is illustrated in Figure 4.6, which shows that both methods results in sufficiently approximations. Giving the advantage of simplicity and shorter computational time the twp-point derivative was chosen.

$$f'(x) \approx \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} \tag{4.4}$$

$$f'(x) \approx \frac{f(x+h) + f(x)}{h} \tag{4.5}$$

Squaring Operation: Squaring is the simple process described in Equation 4.6.

$$y(nT) = |x(nT)|^2 \tag{4.6}$$

Moving-Window Integrator: A mowing-window considers a small "window" of the signal at the time, and as the algorithm proceeds, slides across the signal sample, one time step at the time. The integrator is calculated from Equation 4.7, where N is the number of samples in the width of the

integration window. Choosing the optimize value for N is important. If the window is to wide will several peaks overlap, and merging between a QRS-complex and the T-peak is a typical outcome. If the window is to narrow will the algorithm produce several peaks in one QRS-complex. A rule of thump is choosing N to be approximately the size of the widest QRS-complex in the signal.

$$y(nT) = (\frac{1}{N})[x(nT - (N-1)T + x(nT - (N-2)T + ... + x(nT)] \quad (4.7)$$

Peak Detection: The implemented peak detection procedure is straight forward, counting the number of peaks from the integrator output that are above a static threshold.

Several attempts over a period of weeks was made to make the Pan-Tompkins algorithm detect steps accurately. Considering the sensor setup in this thesis with the one described in the literature it eventually became obvious that the algorithm would not be suitable for the system in this thesis. First of all, Pan-Tompkins is based on detecting a QRS-like signal, which are more identical with foot-mounted IMUs than hip-mounted IMUs. Additionally, the Spatial IMU are much more advanced than most sensor reviewed in the literature, and do therefore not require as much filtering. Nonetheless, Pan-Tompkins was the starting process of this work since it was found in several of the reviewed literature papers. The implementation details can be view in section 7.5.1.

4.4.2 Libby-Peak and the Developed Step Detection Algorithm of this Thesis

In search for a more suitable step detection algorithm than Pan-Tompkins, [42] emerged from the big pile of available step detection algorithms. A winning argument was the similar difficulties with implementing Pan-Tompkins, such unsuitable sensor placement and complex coefficient tuning. The paper further presented an algorithm, from now on named Libby-Peak, for an IMU clipped to the clothing at the hip, and was therefore expected to be suitable for the work in this thesis as well.

In contrast to Pan-Tompkins, uses Libby-peak magnitude of the triaxial accelerometer signal. This is the first of three processes that manipulate ac-

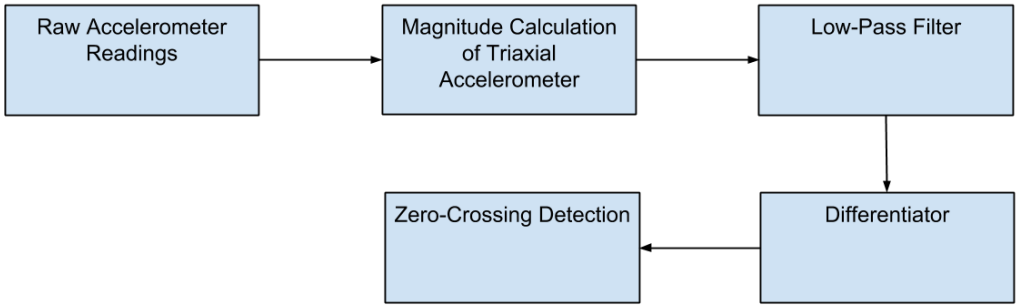


Figure 4.7: Block Diagram of Libby-Peak Algorithm

celerometer signals before detecting steps. Following the magnitude calculation is a low-pass filter, before the derivative of the filter output is approximated. Step detection is then performed based on the assumption that footfalls correlate to local maximum in the magnitude signal, hence a zero-crossing in the derivative signal. This process can be viewed in Figure 4.7.

However, because of the experienced difficulties of implementing suitable filters in Pan-Tompkins, and the realization that Spatial IMU already did most of the filtering job, the cumbersome low-pass filter in Libby-Peak was neglected. However, the idea of using magnitude of a triaxial accelerometer signal was proven to be effective. The same approach was later found in [43], which emphasized the benefit of using magnitude: Magnitude determination is a simple and relatively cheap approach, and the algorithm will not be sensitive to the orientation of the sensors.

The implementation of the algorithm is quite easy. First is the magnitude of the triaxial accelerometer signal Acc calculated, as described by Equation 4.8, where x, y, and z are measured acceleration in the three directions.

$$|Acc| = \sqrt{x^2 + y^2 + z^2} \tag{4.8}$$

A two-point differentiator, as described in Equation 4.5, is thereafter applied to the magnitude signal. The algorithm finishes by calculating the number of zero-crossings from negative to positive, which are assumed to correlate

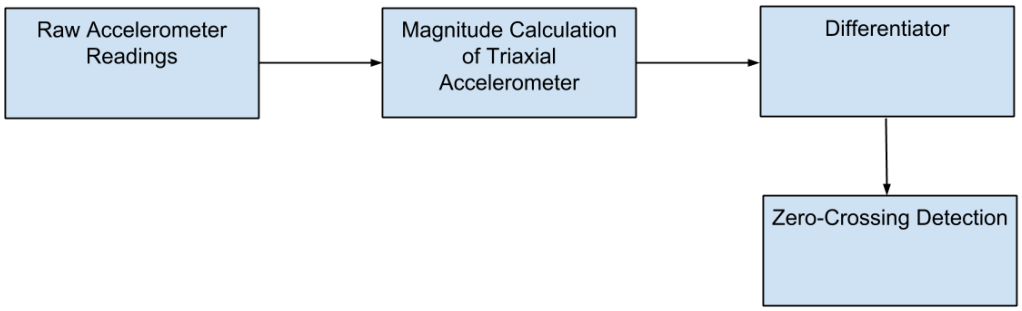


Figure 4.8: Block Diagram of the Libby-Peak Based Algorithm

to footfalls. The process the Libby-Peak based algorithm can be viewed in Figure 4.8 and the implementation details can be view in section 7.5.1.

Chapter 5

Step Length Estimation

Assuming that Chapter 4 successfully counts the number of steps taken by an user wearing a WiLoc node, the next challenge is to estimate the user's step length. This chapter explains the theory behind step length estimation (SLE), and look at related algorithms reviewed in previous works. The chapter ends with an in-depth description of the step length estimation algorithm implemented in this thesis.

5.1 The Peculiarities of Step length Estimation

Step length is defined as a distance between two successive heel impact [51], and depends on several physical parameters, such as weight, muscle strength and leg-length, which may differ between left and right steps. Leg-length is the crucial parameter for step length estimations when a person is walking, but as the step frequency increase and the person begins running will muscle strength and weight influence the estimations as well [17]. Since step length both varies from person to person, and is effected by walking velocity, are real-time step length calibrations necessary in order to accurately calculate traveled distance. Additionally, forward-, backward-, and sideways walking, in addition to factors like ground inclination will also cause step lengths to differ. It may therefore be necessary to use separate step length estimation

functions both for different walking velocity, as well as for different step classes.

5.2 Step Length Estimation Algorithms

Several papers try to classify step length estimation (SLE) algorithms [52], [53], [54], where the latter seem to be consistent with regards to the others.

Firstly, step lengths can be calculated based on characteristics from foot-mounted sensors. As described in Section 4.2, foot-mounted sensors easily detect stance phases, which makes it easy to determine step durations. The ZUPs makes velocity updates very accurate, and step lengths can therefore be estimated based on velocity integrations during the time interval of steps. This method is proven to be very effective and [13] reports of accuracy under 2.5 %. However, the system presented in this thesis uses a hip-mounted sensor, and step length estimation based on foot inertial navigation is therefore not further discussed.

Secondly, step lengths can be calculated based on an inverted pendulum model, also known as biomechanical models. Such approaches use parameters of the vertical excursion of the center of gravity, in addition to leg and foot length. Placing the sensor(s) near the center of gravity, the user's leg are modeled as an inverted pendulum. [55] is a study based on this method. The authors developed a method to determine spatio-temporal gait parameters, such as step length or stride duration, from trunk acceleration data. The inverted pendulum model of the body's center of mass trajectory provides a mathematical relation between trunk acceleration and gait parameters during walking. Unfortunately, their method failed to distinguish between left and right foot, where 12 % of 756 foot contacts were labeled falsely. Additionally, step lengths were underestimated in all subjects and at all speeds. [56] states that inverted pendulum models are prone to underestimating step lengths, and try to correct the error with a modified pendulum model, where all parameters correspond to anthropometric data of the individual. Though traversed distance was estimated between 94.5 % to 106,7 % of the actual traversed distance, can the step length estimation in this thesis not rely on individual data on leg and foot length. It is desirable that the SLE algorithm in this thesis is adaptive and autonomous, and necessary calibration

parameters should therefore be calculated in real-time.

The third class of SLE algorithms express step length as a function of sensor data [17]. Such methods are named parametric models, and use parameters such as walking frequency and variation in accelerometer data, either combined or independently to estimate the step length. [17] apply two different suggestions for such step length functions on a chest-mounted INS. The first suggestion express step length as a function of the difference between the maximum and minimum vertical acceleration during a step. The second suggestion use the variance of the vertical acceleration as step length function. Results proved that the former performed better, with a root-mean-square (RMS) distance error of around 2 %, whereas the latter had a RMS-error of around 3.5 %. The first method described by [17] was also applied in [47], but with use of a hip-mounted sensor. The obtained average relative distance error was 3 to 8 %. It is difficult to state why the results differ, since several factors can effect the system accuracy, such as sensor hardware, sensor placement, step detection, and drift error correction, or the lack of such. [43] determined step length by applying a linear combination of walking frequency and acceleration variance in the step length function. Using optimal parameters and movement status awareness, the authors gained results with estimation error below 1 % for 53.5 % of the data, and an error over 3 % for 20 % of the data.

All the mentioned classes of SLE algorithms are up until now using body-fixed sensors. [57] bring up a fourth class of SLE algorithms; using handheld inertial sensors. The general idea is that smart phones are often already embedded with accelerometers and gyroscopes, which can be used for navigation purposes. Such devices are usually carried in hands, bags, pockets, and other non-body fixed frames, but previous studies [58] have put constrains on the sensor's location, making sure that the device was relatively stable while the user was walking. Therefore, [57] performed an extensive analysis and development of a step length model based on handheld sensor without any constraints. Though the method show surprisingly good result, with an error between 2.5 and 5 % of the traveled distance, will handheld devices be unpractical for the purpose of this thesis.

It should also the mentioned that [53] demonstrate that in most cases is step length not as critical as heading determination. The paper present and compare three typical estimation models for step length estimation, and based on

empirical results and concluding literature, is it argued that choosing different models of step length will not influence the positioning accuracy severely when error from the IMU is eliminated. This statement will be discussed, along with the results of the developed step detection algorithm in Chapter 9.

5.3 Implemented Step Length Estimation Algorithm

The implemented step length estimation algorithm presented in this thesis is based on [43]. Tests performed in this article, as well as other papers, discovered that step length is linear related to both walking frequency and acceleration variance. To verify this statement, Figure 5.1 show the relation between step length and walking frequency for a person wearing the hip-mounted Spatial IMU. It can be seen that step length increases with increasing walking frequency, and in a approximate linear manner. The same is true for acceleration variance. When a person is walking fast will the foot hits the ground with more force than usual, and acceleration measurements will therefore increase, giving a higher amplitude between positive and negative acceleration peaks. Walking Frequency, WF, was calculated using Equation 5.1, where t is the time period of the walk, and n_s is the number of footsteps during that time.

$$WF = \frac{n_s}{t} \tag{5.1}$$

Acceleration variance can be calculated using Equation 5.2, where AV denotes acceleration variance. n and a are the number of samples and acceleration mean during a step, respectively. An illustration of the relation between step length and acceleration variance will not be presented at the moment, since measurement errors make this graph unsuitable as an example of linear relations.

$$AV = \frac{1}{n-1}\sum_{k=1}^{n}(a_k - a)^2 \tag{5.2}$$

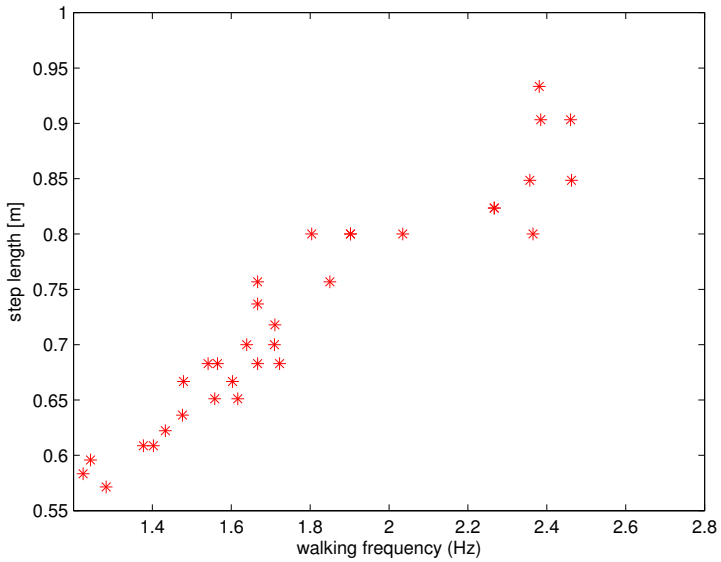


Figure 5.1: The relation between step length and walking frequency

As illustrated in Figure 5.1, can a set of several sensor readings with variation in walking frequency form an approximation of a straight line. The parametric model described in [43] expresses therefore an estimated step length based on the linear equation $y = ax + b$. Equation 5.3 describes the step length estimation $\hat{s}_{wf}$ as a function of walking frequency, while Equation 5.4 describes the step length estimation $\hat{s}_{av}$ as a function of acceleration variance.

$$\hat{s}_{wf} = a_{wf} \cdot WF + b_{wf} \tag{5.3}$$

$$\hat{s}_{av} = a_{af} \cdot AV + b_{av} \tag{5.4}$$

The parametric model uses a linear combination of these estimates, $\hat{s}_{wf}$ and $\hat{s}_{av}$, defined by weight parameters k_{wf} and k_{av}. Equation 5.5 renders the parametric model as stated in the paper.

$$\begin{aligned}\hat{s} &= k_{wf} \cdot \hat{s}_{wf} + k_{av} \cdot \hat{s}_{av} \\ &= k_{wf} \cdot (a_{wf} WF + b_{wf}) + k_{av} \cdot (a_{av} AV + b_{av}) \\ &= (k_{wf} a_{wf}) \cdot WF + (k_{av} a_{av}) \cdot AV + (k_{wf} b_{af} + k_{av} b_{av}) \\ &= \alpha \cdot WF + \beta \cdot AV + \gamma\end{aligned} \tag{5.5}$$

One question still remains. While WF and AV can be calculated based on data from the IMU sensor, α, β, and γ are parameters that must be predefined. Having a set of sensor data, with various walking frequencies and acceleration variance, curve fitting can be used to derive linear equations and their parameters. When fitting a line to a set of data, a definition of good fit must be stated. Residuals are defined as the difference between the real value of a data set, and the fitted value defined in the fitted curve, $F(x)$. Thus is r_i the residual for the data pair (x_i, y_i), giving the vertical distance between the known data point and the fit function $F(x) = \alpha x + \beta$ [59].

$$\begin{aligned}r_i &= y_i - F(x_i) \\ &= y_i - (\alpha x_i + \beta)\end{aligned} \tag{5.6}$$

It seems reasonable that the "best" fit is to minimize the residual sum of square, as defined in Equation 5.7, where m is the number of data points. This approach is known as least square fit, where the best fit is obtained by finding the values of α and β that minimize p.

$$\begin{aligned}p &= \sum r_i^2 \\ &= \sum_{i=1}^{m} [y_i - (\alpha x_i + \beta)]^2\end{aligned} \tag{5.7}$$

Finding α and β so that p is minimized requires

$$\frac{\partial p}{\partial \alpha}|_{\beta=\textbf{constant}} = 0 \tag{5.8}$$

and

$$\frac{\partial p}{\partial \beta}|_{\alpha=\textbf{constant}} = 0 \tag{5.9}$$

Carrying out the differentiation leads to:

$$S_{xx} + S_x\beta = S_{xy} \tag{5.10}$$

$$S_x\alpha + m\beta = S_y \tag{5.11}$$

where

$$S_{xx} = \sum_{i=1}^{m} x_i x_i \qquad S_x = \sum_{i=1}^{m} x_i$$

$$S_{xy} = \sum_{i=1}^{m} x_i y_i \qquad S_y = \sum_{i=1}^{m} y_i$$

Chapter 6

Heading Determination

Since the preceding chapters provided information about steps and step length, the remaining problem of the presented INS is to determine the direction of each step. Commonly used sensors for measuring orientation of an object are tilt-sensors, gyroscopes, and electronic magnetic sensors [60], [61]. The different sensors are used either individually or combined, where the latter creates an integrated system for heading determination, as proposed in [62] and [16]. Because of the time limit of the presented work in thesis are research on integrated systems for heading determination not investigated.

The focus of this chapter is heading determination using gyroscopes and magnetometers since these sensors are available on the Spatial IMU. The first section presents a brief explanation to heading determination using gyroscopes. This method is not implemented as part of the work presented in this thesis. However, the theory on gyroscope based heading determination in addition to its pros and cons, are necessary in order to evaluated the proposed solution, and understand the suggestions on further work. The second section deals with heading determination using magnetometers. Since this method is implemented as part of presented work, is a more in-depth explanation given, including both research on the earth's magnetic field and magnetometer calibration. This chapter end with a description of the implemented heading determination algorithm used in the INS presented in this thesis.

6.1 Heading Determination using Gyroscope

Determining heading using gyroscope measurements is quite intuitive since gyroscopes in fact measure rotation of an object. In theory, integrating data from the gyroscope will thus give an object's angular changes. Given that the integration process was successful can an objects current heading be calculated as expressed in Equation 6.1 [62]:

$$\psi_g(n) = \psi(0) + \sum_{k=1}^{n} \Delta\psi_g(k) \tag{6.1}$$

where $\psi_g(n)$ is the current heading at the time nT, $\psi(0)$ is the initial heading of the object, and $\Delta\psi_g(k)$ is the angular changes that the object has experienced before time nT. The initial heading must be defined since the gyroscope only measures a person's relative heading. Examples for determining an initial heading include using an external positioning method or a compass which provides heading information.

There are two specifically good advantages of heading determination using gyroscopes. Firstly, gyroscopes give good heading performance on quick turns. Secondly, compared to other heading determination methods are gyroscopes not subject to external disturbances. In addition is installation of a gyroscope relatively simple, and as new MEMS sensors are becoming cheaper and more accurate, are gyroscopes becoming a more and more attractive alternative for many INS applications [62].

On the other hand, the theoretical case on gyroscope data integration is not realistic in real-life. The sensor readings from the IMU's gyroscope are discrete, inherently noisy, and are subject to bias drifts [**?**]. Integration will thus lead to exponential error for longer time intervals, making the gyroscope unsuitable for long-term applications.

6.2 Heading Determination using Magnetometer

This section explains the general theory behind heading determination using magnetometers. Required information on the Earth's magnetic field are provided before heading determination of a pedestrian is explained. Suggestions for magnetometer calibrations are then presented in order to increase heading accuracy.

6.2.1 Earth's Magnetic Field

The Earth's magnetic field is similar to that of a dipole bar magnet, where the magnetic field points from the magnetic south pole and to its the magnetic north pole [**?**]. The geographical poles lies on the Earth's spin axis, and are referred to as true north and true south. The magnetic poles on the other hand do not coincide with the spin axis, but are tilted 11.5 degrees as illustrated in Figure 6.1. Hence, the true magnetic north pole lies north of Canada, but is slowly changing due to motion in the Earth's inner core. The angle between true north and magnetic north is called magnetic declination, which varies all over the surface on Earth [**?**]. The declination angle is positive in eastward direction, while considered negative in westward direction.

Depending on an object's location will the magnetic field on Earth vary both in direction and strength. The magnitude of the filed ranges from about 0.3 to 0.6 gauss (G) at the Earth's surface, where the strength at the magnetic poles are twice the size of the intensity near equator [**?**]. The magnetic field line will always point towards magnetic north, as seen in Figure 6.1, where they always will point up from the surface on the southern hemisphere, and down towards the surface on the northern hemisphere, a characteristic called magnetic inclination. Inclination is defines as the angel between the horizontal plane and the direction of the total field, where inclination at magnetic equator is 0 degrees, and on the magnetic poles 90 degrees [**?**]. Information about inclination and declination angles for any location on Earth can be found in [**?**].

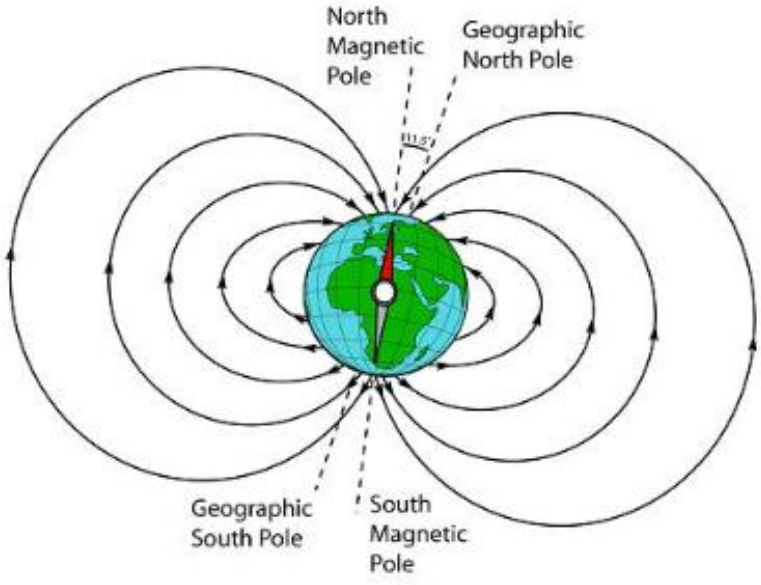


Figure 6.1: The Earth's magnetic field [**?**]

6.2.2 Determining Heading of a Pedestrian

As described in Chapter 3, the Spatial IMU contains three orthogonally mounted magnetometers which measure the Earth's magnetic field. Thus, each magnetometer senses the component of the magnetic field parallel to its axis. Using all three magnetometers will therefore provide three-dimensional space information about strength and direction of the magnetic field in which the sensors are applied. However, to determine magnetic heading is only the magnetic field's horizontal components of interest for the method described in this section. The vertical portion of the earth's magnetic fields is ignored.

An example of the ideal output of a horizontal magnetic field can be viewed in Figure 6.2. In this example was Spatial IMU exclusively rotated around its z-axis, which is the rotation of interest when determining heading of a person wearing the IMU. It can be seen that the magnetic field appear as an approximately complete circle, centered at the origin and with a size proportional to the strength of the field. To determine the heading of an walking pedestrian is basic triangle trigonometry applied to the horizontal

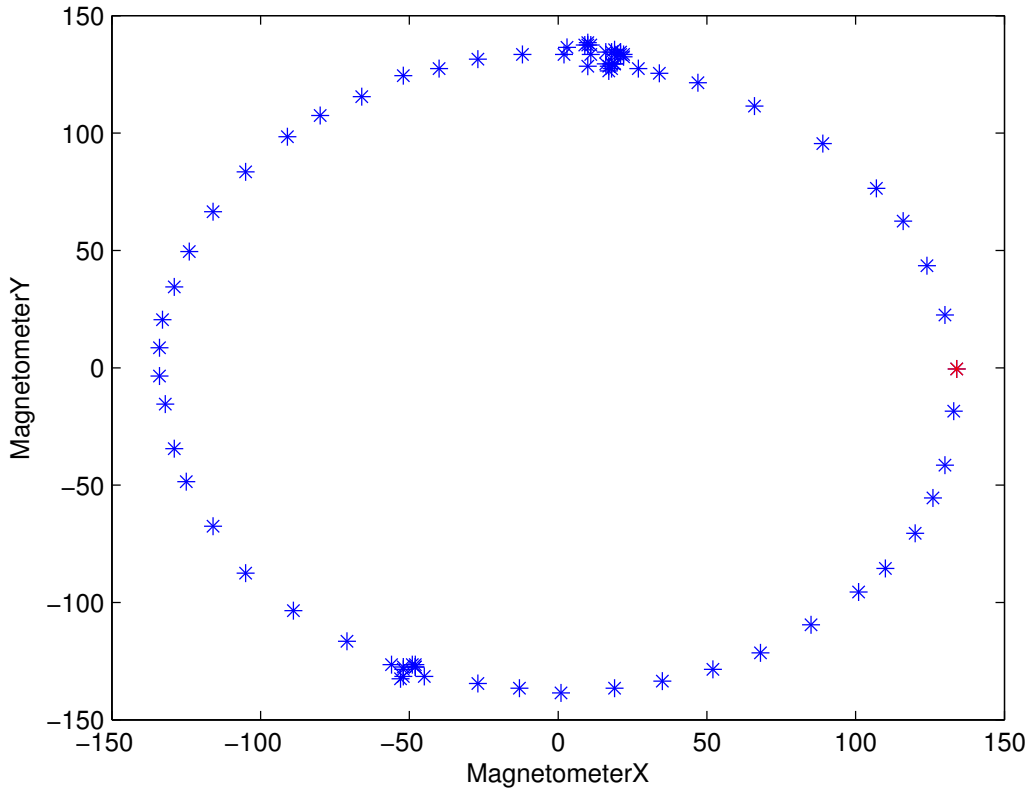


Figure 6.2: Magnetometer readings from the Spatial IMU while rotating around its z-axis

magnetometer readings, as shown in Equation 6.2, where $H(M_x, M_y)$ is the heading and (M_x, M_y) are the magnetic readings in the horizontal plane [53].

$$H(M_x, M_y) = \arctan(-\frac{M_y}{M_x}) \tag{6.2}$$

By defining an initial heading, as marked in red in Figure 6.2, an objects rotation from an initial position can be determined. Consequently, this method for heading determination is dependent on knowledge about the magnetic field where the sensor are applied in order to make an approximation to an ideal circle with center at the origin and with a size that represents the magnetic field where the sensor are applied.

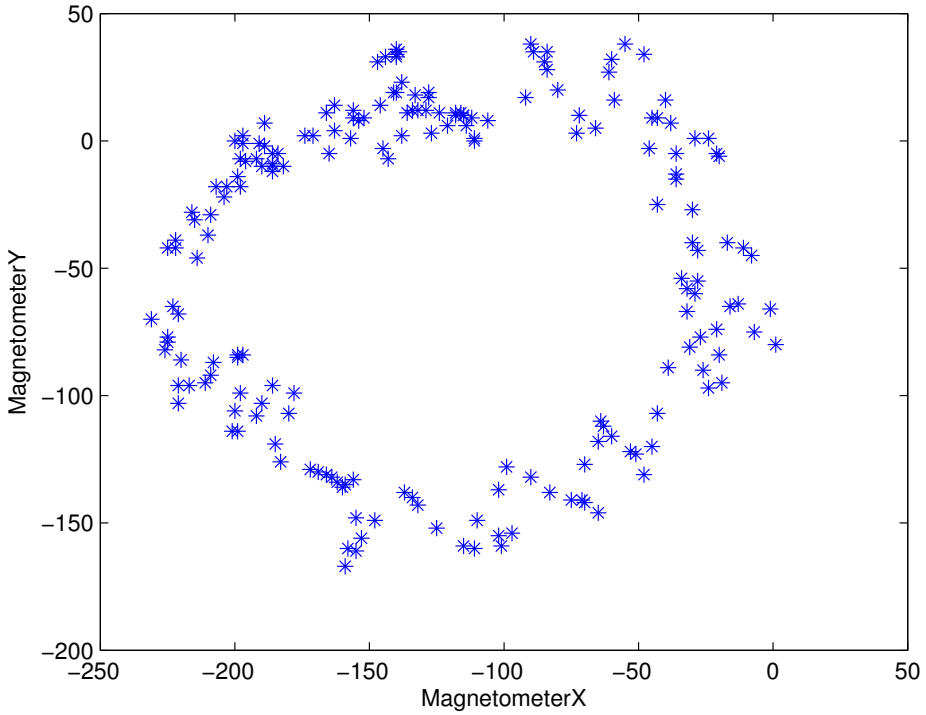


Figure 6.3: Magnetometer readings from the Spatial IMU while object turning around its z-axis

Figure 6.3 illustrate how the horizontal plane of the magnetic field may appear when an object wearing the hip-mounted Spatial IMU is turning 360 degrees. Outlines of a circle can be seen, but with deviations compared to the ideal example. In addition is the circle not centered at the origin and heading can therefore not be calculated using Equation 6.2. These deviations illustrate the fact that sensor readings from the magnetometers are not only affected by the Earth's magnetic field. Predictable and unpredictable error sources effect the magnetometers as well [53], causing the ideal circle to deform. In order to get an accurate heading for a walking pedestrian must the magnetometers be calibrated to remove such error sources. The next section explains some common approaches for magnetometer calibration.

6.2.3 Magnetometer Calibration

As mentioned, magnetic fields besides that of the earth can also impact magnetometer readings. Error sources include tilt of sensor platform, hard and soft iron interferences, installation misalignment, magnetic declination, in addition to unpredictable errors from environmental magnetic disturbances [53]. Most errors can be eliminated by a calibration process or a filter method, such as Kalman filter. Unpredictable errors are harder to remove. In this section are hard- and soft iron corrections explained. Additional research on error sources for magnetometers and associated magnetometer calibration can be conducted in papers such as [**?**], [**?**], [61], [53] and [**?**].

Hard-iron interference refers to the presence of permanent magnets around the magnetometers [**?**]. If a magnet is fixed in location and orientation with respect to the magnetometers, will the hard-iron influence on the magnetometers' axes not change when an object wearing the IMU is moving. Hard-iron interference produce therefore an constant offset from the center at $(0, 0)$ for the ideal circle in Figure 6.3 [**?**]. Compensating for hard-iron distortions is simple. The offsets are calculated based on measurements from an object turning at least 360 degrees. The distance from $(0, 0)$ to the center of the measured circle are then determined as described in Equation 6.3 [**?**]. After the offsets are calculated are these values added to the measured data, removing the hard-iron interference.

$$\begin{aligned} \alpha &= \frac{x_{max} - x_{min}}{2} \\ \beta &= \frac{y_{max} - y_{min}}{2} \end{aligned} \tag{6.3}$$

Soft-iron interference refers to distortions caused by magnetically absorbent objects around the magnetometers. Such objects are not magnetic in themselves, but attracts and distorts the magnetic field [**?**]. Figure 6.4 illustrates this effect. An uniform magnetic field is represented by vertical lines. When an U-shaped piece of steel is applied to this magnetic field will it cause an distortion as shown in A. Rotating the piece of steel 180 degrees, as seen in B, will give a similar distortion as the one in A, only rotated 180 degrees. C illustrate the similarity of A and B, where B is flipped and rotated [**?**]. This illustrate the fact that while hard-iron distortion increase or decrease the

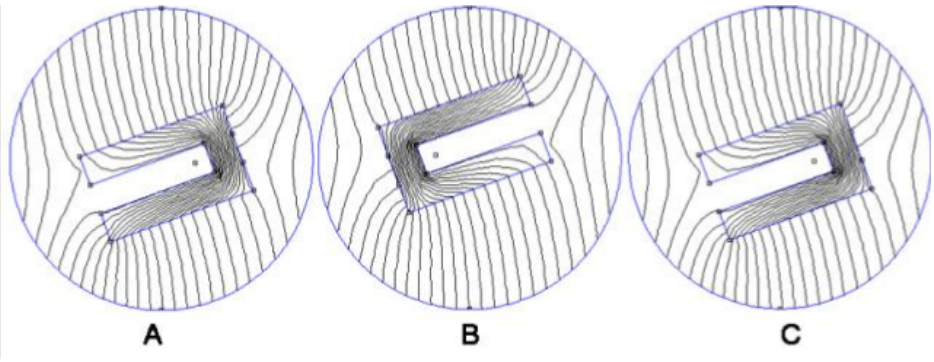


Figure 6.4: Soft Iron Distortion in a Uniform Field [**?**]

strength of the magnetic field regardless of an object's heading, will soft-iron distortions only change the direction of the field when an object is rotating.

Soft iron distortion typically cause the ideal circle to deform into an ellipse [**?**], and scaling must therefore be performed on both axes to convert the ellipse to an circle. [**?**] suggested the following approach to compensate for both the hard- and soft-iron distortions. In order to find the scaling factors, X_{sf} and Y_{sf} are the maximum and minimum values of the horizontal components of the magnetometers determined, as expressed in Equation 6.4.

$$\begin{aligned} X_{sf} &= max1, \frac{y_{max} - y_{min}}{x_{max} - x_{min}} \\ Y_{sf} &= max1, \frac{x_{max} - x_{min}}{y_{max} - y_{min}} \end{aligned} \tag{6.4}$$

$$\begin{aligned} X_{off} &= [\frac{x_{max} - x_{min}}{2} - x_{max}] \cdot X_{sf} \\ Y_{off} &= [\frac{y_{max} - y_{min}}{2} - y_{max}] \cdot Y_{sf} \end{aligned} \tag{6.5}$$

Defining the offset values X_{off} and Y_{off} as seen in Equation 6.5, the calibrated magnetometer readings be found as:

$$\begin{aligned} X_{cal} &= X_{sf} \cdot m_x + X_{off} \\ Y_{cal} &= Y_{sf} \cdot m_y + Y_{off} \end{aligned} \tag{6.6}$$

Though this section only include theory on hard- and soft-iron interference, will several other sources of error decrease accuracy of a heading determination algorithm. Therefore, it is recommended to extended the theory given in this section with research on the following list of error sources for converting magnetometer readings to a certain heading [**?**]:

- Orientation of sensor axes relative to the horizontal plane
- Hard- and soft-iron interference
- Magnetic declination
- Earth's magnetic field direction is not perfectly parallel to level ground
- Orientation of sensor axes relative to horizontal plane changes from moment to moment as the object walks
- Magnetic declination and inclination may change if the object is moving a long distance
- Magnetic declination and inclination changes significantly over time

6.3 Implemented Heading Determination Algorithm

The implemented heading determination algorithm in the work presented in this thesis is based on the theory given in the previous section, and in particular equations 6.2, 6.3 and 6.6. In order to define the magnetic field where the IMU is applied, is a calibration process needed. This calibration is known as spot turn, and involves a complete 360 degrees spin of the user wearing the IMU. This technique is convenient, but suffer from few calibration points. [34] propose three other calibration techniques, in addition to spot turn. These techniques can be seen in Table 6.1, but only spot turn was performed in the INS presented in this thesis.

Compared to heading determination using gyroscope where integrating error increase, magnetometers provide long term stable accuracy. On the other hand give magnetometers poor performance on quick turns and can easily be corrupted by surrounding magnetic disturbance. In fact, advantages of

Method	**Description**	**Advantage**	**Disadvantage**
Spot turn	User spins 360 °	Convenient	Few calibration points
Full Circle	User walks around a circle of approximately 4 m diameter	Calibrated in pedestrian mode	Requires open level calibration area, more time consuming
GPS Calibration	Compass calibrated by GPS heading	Convenient, continuous	Requires GPS, user must be moving
Gyroscope Correction	Measure relative heading change	removes effect of short term field disturbances	Added sensor expense. Low cost gyros have substantial drift

Table 6.1: Comparison of Magnetometer Calibration Techniques [34]

magnetometers are disadvantages of gyroscope, and vice versa. However, due to the time limit of the presented work in this thesis was gyroscopes not used for determining heading. Further investigation on heading determination by integrating the gyroscopes with the magnetometers can be conducted in [16].

Generally, the problem of heading determination is how to relate a quantity measured by the sensor to a geographical location. In INS will sensor readings be given in a sensor frame, where the frame is moving and rotating with the sensor. Further, an user may be moving and rotating in a completely different frame, called the user frame, while the geographical location of an user must be defined in a global frame of reference, where the geographic directions north, east, west and south is defined. For the work presented in this thesis is the global frame not of interest since RFID, Bluetooth, GPS or other positioning methods are assumed to provide information about a starting position. The IMU should therefore only perform navigation inside an already defined geographic area. This makes the challenge of heading determination somewhat easier. However, a method for relating magnetometer readings in the sensor frame to a global frame of reference is presented in [14], and is recommend for those interested in INS using only IMU data.

Chapter 7

Implementation

It it time to "wrap up" the main points from the preceding chapters. The INS presented in this thesis does not use integration of acceleration to obtain position estimated, but rather look at step detection, step length estimation and heading determination. This is achieved with an IMU manufactured at Advanced Navigation, the Spatial V3. Using accelerometer readings from this IMU, step detection estimates are calculated in two different algorithms, the Pan-Tompkins algorithm and a Libby-Peak based algorithm. Further, step length estimations are made based on a linear model of walking frequency and acceleration variance, using predefined parameters. The INS is completed by heading determination based on magnetometer readings from the IMU, where hard- and soft-iron interference are compensated for.

This chapter provides a detailed presentation of the suggested solution for a prototype INS. Unfortunately was none of the external positioning methods from Chapter 2 implemented because of the time limits of this work, but when needed in the implemented algorithms is it assumed that an initial position is given by an external positioning method.

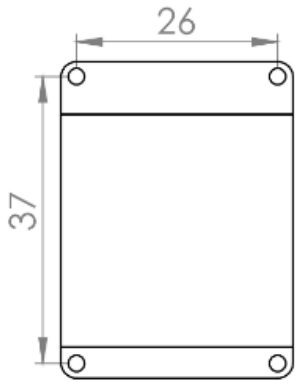


Figure 7.1: Illustration of Spatial Mounting Plate

7.1 Equipment and Requirements

In addition to the Spatial V3 IMU was a few additional items needed. First, in order to mount the IMU on an users's hip, the sensor was sewed on a belt using the holes on the mounting plate, as illustrated in Figure 7.1.

As described in Chapter 3, the IMU sends sensor data to Spatial Manager via an USB interface cable. Thus, a computer with an available USB port is required. In this work was a Dell XPS 12 used, booted with both Ubuntu 13.01 and Windows 7. In order to gain contact with the sensor must Java be up to date, in addition to the latest FTDI driver, as explained in reference guide [8]. The reference guide makes it very easy to have things up and running. The only encountered problem was when running the Spatial Manager from Ubuntu, where it was found necessary to give serial port permission. This was achieved with the following command: *sudo chmod 666 /dev/TTYUSB0*.

Matlab was used to import data and implement the suggested INS since it contains many built-in algorithms and functions for signal processing.

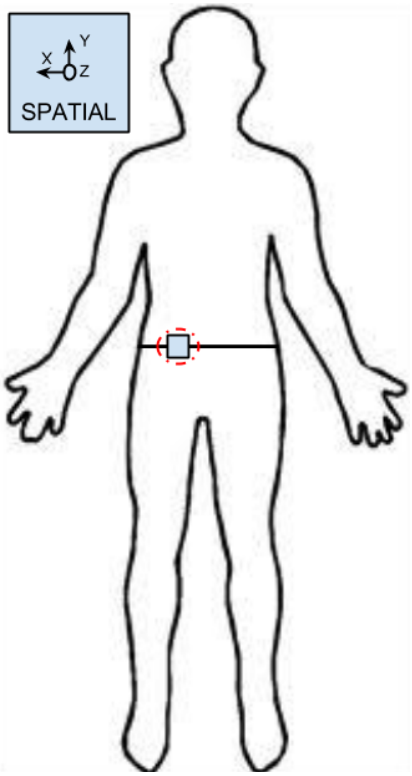


Figure 7.2: Position and Alignment of Spatial

7.2 Alignment

For the easiest alignment, Spatial should be placed with the sensor axes aligned with the vehicle axes [8], having the x-axes pointing forward towards the front of the vehicle and the z-axes pointing downwards. For alignment in this application, Spatial is placed on the hip, and recommended alignment was therefore not possible. As illustrated in Figure 7.2, the x-axis is pointing left in horizontal direction, y-axis points up, and z-axis is pointing backwards.

Since Spatial have a different alignment than recommended, it introduces an alignment offset which should be configured using the Spatial Manager software in the following way: Run Spatial Manager, open Alignment in the Configuration menu. Pushing the "Zero Current Orientation" while standing in correct zero position will configure the alignment offset. This calibration was executed every time Spatial Manager started running.

7.3 Logging

Since this thesis presents a prototype system is the positioning estimations not performed in real-time. Fortunately, Spatial Manager features a fully automatic logging system. Spatial Manager starts a new log every time the serial port button "Connect" in the main view is clicked. The log is saved in the same folder as *SpatialManager.jar* when the "Disconnect" button is clicked, and is given the file name *SpatialLog_date_time.anpp*, which contain all raw data received from Spatial in the AN packet protocol, which is defined by Advanced Navigation. In order to convert these files to readable formats, a log converter in Spatial Manager is used. By running the Spatial Manager and opening "Log Converter" in the tools menu, a spatial log file can be selected, converted and saved. The log converter generates CSV (Comma Separated Values) files that can easily be accessed by Microsoft Excel, Matlab and other data analysis programs. Additionally, if the GNSS antenna was connected during logging will a GPX file of position be created, which is designed to be opened with Google Earth.

7.4 Sensor Ranges

One last setting was changed before the system was up and running. By default Spatial comes configured in the lowest sensor ranges. In this configuration it is possible to send the gyroscopes over range by quickly rotating the IMU. Thus, the sensor ranges was set to the following in order for the INS to work properly:

- Accelerometer range: 16 g
- Gyroscope range: 2000 degree/s
- Magnetomer range: 8 Gauss

This was achieved by running the Spatial Manager, and opening Sensor Ranges in the Configuration menu.

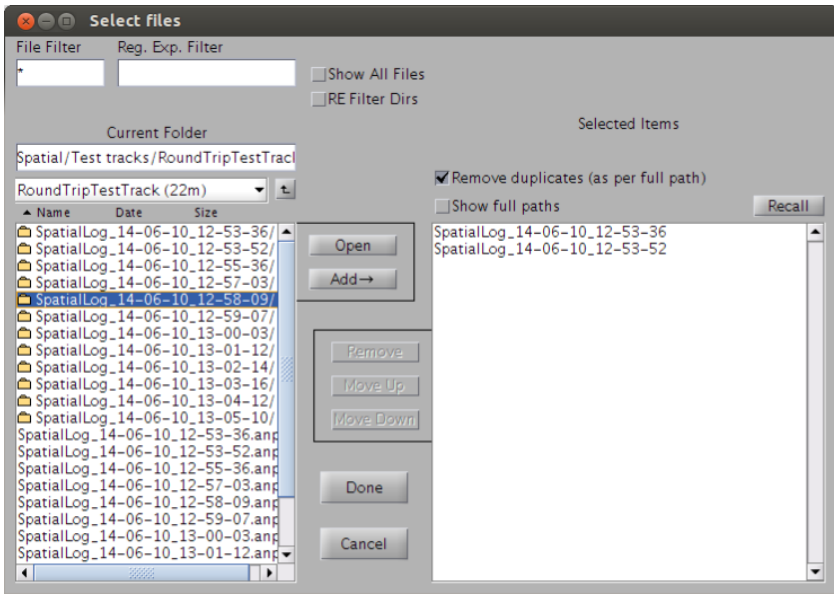


Figure 7.3: Uipickfiles: GUI-application that allows for running several log files in one operation

7.5 Suggested Inertial Navigation System

The suggested prototype INS works in the following way. An user sets the sensor ranges and zero current orientation as described in sections 7.2 and 7.4, before doing a spot turn, as described in section 6.3. The system is then ready for logging, and the user wearing the hip-mounted IMU can reconnect the Spatial Manager and start moving. The user can the entire time while moving use Spatial Manager for checking sensor status and looking at the raw sensor data. After a test track have been conducted, the raw data from the Spatial IMU is converted to a CSV file using the logging system, as described in Section 7.3. The sensor data can then be analyzed using the implemented Matlab code.

Running the Matlab code *main.m*, will initiate "uipickfiles.m", a GUI-application that allows multiple files or directories to be selected [**?**]. This application is illustrated in Figure 7.3, and allows for running several log files in one operation. It is easy to select and remove the log files of interest. In this example have two log files already been selected.

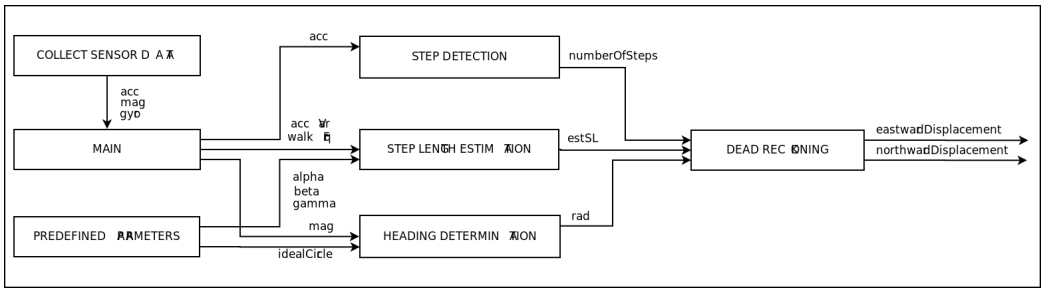


Figure 7.4: FlowChart of Matlab code

For each selected log file in "uipickfiles" will *main.m* open the selected log file and collect measured data from the sensors in the IMU. When the data is imported will the selected step detection algorithm be initiated by running the script *stepDetection.m*, using accelerometer data from the sensor. After step detection are step length estimation performed in *stepLength.m*. This requires predefined values for the linear model. The heading determination in *headingDetermination.m* uses the spot-turn calibration to determine the variables of the ideal circle. The magnetometer data are then scaled to fit inside this circle. The last script initiated by main is *deadReckoning.m*, which uses the number of steps obtained by the step detection algorithm, the step length estimations and a value given in radianer from the heading determination. The matlab code's flow chart can be viewed in Figure 7.4.

7.5.1 Step Detection Algorithms

Pan-Tompkins Algorithm

The implementation of Pan-Tompkins Algorithm is based on the theory outlined in Section 4.4.1. After assembling sensor data, the Pan-Tompkins Algorithm uses $AccelerometerX$ as input to a Bandpass filter with low frequency cutoff ω_1 set to 0.5 Hz, and an upper cutoff frequency ω_2 set to 6 Hz. The cutoff frequency was set based on the assumption that a person never will be able to walk at a frequency higher than 6 Hz, ie six steps per second. The bandpass filter is implemented in Matlab using the Butterworth filter design described in [63] and the Matlab code can be seen in Listing 7.1.

Listing 7.1: Bandpass Filter

```
%% BANDPASS FILTER
f1=0.5; % cuttoff low frequency to get rid of baseline wander
f2=6; % cuttoff frequency to discard high frequency noise
Wn=[f1 f2]*2/30; % cutt off based on fs
N = 3; % order of 3 less processing
[a,b] = butter(N,Wn); % bandpass filtering
highPass = filtfilt(a,b,chosenAccelerometer);
highPass = highPass/ max( abs(highPass));
```

The Differentiator and Squaring Operation are straight forward, and can be viewed in Listings 7.2 and 7.3, respectively.

Listing 7.2: Differentiator

```
%% DERIVATIVE
%Two-point differentiation
derivative = diff(highPass);
derivative = [derivative; zeros(1,1)];
derivative = derivative / max(abs(derivative));
```

Listing 7.3: Squaring Operation

```
%% SQUARING FUNCTION
squaring = derivative.^2;
```

Listing 7.4 show the Moving-Window Integrator, which uses convolution between the output from the Squaring Operation and a sample of data point defined by N. N is empirical chosen based on the assumption that is should be approximately the number of time stamps of the widest peak in the signal.

Listing 7.4: Moving-Window Integrator

```
%% MOVING-WINDOW INTEGRATION
N = 5;
movingWindow = conv(squaring,ones(1,N)/N);
movingWindow(numberOfsamples+1:end) = [];
```

Peak detection is simply performed by counting the number of peaks from the output of Moving-Window Integrator over an empirically chosen threshold.

In addition, peak detection is also executed on the output from the squaring operation. The reason for this is that the Moving-Window Integrator is extremely dependent on correct parameter tuning, and since the algorithm is static, it was found reasonable to check if possibly bad step detections were made because of poor estimates from the moving-Window Integrator or the algorithm as a whole. An example of peak detection can be seen in Listing 7.5.

Listing 7.5: Peak Detection

```
%Find signal peaks - peaks under a threshold value are
    considered as noise.
[movingWindowPks, mwPeakLocation] = findpeaks(movingWindow, '
    MINPEAKHEIGHT', 0.1);

if isempty(movingWindowPks)
    text = 'MovingWindow: No detected peaks'
else
    numberOfStepsMovingWindow=length(movingWindowPks);
```

Libby-Peak based Algorithm

The implementation of the Libby-Peak based Algorithm is based on the theory outlined in Section 4.4.2. The Matlab code for magnitude calculation from the triaxial accelerometer readings can be seen in Listing 7.6.

Listing 7.6: Magnitude Calculation

```
%% Calculate Magnitude of Triaxial Accelrometer
magnitude = sqrt(accelerometerX.*accelerometerX +
    accelerometerY.*accelerometerY + accelerometerZ.*
    accelerometerZ);
magnitude = magnitude - magnitude(1); % starting at zero
```

The implemented Differentiator is the same as implemented for Pan-Tompkins and therefore be seen in Listing 7.2. After the differentiator is zero-crossing executed, where a step is counted for every crossing from negative to positive. The Matlab code can be seen in Listing 7.7.

Listing 7.7: Zero-Crossing Detection

```
%% Zero-crossing
% Find signal peaks.
% - peaks under a threshold value are considered as noise.
derivativePks = findpeaks(derivative, 'MINPEAKHEIGHT',
    thresholdValue);

numberOfStepsDer = 0;

if isempty(derivativePks)
    text = 'Differentiator: No detected peaks'
else
    timeOfStepsDer = zeros(1, numberOfsamples);
    zeroCrossing = zeros(1,numberOfsamples);

    % Count the number of steps, where a step is defines as a
        zero-crossing from negative to positive:
    signalLessThanZero = false;
    lessThanZeroPoint = 0;
    peakCounter = 1;
    plottingCounter = 1;

    for i=1:numberOfsamples

        % Find min peak, before a max peak
        if derivative(i) < -thresholdValue
            signalLessThanZero=true;
            lessThanZeroPoint = i;
        end

        if derivative(i) == derivativePks(peakCounter)

            % Adjust pekCounter
            if(peakCounter==length(derivativePks))
                peakCounter = length(derivativePks);
            else
                peakCounter = peakCounter+1;
            end

            %Peak after crossing zero from negative to
                positive
            if signalLessThanZero == true
                signalLessThanZero = false;
                numberOfStepsDer = numberOfStepsDer +1;

```

```
42                  % Find time stamp of zero-crossing, using
                        interpolation
43                  samples = i - lessThanZeroPoint+1;
44
45                  xi = zeros(1,samples);
46                  yi = zeros(1, samples);
47                  counter = 1;
48
49                  for k=1:samples
50                      if k==1 && derivative(k+lessThanZeroPoint
                            -1)>=derivative(k+lessThanZeroPoint)
51                          %ikke lagre
52                      elseif k==samples && derivative(k+
                            lessThanZeroPoint-1) <= derivative(k+
                            lessThanZeroPoint-2)
53                          % ikke lagre
54                      elseif ne(k,1) && ne(k,samples) && (ge(
                            derivative(k+lessThanZeroPoint-1),
                            derivative(k+lessThanZeroPoint)) || le
                            (derivative(k+lessThanZeroPoint-1),
                            derivative(k+lessThanZeroPoint-2)))
55                          % ikke lagre
56                      else
57                          xi(counter) = second(k+
                                lessThanZeroPoint-1);
58                          yi(counter) = derivative(k+
                                lessThanZeroPoint-1);
59                          counter = counter +1;
60                      end
61                  end
62                  xi(counter:end) = [];
63                  yi(counter:end) = [];
64
65                  zeroCrossingX = interp1(yi,xi, 0);
66                  zeroCrossingY = interp1(xi,yi, zeroCrossingX);
67                  timeOfStepsDer(plottingCounter) =
                        zeroCrossingX;
68                  zeroCrossing(plottingCounter) = zeroCrossingY;
```

Since Spatial delivers already filtered data was peak direction performed directly on the magnitude signal. For this reason was peak detection performed on the magnitude signal, similar to the approach in Listing 7.5.

7.5.2 Step Length Estimation

The implementation of the developed SLE algorithm is based on the theory outlined in Section 5.3. It was first necessary to calculate the predefined parameters α, β, and *gamma*. In order to do so must a data set containing real step length, walking frequency and acceleration variance be obtained. 35 tests of a subject walking down a straight line with different walking velocity were used, where the walking frequency and acceleration variance of each test was calculated using the Matlab code presented in Listing 7.8.

Listing 7.8: Calculation of time period t and walking frequency WF

```
% Calculate acceleration variance and walking
    frequency
accelerationVariance(iFile) = var(magnitude);
testTrackWalkingFrequency(iFile) = numberOfStepsDer/
    timeOfWalking;
```

When having a fully operational data set of real steps, walking frequency and acceleration variance for several test tracks, was curve fitting used to find α, β, and *gamma*. Using the Matlab function "$polyfit(, x, y, N)$", the slope and intercept of the polynomial $P(x)$ of degree N that fits the data y best in a lest-squares sense is returned. Listing 7.9 show how curve fitting for step length as a function of walking frequency is predicted, with $N = 1$, providing a linear model.

Listing 7.9: Curve Fitting

```
%% CURVE FITTING - WALKING FREQUENCY
% 1: Use polyfit to compute a linear regression that predicts
    y from x:
fittingWF = polyfit(testTrackWalkingFrequency, realSL, 1);
slopeWF = fittingWF(1)
interceptWF = fittingWF(2)
```

Listing 7.10 further check how good the returned fit is by calculating the square sum of all residuals, as described in Section 5.3, by Equations 5.10 and 5.11.

Listing 7.10: Calculation of residuals

```
% 2: Call polyval to use p to predict y
yfitWF = polyval(fittingWF,testTrackWalkingFrequency);
% yfitWF = fittingWF(1)*x + fittingWF(2)

% 3: Compute the residual values as a vector signed numbers:
yresidWF = realSL - yfitWF;

% 4: Square the residuals and total them obtain the residual
    sum of squares:
SSresidWF = sum(yresidWF.^2);

% 5: Compute the total sum of squares of y by multiplying the
    variance of y by the number of observations minus 1:
SStotalWF = (length(realSL)-1)* var(realSL);

% 6: Compute R2 using the formula given in the introduction of
     this topic:
rsqWF = 1 - SSresidWF/SStotalWF
```

$rsqWF$ returns a value between zero and one, and gives an indications on how the linear equation $slopwWF \cdot x + interceptWF$ predicts the variance in the variable y. When the slope and intercept of both Equation 5.3 and 5.4 are predicted, can α, β, and $gamma$ be defined as shown in Listing 7.11, and estimated step length is then calculated using Equation 5.5.

Listing 7.11: Step Length Estimation

```
%% Calculate Step Length
alpha = k_wf*slopeWF;
beta = k_av*slopeAV;
gamma = k_wf*interceptWF + k_av*interceptAV;

estSL(iFile) = alpha*testTrackWalkingFrequency(iFile) + beta*
    accelerationVariance(iFile) + gamma;
```

k_{wf} and k_{av} defines the weighting of walking frequency and acceleration variance.

7.5.3 Heading Determination

The implementation of the heading determination algorithm is based on the theory outlined in Section 6.3. Prior to heading determination was three different approaches for magnetometer calibration implemented. First, removing hard-iron distortions was implemented as seen in Listing 7.12.

Listing 7.12: Magnetometer Calibration: Removing Hard-Iron Distortions

```
% Remove hard-iron distortions
offsetX(iFile) = min(magnetometerX) + range(magnetometerX)/2;
offsetY(iFile) = min(magnetometerY) + range(magnetometerY)/2;
magnetometerXcircle = magnetometerX - offsetX(iFile);
magnetometerYcircle = magnetometerY - offsetY(iFile);
```

[**?**] suggested an approach for compensating for both hard- and soft-iron distortions, as given in Equation 6.6. This method was also implemented and can be seen in Listing 7.13.

Listing 7.13: Magnetometer Calibration: Removing Hard- and Soft-Iron Distortions

```
% Remove hard- and soft-iron distortions
scaleFactorX = max(1, (max(magnetometerY)-min(magnetometerY))
    /(max(magnetometerX)-min(magnetometerX)) )
scaleFactorY = max(1, (max(magnetometerX)-min(magnetometerX))
    /(max(magnetometerY)-min(magnetometerY)) )
xOff = (((max(magnetometerX)-min(magnetometerX))/2)-max(
    magnetometerX))*scaleFactorX
yOff = (((max(magnetometerY)-min(magnetometerY))/2)-max(
    magnetometerY))*scaleFactorY

magnetometerXcircle = scaleFactorX*magnetometerX + xOff;
magnetometerYcircle = scaleFactorY*magnetometerY + yOff;
```

The final magnetometer calibration was the spot-torn calibration, as described in Section 6.3, which can be viewed in Listing 7.14.

Listing 7.14: Magnetometer Calibration: Spot-turn

```
% Spot turn
offsetX(iFile) = min(magnetometerX) + range(magnetometerX)/2
offsetY(iFile) = min(magnetometerY) + range(magnetometerY)/2
magnetometerXcircle = magnetometerX - offsetX(iFile);
magnetometerYcircle = magnetometerY - offsetY(iFile);

idealCircleX = 130;
idealCircleY = 130.5;
scaleFactorX = idealCircleX/max(magnetometerXcircle);
scaleFactorY = idealCircleY/max(magnetometerYcircle);
magnetometerXcircle = magnetometerXcircle*scaleFactorX;
magnetometerYcircle = magnetometerYcircle*scaleFactorY;
```

Using one of the listed methods for magnetometer calibration was heading determined based on the implementation given in Listing 7.15

Listing 7.15: Heading Determination

```
for i=1:numberOfsamples
    if magnetometerXcircle(i) > 0 && magnetometerYcircle(i) >
        0
        radianer(i) = atan(magnetometerYcircle(i)/
            magnetometerXcircle(i));
    elseif magnetometerXcircle(i) > 0 && magnetometerYcircle(i
        ) < 0
        radianer(i) = atan(magnetometerYcircle(i)/
            magnetometerXcircle(i))+ 2*pi;
    else
        radianer(i) = atan(magnetometerYcircle(i)/
            magnetometerXcircle(i)) + pi;
    end
end
```

7.5.4 Dead Reckoning

The final implementation of the presented INS was DR. The implementation was based on the equations given for read reckoning in Section 3.3, and can be seen in Listing 7.16.

Listing 7.16: Heading Determination

```
if numberOfStepsDer == 0
  text = 'Object not moving'
else
    eastwardDisplacement = zeros(numberOfStepsDer,1);
    northwardDisplacement = zeros(numberOfStepsDer,1);

    % for peak i,
    for i=1:numberOfStepsMag
        eastwardDisplacement(i) = estSL(1)*cos(radianer(
            peakLocation(i)));
        northwardDisplacement(i) = estSL(1)*sin(radianer(
            peakLocation(i)));
    end

    prevPositionX = estSL(1)*cos(radianer(firstTimeSample));
    prevPositionY = estSL(1)*sin(radianer(firstTimeSample));

    newPositionX = zeros(numberOfStepsDer,1);
    newPositionY = zeros(numberOfStepsDer,1);

    for i=1:numberOfStepsMag
        newPositionX(i) = prevPositionX + eastwardDisplacement
            (i);
        prevPositionX = newPositionX(i);
        newPositionY(i) = prevPositionY +
            northwardDisplacement(i);
        prevPositionY = newPositionY(i);
    end
```

Chapter 8

Experiments

Several experiments have been performed throughout the presented work of this thesis for the purpose verifying the implemented algorithms for step detection, step length estimation and heading determination. In total, two subjects, collected analysis and training data. The first subject is a 25 years old, 175 cm high female. The second subject is a 23 year old, 185 cm high male. To evaluate the algorithms, data in total duration of 323 minutes were collected from the hip-mounted IMU at a sample rate up to 1000 Hz.

In this chapter is a selection of three field tests presented in order to demonstrate individual and overall performance of the suggested solution. Two of the field tests was set up indoors at NTNU's campus Gløshaugen, in the basement of building Elektro-B. The third field test was performed on an outdoors running track at Øya. A more detailed description will be given for each field test in its respective section. Results from each test is discussed consecutively, yet a final discussion on system performance is saved for Chapter 9.

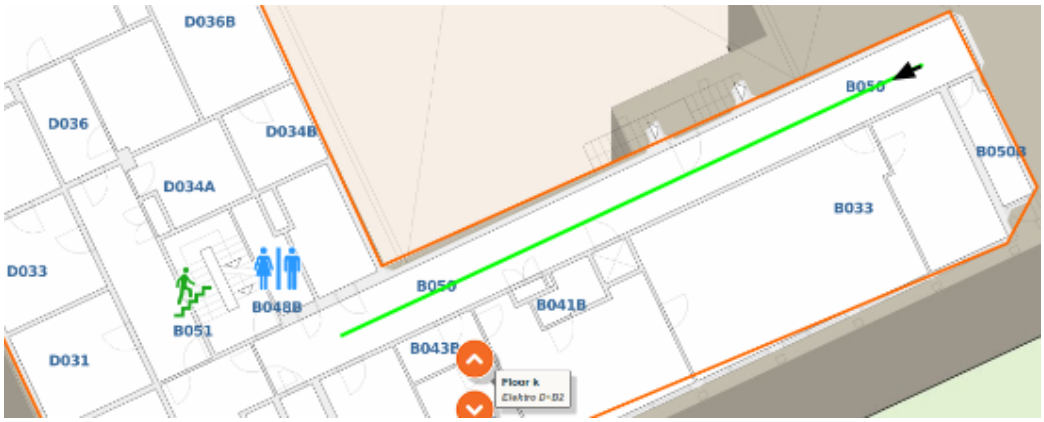


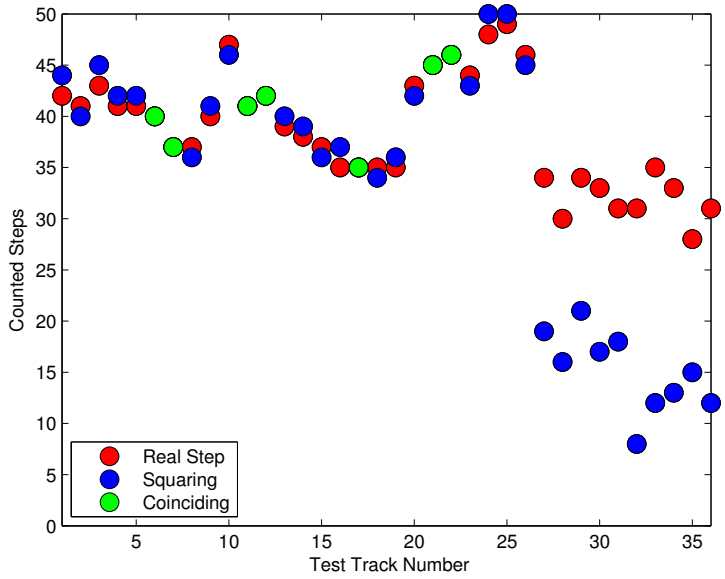
Figure 8.1: 28 m long test trajectory in the basement of Elektro-B

8.1 Straight Line Test Track

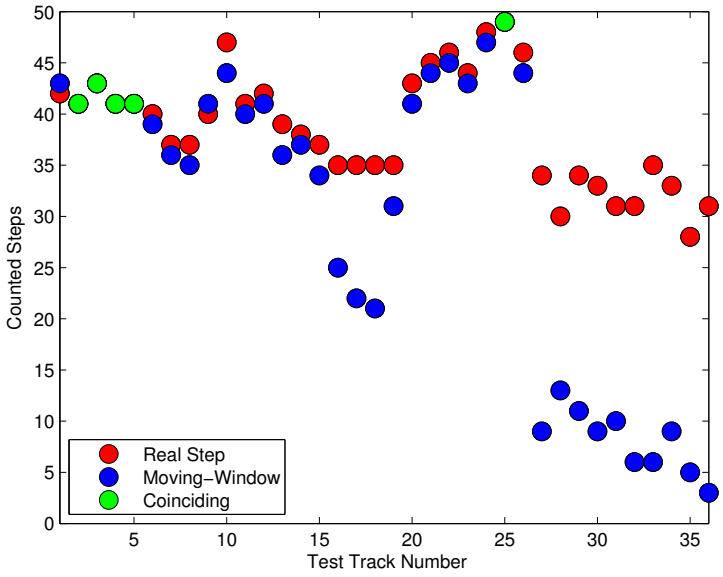
Figure 8.1 illustrates the corridor where the straight line field test was performed. The test trajectory was 28 m long. A total of 70 tests were conducted with two test objects: One female subject and one male subject. Various walking frequencies was applied throughout the test track set, where the subjects tried to keep a steady pace at each test. In order to check accuracy of the step detection algorithm and step length estimation algorithm, the female test set was conducted at an early stage of the implementation of the presented INS in this thesis. For this reason are results from an overall system performance not presented since the heading determination algorithm was not yet implemented at this stage. The male test set was conducted at a later stage to verify the results from the female test set.

8.1.1 Step Detection

The detected steps from the female test set of the Pan-Tompkins algorithm are compared to the real counted number of steps and can be viewed in Figure 8.2. As mentioned in Section 7.5.1, peak detection is performed on both the output from the squaring operation and the moving-window integrator, and results from both filter processes are therefore given.

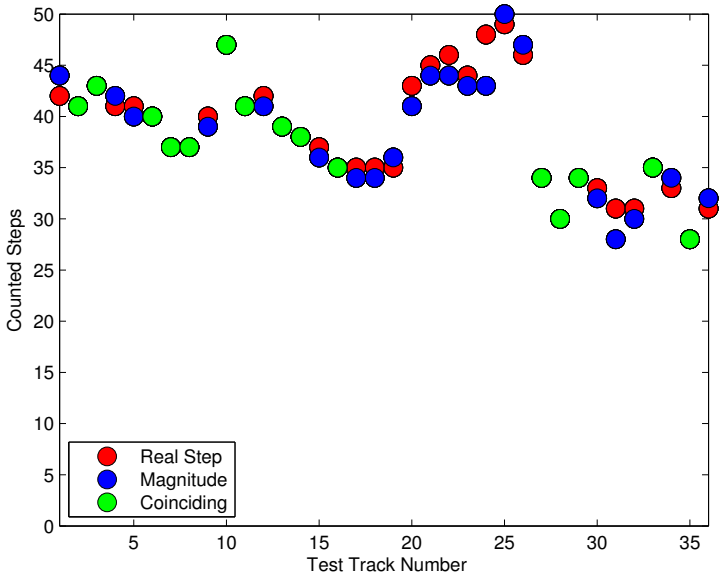


(a) Squaring Operator

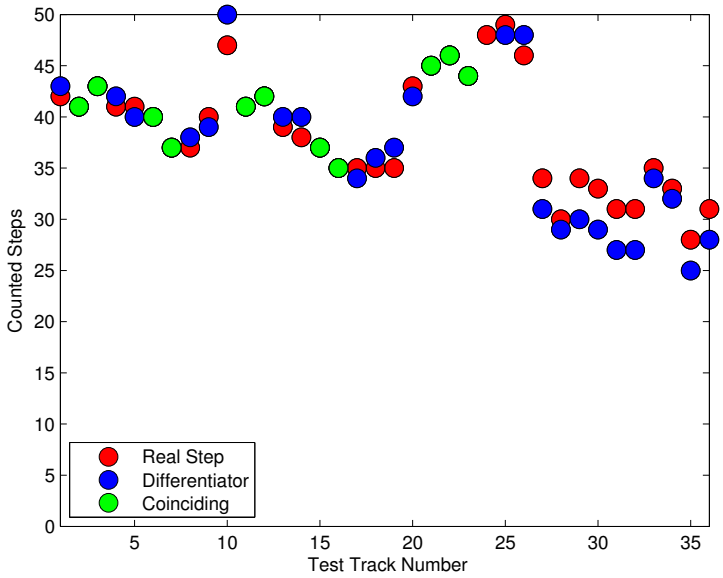


(b) Moving-Window Integrator

Figure 8.2: Step calculations from the Pan-Tompkins algorithm



(a) Magnitude



(b) Differentiator

Figure 8.3: Step calculations from the Libby-peak based algorithm

Figure 8.2a presents the results of the squaring operator of the Pan-Tompkins algorithm for the 35 tests from the female test set. This step detection provides fairly good results for the first 25 tests, but has poor performance for the last ten test tracks. 7 of 35 tests are estimated correctly, and 24 tests are within 5 % accuracy.

The second, and final step detection of the Pan-Tompkins algorithm is the moving-window integrator. The results are presented in Figure 8.2b, and are similar to the results obtained by the Squaring Operator. The results are quit accurate for the first test tracks, but when the real number of steps lies around 30-35 are the step detections off by several steps. 4 of 35 tests are estimated correctly, and 18 tests are within 5 % accuracy.

The detected steps from the female test set of the Libby-Peak based algorithm are compared to the real counted number of steps, and can be viewed in Figure 8.3. As mentioned in Section 7.5.1, peak detection is executed on both the output from the magnitude calculations and the differentiator, and results from both filter processes are therefore given.

Figure 8.3a presents the results from step detection on the magnitude signal in the Libby-Peak based algorithm. This peak detection provides fairly good results, with small deviation from the real counted steps. 15 of 35 tests are estimated correctly, and 33 tests are within 5 % accuracy.

The second step detection of the Libby-Peak algorithm is performed after a differentiator is applied to the signal. The results of this zero-crossing detection is presented in Figure 8.3b. The zero-crossing detection performs similar to the peak detection, but with a slightly poorer result, where 11 of 35 tests are estimated correctly, and 24 tests are within 5% accuracy.

Looking at the walking frequency for each test gives a good indication to why some step detection approaches do not perform well for some test tracks. As can be seen in Figure 8.4, test track 14-18 lie around 2 Hz, and test track 26-35 have frequencies ranging to 2.5 Hz. These are the exact areas where the moving-window integrator performed poorly. This indicate that this step detection approach is not suitable for subjects walking with a high walking velocity. The squaring operator performs better than the moving-window integrator around 2 Hz, but does also have problems with step detection for walking velocity around 2.5 Hz. As previously concluded is the Libby-peak based algorithm most accurate, and performs well for all tested frequencies.

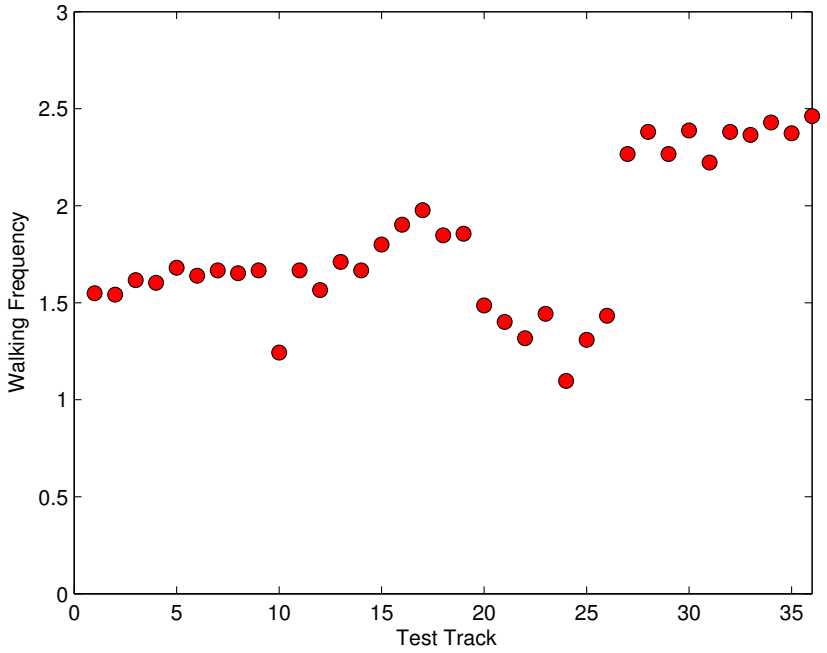


Figure 8.4: Walking Frequency in Test Track

In order to determine average performance of the different step detection approaches the root-mean-square-error (RMSE) of each approach is calculated. RMSE is exactly what it claims to be, the square root of the variance of all estimated and real value, stating how close the estimated values are to the real data points. Lower values of RMSE indicate a better fit, where $RMSE = 0$ is a perfect fit. RMSE from the four step detection approaches are presented in Table 8.1, for both the female and male test sets. For the female test set it can be seen that peak detection on the triaxial magnitude signal is most accurate for overall system performance. For the male test set are the differentiator most accurate. The differing results from the RMSE calculations illustrates how the human gait will impact the different step detection algorithms, and why it is important to test the INS system with several subjects, preferably at different age, height, weight, and other factors which affect the human gait, as described in Section 4.1. The Pan-Tompkins algorithm performs unsatisfactory for both test subjects. However, discarding Pan-Tompkins completely is not necessary at this stage since the algorithm performs well for lower frequencies. The challenge of this algorithm is the

Step Detection Approach	**RMSE Female Test Set**	**RMSE Male Test Set**
Magnitude	1.3	1.7
Differentiator	1.9	1.4
Squaring	9.2	10.7
Moving-Window Integrator	13.3	15.7

Table 8.1: Root-mean-square-error of each step detection approach

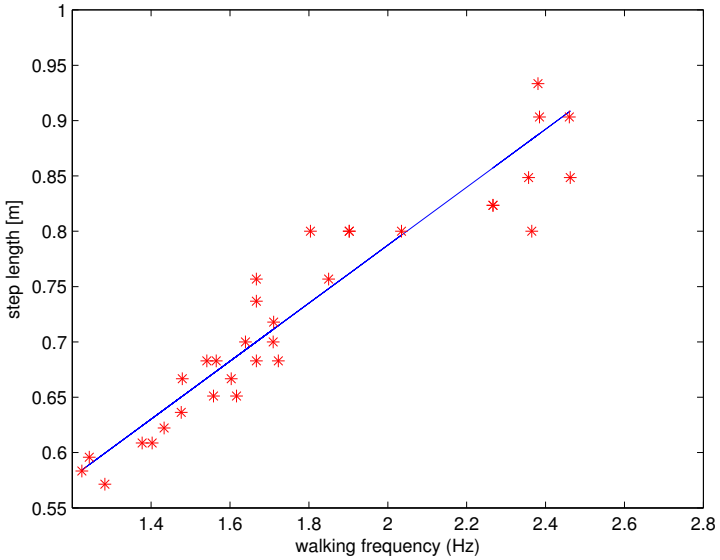
predefined filter values, which are proven to perform better for the cases of lower frequencies. Making these predefined values adaptive could therefore increase the algorithm's performance. However, tests on this matter have not been made, and remains therefore as further work.

8.1.2 Step Length Estimation

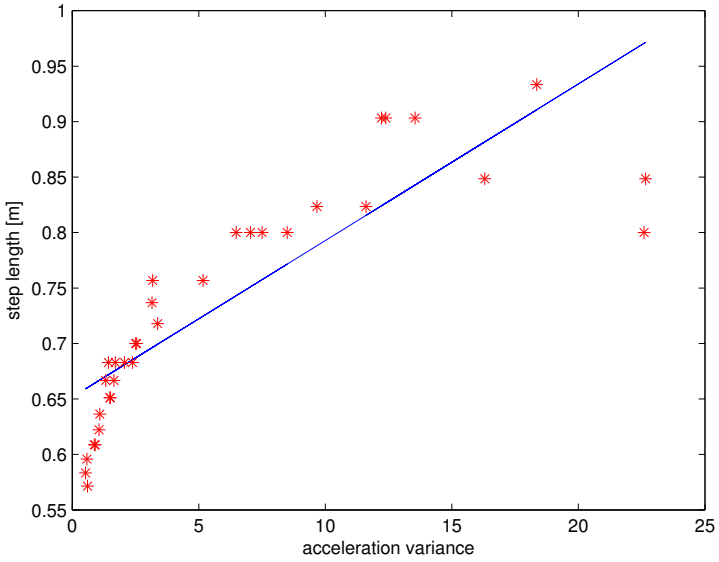
The curve fitting of the relation between step length and walking frequency for the female test set can be viewed in Figure 8.5a, while the result of curve fitting the relation between step length and acceleration variance for the female test set can be viewed in Figure 8.5b.

For Figure 8.5a it should be noticed that the fitted curve is more precise for lower frequencies than higher frequencies. This indicates that subjects walking with lower frequency will gain better step length estimations, $\hat{s}_{wf}$, than if the subject was walking at a higher velocity. The calculated sum of squared residuals is 0.8868, meaning that the curve fitting predicts 88.7 % of the variance in the different walking frequencies.

Figure 8.5b shows that a good linear fit is harder to find for acceleration variance since higher acceleration variance seem to have random variations. The fitted curve for acceleration variance predicts 69.6 % of the acceleration variance. It is therefore expected that estimated step lengths based on acceleration variance, $\hat{s}_{av}$, performs worse than estimated step lengths based on walking frequency, $\hat{s}_{wf}$. Compared to the reviewed literature was this performance not satisfactory. For this reason a verification was performed based on the male test set. The curve fitting then predicted 75.7 % of variance in walking frequency using data from the male test set. For the acceleration



(a) Relation between walking frequency and step length



(b) Relation between acceleration variance and step length

Figure 8.5: Curve Fitting

variance the curve fitting predicted 64.4 % of the variance. Based on these test sets is it therefore not trustworthy that the acceleration variance shall increase with increasing step length in a linear matter above a certain walking frequency. In fact, comparing to the real world this seems legit. If a professional runner where to increase acceleration variance linear to its step length the impact on the neuromuscular system would be considerable. However, further tests must be performed on several subjects in order to draw a final conclusion.

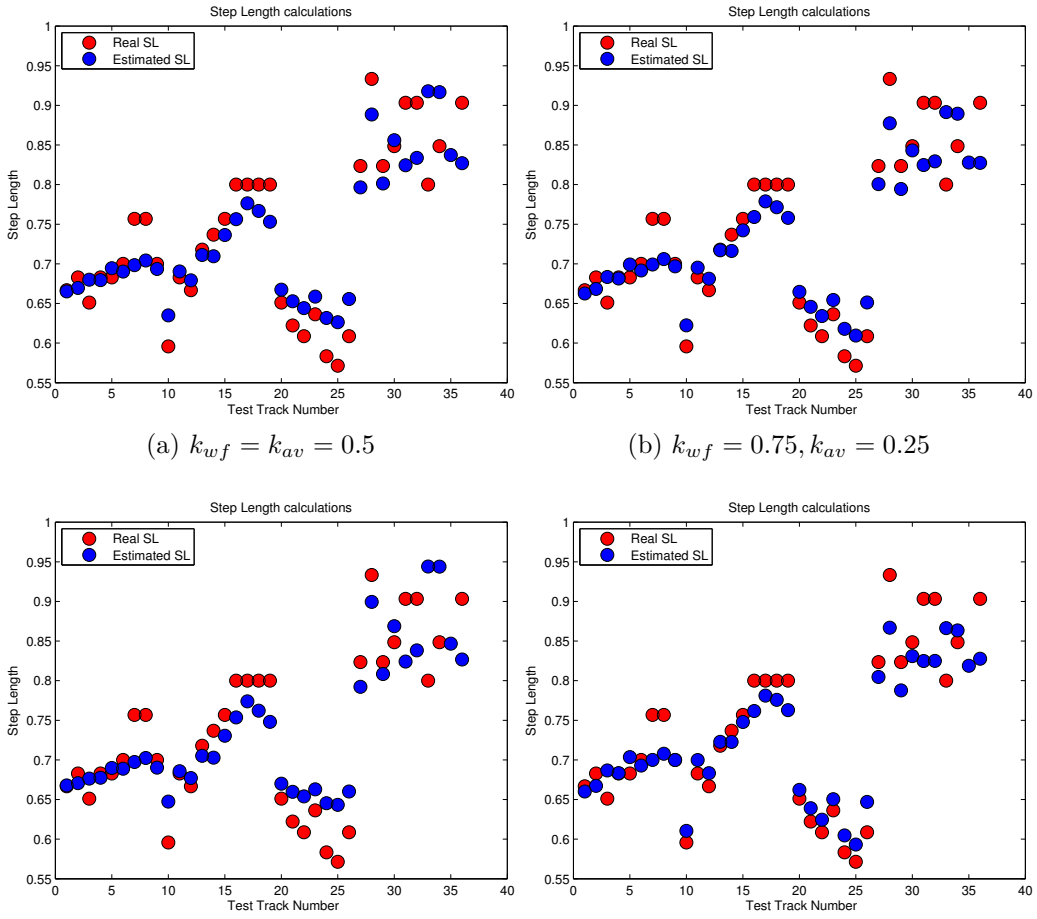
The step length estimation model, $\hat{s}$, use parameter k_{wf} and k_{av} to weight the two functions for step length estimation, as described in Section 5.3, Equation 5.5:

$$\hat{s} = k_{wf} \cdot \hat{s}_{wf} + k_{av} \cdot \hat{s}_{av}$$

Based on the results from the curve fitting should k_{wf} be weighted higher for a more accurate estimation since the curve fitting for walking frequency was better. To verify this was step length estimations tested with different weighting, as shown in Figure 8.6.

The overall performance for all four step length calculations is relatively similar for the first 25 test. For the last 10 tests are performance worse when $\hat{s}_{av}$ are weighted higher than, or equal to $\hat{s}_{wf}$. This was expected since the curve fitting for acceleration variance was unsatisfactory for higher step lengths. The residuals for all four possibilities have been calculated, and can be viewed in Table 8.2. The overall performance of step length estimation for the 35 tests from the female test set is better when step length estimations are made with $k_{wf} = 0.99$ and $k_{av} = 0.01$, which in essence means that one should discard the acceleration variance completely as parameter for step length estimation.

Repeatedly, no final conclusion can be drawn on the performance of the step length estimation algorithm since several subject must perform the test to verify the results. It can however be stated that for human beings with gait characteristics similar to the female subject's gait will the algorithm provide step length estimations at 88.6 % of real step length. This gives RMSE at 2.9 m on the total estimated distance for the female test set.



(a) $k_{wf} = k_{av} = 0.5$

(b) $k_{wf} = 0.75, k_{av} = 0.25$

(c) $k_{wf} = 0.25, k_{av} = 0.75$

(d) $k_{wf} = 0.99, k_{av} = 0.01$

Figure 8.6: Step Length Calculations with various k_{wf} and k_{av}

Parameter settings	**Best fit (%)**
$k_{wf} = k_{av} = 0.5$	0.8283
$k_{wf} = 0.75, k_{av} = 0.25$	0.8668
$k_{wf} = 0.25, k_{av} = 0.75$	0.7714
$k_{wf} = 0.99, k_{av} = 0.01$	0.8863

Table 8.2: Residuals step length estimations with different values for k_{wf} and k_{av}

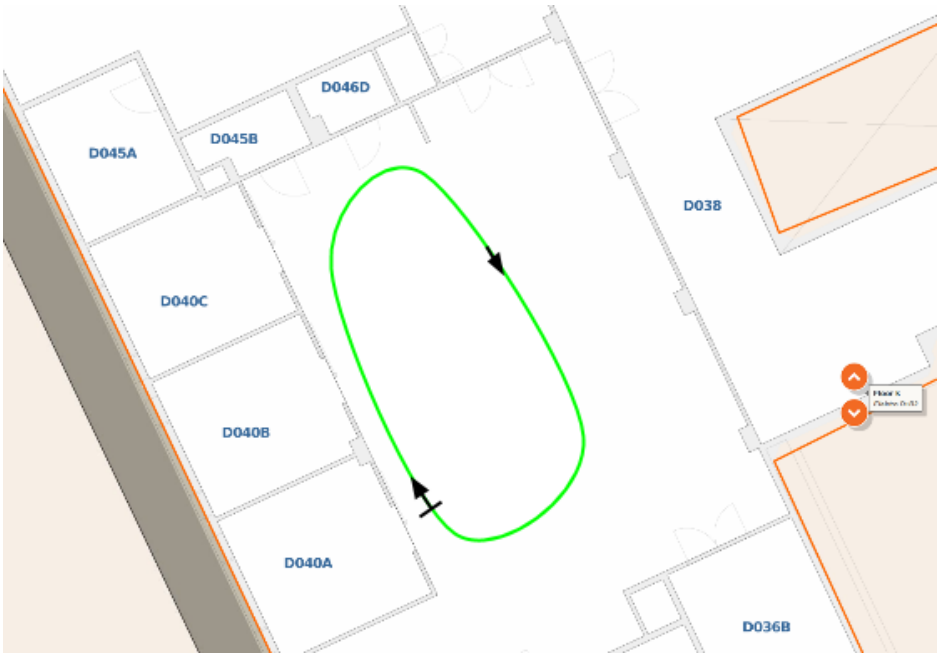


Figure 8.7: 22 m long test trajectory in room D040 at Elektro-B

8.2 Roundtrip Test Track

The round-trip test track was performed in room D040 in building Elektro-B at Gløshaugen, as illustrated in Figure 8.7. The test trajectory was an 50 m^2 office area providing an 22 m long elliptical rout in a room filled with various electronic equipment. Since the straight line test track showed better performance when using the Libby-peak based algorithm, was this algorithm applied in this test track set. In addition, the weighting of k_{wf} and k_{av} was set such as to increase system performance, hence $k_{wf} = 0.99$ and $k_{av} = 0.01$.

The round-trip test was conducted with the main purpose of verifying heading determination. Thus, the focus of this section is on magnetometer calibration in order to check the entire system performance. For this reason are step detection and step length estimations only briefly discussed, and the results will not be graphically illustrated. The female subject walk 10 test tracks at a steady pace. Data from other subjects were not conducted for this test track since heading determination is not dependent on gait characteristics.

Step Detection Approach	RMSE
Magnitude	1.2
Differentiator	0.8

Table 8.3: Root-mean-square-error of each step detection approach

Test track number	Estimated step length	Estimated Distance	Distance error
1	0.6698	24.1135	2.1135
2	0.7037	26.0366	4.0366
3	0.6998	25.1911	3.1911
4	0.6958	25.0471	3.0471
5	0.6919	24.9087	2.9087
6	0.6919	24.9073	2.9073
7	0.6997	24.4907	2.4907
8	0.7037	26.0369	4.0369
9	0.7038	25.3382	3.3382
10	0.7084	24.0873	2.0873

Table 8.4: Estimated step length results

8.2.1 Step Detection and Step Length Estimation

RMSE values for the Libby-peak based step detection algorithm can be seen in Table 8.3. For this test set was the walking frequency more suitable for the differentiator, which performed slightly better than the magnitude peak detection. The differentiator was therefore applied when estimating the total traveled distance.

Table 8.4 summarize the step length estimation results. The estimated step length for all ten test tracks gives RMSE 0.0875 m, translating into a RMSE for estimated distance at 3.0845 m. Comparing to the straight line test one can see that the performance are worse for the roundtrip test track. This supports the theory in Section 5.2 which stated that in most cases was step length not as critical as heading determination.

8.2.2 Heading Determination

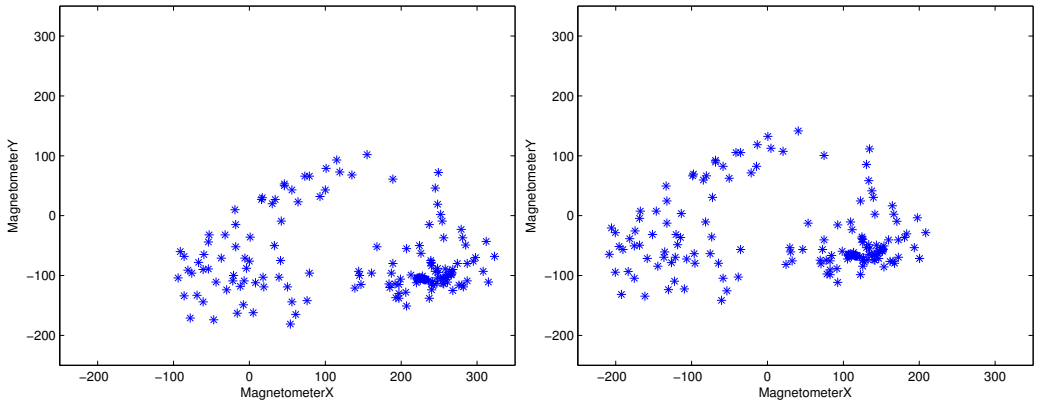
Before determining the heading of the female subject must the magnetometers be calibrated. Based on the theory in Chapter 6 was several approaches implemented and tested. Figure 8.8 presents the results from the various magnetometer calibrations, where sensor readings from the first conducted test track are used as an example.

Figure 8.8a shows how the sensor data from the horizontal magnetometer axes looked prior to magnetometer calibration. It can be seen that the magnetic field appears as an ellipse which are not centered at $(0, 0)$, just as described in the theory in Section 6.2.3. Based on these observations can it be concluded that both hard- and soft-iron distortions are present in room D040.

Figure 8.8b shows the magnetic field after compensating for the hard-iron interferences. The field are now centered at $(0, 0)$, but are still affected by soft-iron distortions, causing the elliptical form.

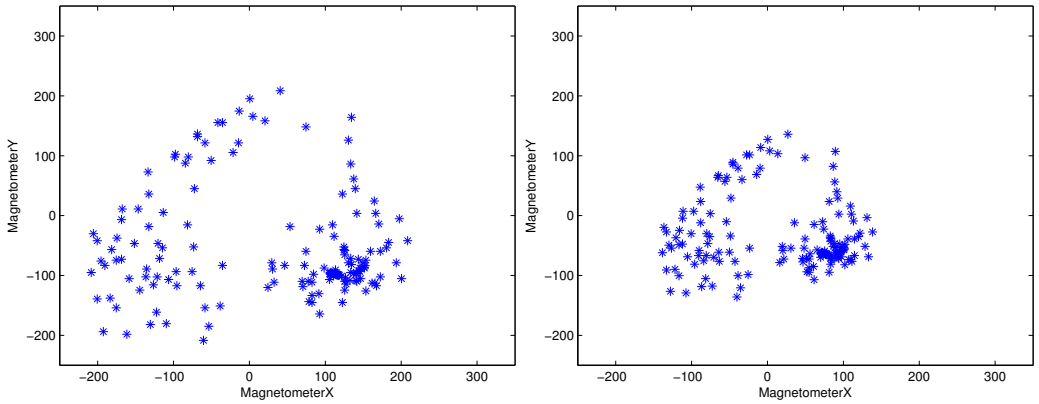
In Figure 8.8c are both hard-iron and soft-iron distortions removed from the magnetic field. The field now appears more like a circle than for the previous cases, though with some deviations. It is expected that these deviations will effect the performance of the heading determination.

The magnetometer calibrations used in figures 8.8b and 8.8c are both based on maximum and minimum values of the horizontal components of the magnetic field. For a real-time system will these values not be available. A spot turn calibration was therefore performed by the female subject prior to walking, as described in Section 6.3. Using maximum and minimum values from the magnetic field conducted by a spot turn results in the magnetic field shown in Figure 8.8d. It can be seen that this magnetic field appears similar to the one in Figure 8.8c. Since scaling not will affect the triangular based heading determination it can be expected that the spot turn calibration performs just as well as the methods based on the predefined magnetometer set.



(a) Magnetic Field with no magnetometer calibration

(b) Magnetic Field after removing hard-iron distortions



(c) Magnetic Field after removing soft-iron distortions

(d) Magnetic Field with spot turn calibration

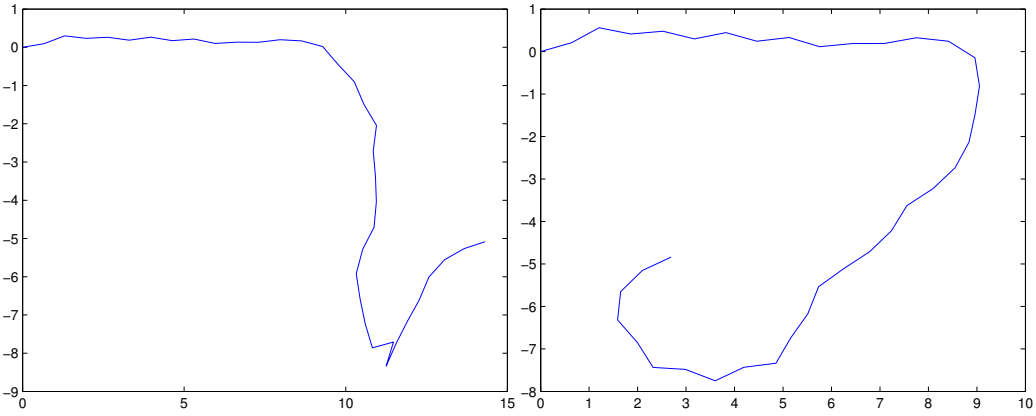
Figure 8.8: The horizontal magnetic field before and after magnetometer calibrations

After the heading is determined based on a calibrated magnetic field, the INS performs dead reckoning in order to determine the walked track of the female subject. Figure 8.9 shows the final performance of the suggested INS, where each figure is based on the each of the magnetic fields presented in Figure 8.8. Since the original track is not graph together with the estimated track should an explanation be given on the expected outcome. As stated earlier are the office area 10 meter long and 5 meters wide. Thus, the expected estimated track should be an approximately 10 m straight line before turning right, and walking approximately 4-5 meters before turning right again, walking a new 10 meters long straight line before turning right again, heading back to the start position.

Figure 8.9a shows system performance based on the magnetic field with no magnetometer calibrations. The first part of the track is correct. It can be seen that the subject walks approximately 10 meters before making a right turn. However, something causes the signal to distort significantly after this first turn, and the second right turn never appears on the graph. Probably are error sources present in this lower right corner of the room, causing an interference in the measured magnetic field. This illustrates how important magnetometer calibration is in order to get a well-working INS.

Figure 8.9b shows system performance based on the magnetic field where hard-iron interferences was removed. The performance is better than in the case of no magnetometer calibration. It can be seen that the female subject walks approximately 8-9 meters before turning right. Though the second turn is not sharp enough compared to the real track, it is better than in the preceding graph. The error sources present in this corner of the room is therefore not only hard-iron interferences. It seems like the third right turn is correct with regards to heading, but are effected by the fact that the second turn fails.

Since the only difference that can be seen between the magnetic fields in figures 8.8c and 8.8d are field strength have figures 8.9c and 8.9d similar form. Compared to the system performance where only hard-iron distortions are removed, are the performance not severely improved. Based on these results can it therefore be concluded that the performed magnetometer calibrations are not sufficient.



(a) Inertial navigation with no magnetometer calibration

(b) Inertial navigation after removing hard-iron distortions

(c) Inertial navigation after removing soft-iron distortions

(d) Inertial navigation with spot turn calibration

Figure 8.9: Inertial navigation before and after magnetometer calibrations

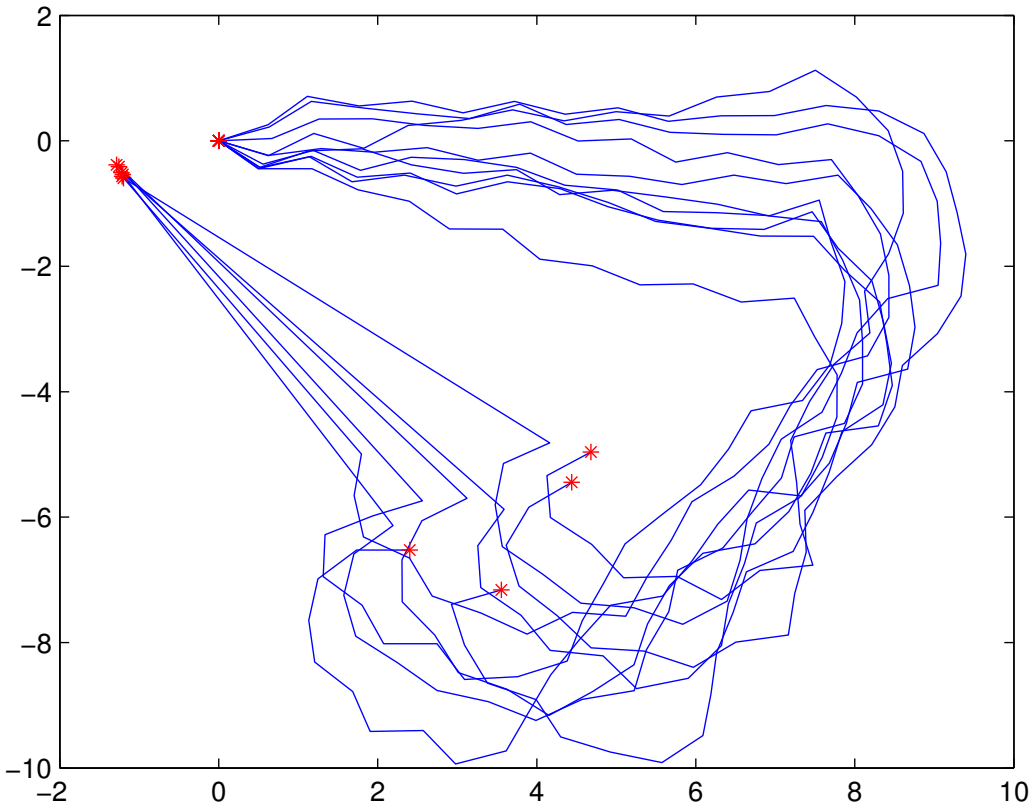


Figure 8.10: Results of the inertial navigation from all ten test track graphed in one figure

While figures 8.8 and 8.9 show the results for the first conducted test track, are results from the entire test track set plotted together in Figure 8.10. In the figure are dead reckoning performed based on detected steps and estimated step lengths as described in Section 8.2.1, and heading determination after a spot turn calibration. The first part of the test track is estimated with a good positioning error, where the estimated track deviates from the original with a maximum of 1 m detour. Interferences in the lower right corner cause the positioning error to increase, and for four of the test tracks are final positioning error several meters. Six of the test tracks actually ends up close to the starting position, but experience fluctuations along the way due to interferences in the magnetic field. Overall system performance will be more throughly discussed in Chapter 9.

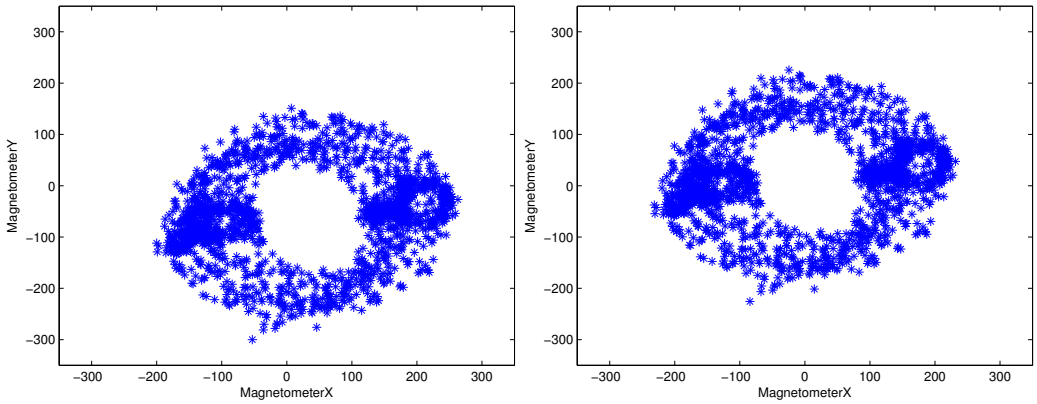
Figure 8.11: 400 m long competitive running track, Øya

8.3 Running track test

A last test track was performed outdoors at the running track shown in Figure 8.11. Since the roundtrip track suffered from poor magnetometer calibration, the running track test was needed to verify system performance in a more magnetometer-friendly environment.

Similar to the figures in Section 8.2, Figure 8.12 shows the resulting magnetic fields when applying different magnetometer calibrations. Except Figure 8.12a are the other magnetic fields close to an ideal circle, with many calibration points. Thus, results from this test track set are expected to perform better than for the case of the roundtrip test track set.

By examining the results in Figure 8.13 it can be seen that the system performs really well when using a magnetometer calibration method. After 400 meters are the positioning error approximately 10 meters.



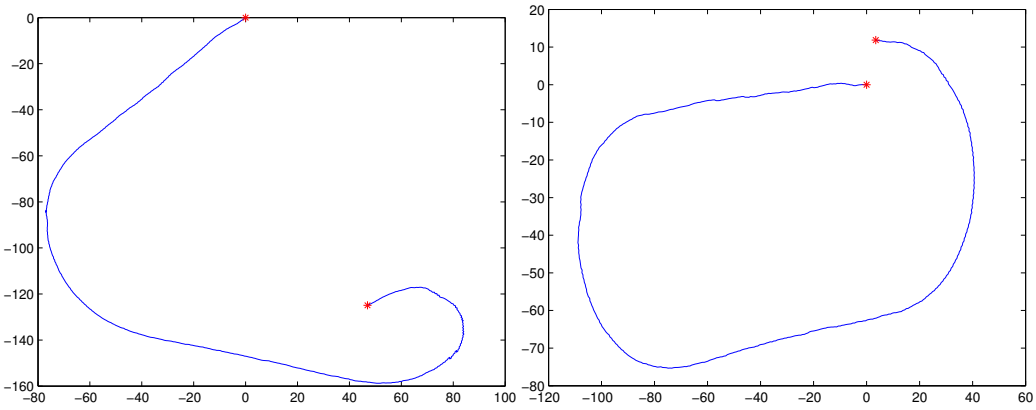
(a) Magnetic Field with no magnetometer calibration

(b) Magnetic Field after removing hard-iron distortions

(c) Magnetic Field after removing soft-iron distortions

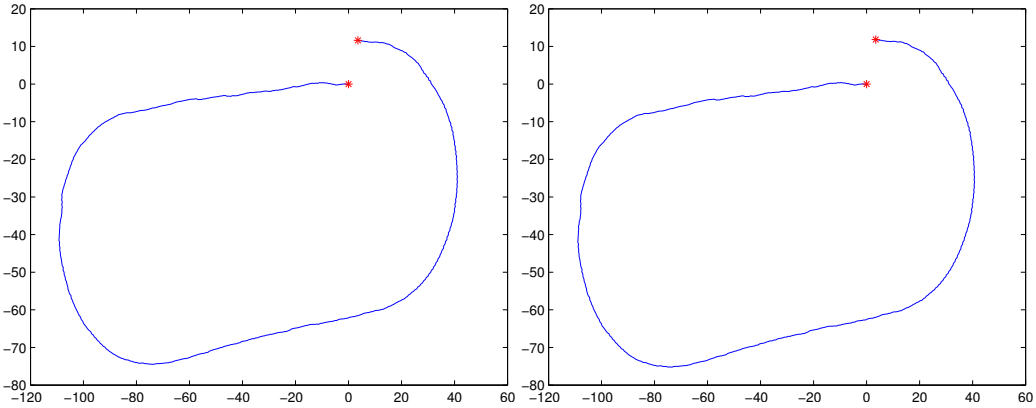
(d) Magnetic Field with spot turn calibration

Figure 8.12: The horizontal magnetic field before and after magnetometer calibrations



(a) Inertial navigation with no magnetometer calibration

(b) Inertial navigation with spot turn calibration



(c) Inertial navigation after removing hard-iron distortions

(d) Inertial navigation after removing soft-iron distortions

Figure 8.13: Inertial navigation before and after magnetometer calibrations

Chapter 9

Discussion, Conclusion, and Suggestions for Future Work

The work presented in this thesis set out to explore the concept of indoor positioning using a hip-mounted inertial measurement unit, and has accomplished to develop a prototype localization system based on the detection of steps, step length estimation and heading determination. The thesis has also investigated different sensor configurations suitable with the proposed IMU to increase system accuracy.

In this chapter, the suggested solution to parts of the WiLoc localization system are discussed and compared with the aim of obtaining an indoor positioning system for industrial facilities with position error up to 5 m. In particular, results from the experiments in Chapter 8 are further analyzed, and decisions made with regards to solving the given assignment are assessed. Conclusions are drawn for each component and the whole system design, in addition to suggestions for future work.

9.1 Step Detection

The step detection algorithms implemented in this thesis was introduced in Chapter 4, where each algorithm performs computing on sensor readings from a hip-mounted IMU. The first algorithm, Pan-Tompkins, performs peak

detection on a filtered output of the vertical acceleration measured by the IMU's sensors. The second algorithm is based on Libby-peak, which uses magnitude of the three dimensional acceleration signal in order to apply zero-detection on a differentiator's output signal.

Results from testing Pan-Tompkins algorithm showed that detected steps for walking frequency below 1.6 Hz are quite accurate compared to the real number of steps taken by the subject. On the other hand, the algorithm performs poorly for frequencies above 1.6 Hz, where the last filter, the moving-window integrator, only adds error to the filtered signal. This indicated that the filter variables for the integrator are adjusted to perform well on test tracks sets with low walking frequency. As mentioned in Section 4.1, to better performance of a step detection algorithm have reviewed literature suggested to implement a motion classifier. A motion classifier can address the experience problems with Pan-Tompkins, in particular the moving-window integrator, and account for the various gait patterns from one individual to another. In addition, higher walking frequencies can be discovered, as well as backwards and sideways walks. The motion classifier acts as a state machine, optimizing step detection parameters for a particular motion, making the predefined filter values adaptive. Nevertheless, without a motion classifier are the performance of Pan-Tompkins algorithm unsatisfactory for the requirements of the WiLoc localization system.

Using magnitude of the triaxial acceleration signal to perform zero-detection in the Libby-peak based algorithm has proven to perform well on all the conducted test tracks. A probable reason may be the fact that Spatial IMU contains an advanced Kalman Filter. The sensor readings have therefore already been filtered, and appear noise-free. In other words, the simplest solution is often the correct solution. Small variations were observed between performance of peak detection on the magnitude signal and zero-crossing detection on the differentiator output when comparing test tracks results from two subjects. Again, this may come as a result of the gait characteristics of the subjects, thus enforcing the need for a motion classifier. However, the overall performance of the Libby-peak based algorithm is satisfying for both test subjects, on all tested walking velocities. Of the implemented and tested algorithms, Libby-peak is the preferred step detection algorithm for the WiLoc localization system.

Evaluating the sensor placement, the hip-mounted IMU has proven to be

suitable for the implemented step detection algorithms. A possible challenge for future work is determining the time interval of step occurrences. Since the suggested solution was a prototype system are the time stamps already known, which will not be the case for a real-time system. The literature suggests that a foot-mounted IMU are more suitable for determining the individual gait phases, but tests on this matter has not been performed. Consequently, is hard to state if the performance will degrade on a real-time system, but the reader should be aware of this matter.

One last aspect of step detection should be pointed out. In the presented implementation of this thesis was the peak detection threshold set based on trial and error during the implementation phase. Though this yielded good results for the conducted testes tracks acceleration signals may suffer from considerable more noise when using a different IMU. This can create false peaks with varying magnitude in the acceleration signal, which are hard to remove with a static threshold. In such cases are an adaptive threshold suggested.

9.1.1 Future work

Based on these conclusions is the following list of suggestions for future work on step detection presented:

- Implement a motion classifier
- Design an adaptive threshold algorithm for peak detection

9.2 Step Length Estimation

The step length estimation algorithm implemented in this thesis was introduces in Chapter 5. The estimation method uses weighted, curve fitted linear models for walking frequency and acceleration variance to define step length parameters prior to system startup. Based on these predefined parameters are step lengths estimated during a test track. The linear models was design on a basis of 35 test tracks from a female subject and verified by 35 test tracks from a male subject.

The performance of the algorithm was first investigated based on the results from curve fitting step length as a function of computed data for walking frequency and acceleration variance. The curve fitting showed satisfying results for a linear model based on walking frequency, where a slightly better fit for low walking frequencies was observed. Contrary, a good linear fit was not obtained for the curve fitting of acceleration variance. Though literature suggests linear dependency between step length and acceleration variance have results from tests performed in this thesis proven the opposite. However, since tests was only performed on two subjects can not a final statement be given. Thus, testing should be made on a larger test set using several individuals with contrasting gait characteristics in order to draw a final conclusion on this matter.

Applying various weights to the obtained linear models provided different results on step length estimations. This was expected since the curve fitting was more successful for walking frequency than acceleration variance. Naturally, the results yielded better performance when the liner model for walking frequencies was weighted high, and worse results when the linear model for acceleration variance was weighted high. In order to increase system performance the linear model for acceleration variance was approximately weighted to zero for inferior tests.

Since the applied step length estimation algorithm is dependent on having good predefined parameters can several suggestions be given in order to increase the algorithm's performance. Firstly, using a motion classifier gives the opportunity of defining different parameter values for various walking velocities. Not only could the parameters be set differently, but different step length estimation algorithms could then be applied for different movements. As seen in the experiments in Chapter 8, the implemented estimation method would perform severely better if a curve fitting was performed only on the low frequencies. Then would also the acceleration variance be suitable as a linear model. Secondly, applying a calibration process where certain step length parameters are calculated and set to the specific user can also increase the accuracy of the estimator. The theory in Chapter 5 describes examples for other step length estimation models, including the inverted pendulum model (biomechanical models) and expressing step length as a function of sensor data other than walking frequency and acceleration variance (parametric models).

9.2.1 Future work

Based on these conclusions is the following list of suggestions for future work on step length estimation presented:

- Perform curve fitting on a larger test set using several individuals with contrasting gait characteristics
- Implement, test, and compare performance of other step length algorithms based biomechanical models or on another parametric model
- Implement a motion classifier in order to use different predefined parameters or step length estimation algorithms
- Perform a calibration process, adjusting step length parameters to the specific user

9.3 Heading Determination and Overall System Performance

The method for heading determination implemented in this thesis is based on the horizontal components of the IMU's magnetometers. The magnetometers are calibrated prior to heading determination, where three different approach for removing error sources such as hard- and soft-iron interferences was tested. Further, results from step detection, step length estimation and heading determination are combined to calculate the subject's trajectory.

Three different approaches for magnetometer calibration was implemented, tested, and compared with regards to each other and to the results in which no magnetometer calibration was performed. The results from the roundtrip track yields that the two magnetometer calibrations which removed hard and soft-iron interferences provide a better system performance than if only hard-iron interferences are removed. The first hard- and soft iron compensation method provides calibration based on the collected test set values, and will thus not work in a real-time system. The second method was based on spot turn calibration, performed by the subject prior to walking. Fortunately, performance was identical for these two calibration methods, and thus should

the spot-turn calibration be performed for a real-time version of the WiLoc localization system.

However, looking at the magnetic field after the spot-turn calibration one sees that it deviates from the ideal circle, and suffer from few calibration points. This indicates that the magnetometers are still affected by error sources. The implemented spot-turn calibration process should therefore be more advance designed. The theory on magnetometer calibration in Chapter 6 listed several other error sources for converting magnetometer readings to a certain heading. For future work should some, or all of these error sources be eliminated from the magnetic field.

The overall system performance of the roundtrip track suffered from error sources which was not removed by the implemented calibration process. The test track set obtained an estimated distance error between 2-4 meters. Given the requirements of the WiLoc localization system are these position errors labeled as okay. However, taking into account that the subject walked only 22 meters in total the INS can be anticipated to suffer from larger drifts for a longer data set. In addition, the suggested solution is suppose to work in an industrial environment which can be expected to challenge the magnetometers further. Further testing in a larger industrial area and for a longer period of time should be conducted in order to state how large drifts the INS will experience.

Nonetheless, the running test track provided good news for the future of the suggested solution. In an outdoors environment where the magnetometers suffered from less error sources was the position error relatively small compared to the total traveled distance. After walking 400 meters was the drift only 10 meters. This indicates that the step detection and step length estimation are providing the INS with satisfying values. This also supports reviewed literature, where it was stated that in most cases are step length estimations not as crucial as heading determination.

To complete the full circle, the suggested solution can in fact be sufficient enough, if it was integrated with another positioning system. As suggested in Chapter 2, GNSS, Bluetooth, WLAN and RFID are all possible indoor positioning methods suitable as an external positioning method for the proposed INS. A recommended priority for future work on the presented INS is testing RFID indoor positioning prior to Bluetooth positioning. RFID is currently more popular for tracking applications since it is easily incorporated

in production facilities and has a simple communication protocol.

9.3.1 Future Work

The suggested solution presented in this thesis was found to perform well for step detection and step length estimation, but are expected to suffer from significant drifts when the user wearing the IMU are making turns. For this reason will previously proposals for future work not improve the system performance as much as the effect of improving the heading determination. The following list of proposals include the most important suggestions for future work, for both individual components of the system as well as the overall system design:

- Implement a motion classifier
- Implement a more advanced magnetometer calibration process
- Perform testing on a larger test set using several individuals with contrasting gait characteristics
- Integrate and test IMU with other positioning methods
- Develop a real-time system

Bibliography

[1] D. R. Mautz, *Indoor Positioning Technologies*. PhD thesis, ETH Zurich, Feb. 2012.

[2] F. Forskningsinstitutt, "Introduction to Inertial Navigation Systems," May 2014.

[3] R. O. o. B. (ROB), "How GNSS Works," Oct. 2011.

[4] D. Smith, "The difference between standard GPS and A-GPS ," June 2007.

[5] Y. S. Vintervold, "Camera-Based Integrated Indoor Positioning," Master's thesis, Norwegian University of Science and Technology, June 2013.

[6] B. Vik, *Integrated Satellite and Inertial Navigation Systems*. 2012.

[7] O. J. Woodman, "An introduction to inertial navigation," tech. rep., 2007.

[8] A. Navigation, "Spatial Reference Manual ," Sept. 2013.

[9] Wikipedia, "Flight Dynamics," May 2014.

[10] "Human Walking Analysis, Evaluation and Classification Based on Motion Capture System," 1994.

[11] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek, "Automatic Step Detection in the Accelerometer Signal,"

[12] M. B. KjÃ¦rgaard, "Indoor Positioning with Radio Location Fingerprinting," Apr. 2010.

[13] S. Godha, G. Lachapelle, and E. Cannon, "Integrated GPS/INS System for Pedestrian Navigation in a Signal Degraded Environment ," *ION GNSS*, Sept. 2006.

[14] R. G. Stirling, "Development of a Pedestrian Navigation System Using Shoe Mounted Sensor," Master's thesis, University of Alberta, 2004.

[15] S. H. Shin, C. Park, J. W. Kim, H. Hong, and J. M. Lee, "Adaptive Step Length Estimation Algorithm Using Low-Cost MEMS Inertial Sensors," in *Sensors Applications Symposium, 2007. SAS '07. IEEE*, pp. 1–5, Feb 2007.

[16] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A Step, Stride and Heading Determination for the Pedestrian Navigation System," *Journal of Global Positioning Systems*, vol. 3, no. 1-2, pp. 273–279, 2004.

[17] E. Andersson, "Motion Classificaiton and Step Length Estimation for GPS/INS Pedestrian Navigation," Master's thesis, KTH Electrical Engineering, 2012.

[18] G. Hasan, K. Hasan, R. Ahsan, T. Sultana, and R. C. Bhowmik, "Evaluation of a Low-Cost MEMS IMU for Indoor Positioning System ," *Inertnaional Journal of Emerging Science and Engineering*, vol. 1, Sept. 2013.

[19] Wikipedia, "Navigation," May 2014.

[20] G. Hein, M. Paonni, V. Kropp, and A. Teuber, "GNSS Indoors ," Mar. 20008.

[21] *Evaluation of Assisted GPS (AGPS) in Weak Signal Environments Using a Hardware Simulator* , ION GNSS 2004 Conferenc, Sept. 2004.

[22] A. Bekkelien, "Bluetooth Indoor Positioning," Master's thesis, University of Geneva, Mar. 2012.

[23] *Experiments on Local Positioning with Bluetooth* , Proceedings of the International Conference on Information Technology: Computers and Communication, 2003.

[24] H. Liu, H. Darabi, P. P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems.," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 37, no. 6, pp. 1067–1080, 2007.

[25] F. Forno, G. Malnati, and G. Portelli, "Design and implementation of a Bluetooth ad hoc network for indoor positioning," *IEEE Proc.-Softw*, vol. 152, Oct. 2005.

[26] T. J. Gallagher, B. Li, A. G. Dempster, and C. Rizos, "A sector - based campus - wide indoor positioning system," Sept. 2010.

[27] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," in *INFOCOM*, pp. 775–784, 2000.

[28] G. Borriello, W. Brunette, M. Hall, C. Hartung, and C. Tangney, "Reminding About Tagged Objects Using Passive RFIDs.," in *Ubicomp* (N. Davies, E. D. Mynatt, and I. Siio, eds.), vol. 3205 of *Lecture Notes in Computer Science*, pp. 36–53, Springer, 2004.

[29] A. Lim and K. Zhang, "A Robust RFID-Based Method for Precise Indoor Positioning.," in *IEA/AIE* (M. Ali and R. Dapoigny, eds.), vol. 4031 of *Lecture Notes in Computer Science*, pp. 1189–1199, Springer, 2006.

[30] P. Vorst, J. Sommer, C. Hoene, P. Schneider, C. Weiss, T. Schairer, W. Rosenstiel, A. Zell, and G. Carl, "Indoor Positioning via Three Different RF Technologies," in *4th European Workshop on RFID Systems and Technologies (RFID SysTech 2008)*, no. 209 in ITG-Fachbericht, (Freiburg, Germany), VDE Verlag, June 10-11 2008.

[31] "Right-handed Coordinate System," Jan. 2014.

[32] "Newton's Laws of Classic Mechanics," Jan. 2014.

[33] B. Muset and S. Emerich, "Distance Measuring using Accelerometer and Gyroscope Sensors," *Carpathian Journal of Electronic and Computer Engineering*, no. 5, pp. 83–86, 2012.

[34] R. Stirling, J. Collin, K. Fyfe, and G. Lachapelle, "An Innovative Shoe-Mounted Pedestrian Navigation System ," *GNSS*, Apr. 2003.

[35] B. Zhang, S. Jiang, K. Yan, and D. Wei, "Human Walking Analysis, Evaluation and Classification Based on Motion Capture System," Dec. 2011.

[36] T. Judd and R. W. Levi, "Dead reckoning navigational system using accelerometer to measure foot impacts," Dec. 1999.

[37] I. Cleland, B. Kikhia, C. Nugent, A. Boytsov, J. Hallberg, K. Synnes, S. McClean, and D. Finlay, "Optimal Placement of Accelerometers for the Detection of Everyday Activities," *Sensors*, vol. 13, pp. 9183–9200, July 2013.

[38] D. O. Olguin and A. Pentland, "Human Activity Recognition: Accuracy across Common Locations for Wearable Sensors," (Nibtreauc), EEE 10th International Symposium on Wearable Computers, Oct. 2006.

[39] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *IAAI'05: Proceedings of the 17th conference on Innovative applications of artificial intelligence*, pp. 1541–1546, AAAI Press, 2005.

[40] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data.," in *Pervasive* (A. Ferscha and F. Mattern, eds.), vol. 3001 of *Lecture Notes in Computer Science*, pp. 1–17, Springer, 2004.

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

[42] R. Libby, "A simple method for reliable footstep detection on embedded sensor platforms," June 2008.

[43] S. H. Shin and C. G. Park, "Adaptive step length estimation algorithm using optimal parameters and movement status awareness," *Medical Engineering & Physics*, vol. 33, pp. 1064–1071, Nov. 2011.

[44] Wikipedia, "Interpolation," May 2014.

[45] S. Godha and G. Lachapelle, "Foot mounted inertial system for pedestrian navigation," *Measurement Science and Technology*, May 2008.

[46] I. Skog, P. Händel, J. Rantakokko, and J. Nilsson, "Zero-velocity detection in pedestrian navigation systems - an algorithm evaluation,"

[47] P. Goyal, V. Ribeiro, H. Saran, and A. Kumar, "Strap-down Pedestrian Dead-Reckoning system," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pp. 1–7, Sept 2011.

[48] G. Thüer and T. Verwimp, "Step Detection Algorithms for Accelerometers," *Paper of the E-lab Master Thesis*, 2008.

[49] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm ," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, Mar. 195.

[50] H. Sedghamiz, "Complete Pan Tompkins Implementation ECG QRS detector."

[51] *MEMS-IMU Based Pedestrian Navigator for Handheld Devices*, Sept. 2001.

[52] *An Integrated GPS and Multi-Sensor Pedestrian Positioning System for 3D Urban Navigation* , 2009.

[53] W. Chen, R. Chen, Y. Chen, H. Kuusniemi, and J. Wang, "An Effective Pedestrian Dead Reckoning Algorithm Using a Unified Heading Error Model," 2010.

[54] Y. Zhuang, H.-W. Chang, and N. El-Sheimy, "Adaptive Step Length Estimation Algorithm for Pedestrian Navigation Using Smart Phones."

[55] W. Zijlstra and A. L. Hof, "Assessment of spatio-temporal gait parameters from trunk accelerations during human walking," *Gait and Posture*, vol. 18, pp. 1–10, 2003.

[56] R. Gonzalez, D. Alvarez, A. Lopez, and J. Alvarez, "Modified Pendulum Model for Mean Step Length Estimation," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pp. 1371–1374, Aug 2007.

[57] V. Renaudin, M. Susi, and G. Lachapelle, "Step Length Estimation Using Handheld Inertial Sensors," *Sensors - Open Access Journal*, vol. 12, June 2012.

[58] *Comparison and Evaluation of Acceleration Based Step Length Estimators for Handheld Devices*, Sept. 2010.

[59] G. Recktenwald, "Least Squares Fitting of Data to a Curve," Nov. 2007.

[60] "Determining your orientation," June 2014.

[61] M. J. Caruso, "Applications of Magnetic Sensors for Low Cost Compass Systems,"

[62] P. Kaniewski and J. Kazubek, "Integrated System for Heading Determination," *Acta Physica Polonica A*, vol. 116, no. 3, 2009.

[63] Matlab, "Butterworth Filter Design ."

Appendix A

Spatial Specifications

Spatial's specifications are summarized in tables A.1, A.2, A.3, A.4 and A.5.

NAVIGATION	
Horizontal Position Accuracy	2.0 m
Vertical Position Accuracy	3.0 m
Position Accuracy (with DGNSS)	0.6 m
Vertical Position Accuracy (with DGNSS)	1.0 m
Velocity Accuracy	0.05 m/s
Roll & Pitch Accuracy (Static)	0.1°
Heading Accuracy (Static)	0.5°
Roll & Pitch Accuracy (Dynamic)	0.2°
Heading Accuracy (Dynamic with GNSS)	0.2°
Heading Accuracy (Dynamic, Magnetic Only)	0.8°
Heave Accuracy	5% or 0.05 m
Orientation Range	Unlimited
Hot Start Time	500 ms
Internal Filter Rate	1000 Hz
Output Data Rate	Up to 1000 Hz
Latency	0.4 ms

Table A.1: Navigation Specifications

GNSS	
Supported Navigation Systems	GPS L1 GLONASS L1 GALILEO E1 BeiDou L1
Supported SBAS Systems	WAAS EGNOS MSAS GAGAN QZSS
Update Rate	10 Hz
Cold Start Sensitivity	-143 dBm
Tracking Sensitivity	-160 dBm
Hot Start First Fix	3 s
Cold Start First Fix	30 s
Horizontal Position Accuracy	2.5 m
Horizontal Position Accuracy (With SBAS)	2 m
Velocity Accuracy	0.05 m/s
Timing Accuracy	25 ns
Acceleration Limit	5 g

Table A.2: GNSS Specifications

COMMUNICATION	
Interface	RS232
Speed	4800 to 1M baud
Protocol	AN Packet Protocol or NMEA
Peripheral Interface	2x GPIO and Auxiliary RS232
GPIO Level	5-20 V
GPIO Functions	1PPS Odometer Stationary Pilot Tube NMEA input/output Novatel GNSSS input Trimble GNSS input AN Packet Protocol input/output Packet Trigger Input Teledyne DVL input Tritech USBL input Custom

Table A.3: Communication Specifications

HARDWARE	
Operating Voltage	5 to 36 V
Input Protection	± 40 V
Power Consumption	100 mA 5V (typical)
Hot Start Battery Capacity	> 24 hrs
Hot Start Battery Change Time	30 mins
Hot Start Battery Endurance	> 10 years
Operating Temperature	- 40°C to 85°C
Environmental Protection	IP67 MIL-STD-810 G
MTBF	> 50 000 hrs
Shock Limit	2000 g
Dimensions (excluding tabs)	30x30x24 mm
Dimensions (including tabs)	30x40.6x24 mm
Weight	37 grams

Table A.4: Hardware Specifications

SENSORS				
PARAMETER	ACCELEROM.	GYROS	MAGNETOM.	PRESSURE
Range (Dynamic)	2 g 4 g 16 g	250°/s 500°/s 2000°/s	2 G 4 G 8 G	10 to 120 KPa
Non-linearity	<0.05 %	<0.05 %	<0.05 %	-
Bias Instability	20μg	3°/hr	-	100 Pa/yr
Scale Factor Stability	<0.05 %	<0.05 %	<0.05 %	-
Cross-Axis Allignment Error	<0.05°	<0.05°	<0.05°	-
Bandwidth	400 Hz	400 Hz	110 Hz	50 Hz

Table A.5: Sensor Specifications